Social Network Analysis of Human Mobility and Implications for DTN Performance Analysis and Mobility Modeling

Theus Hossmann, Thrasyvoulos Spyropoulos, and Franck Legendre

TIK-Report No. 323
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
July 2010

***Abstract*—Social Network Analysis (SNA) has emerged as a promising method for designing data dissemination algorithms over Delay Tolerant Networks (DTN). These algorithms try to identify and exploit macroscopic *regular* relationships between nodes. Despite initial encouraging results, the type and complexity of the alleged underlying social structure has not been sufficiently studied or quantified. In this paper, we perform a systematic study and comparison of 4 mobility traces and 3 state-of-the-art synthetic models with respect to social properties. We represent each model as a weighted contact graph and study community structure, graph spectrum, inter- and intra-community weight distributions, etc. We also discuss the implications for synthetic mobility models. Finally, to underline the importance of these contact graph properties, we (i) show that the delay of distributed estimation depends on the second largest eigenvalue of the normalized weighted contact graph, and (ii) express the performance of various (random and SNA-based) DTN routing schemes as a function of the volume of cuts between communities.**

I. Introduction

The rapid proliferation of small wireless devices creates ample opportunity for novel applications [1], [2], as well as extending the realm of existing ones [3]. *Opportunistic* or *Delay Tolerant Networking* (DTN) [4] is a novel networking paradigm that is envisioned to complement existing wireless technologies (cellular, WiFi) by exploiting a "niche" performance-cost tradeoff. Nodes harness unused bandwidth by exchanging data whenever they are in proximity (*in contact*), with the goal to forward data (probabilistically closer) to a (set of) destination(s). Since actions of interest can only occur during a contact, *contacts and their statistical properties are of key importance in the design and performance evaluation of protocols*.

To this end, a number of efforts have been made to collect mobility traces and analyze contact patterns; this is done either, implicitly, by looking at the access points users are associated with over time [5], [6] or, explicitly, with experiments designed to log peer contacts (e.g. Bluetooth, WiFi ad-hoc) [7], [8], [9]. The majority of these traces reveal a considerable heterogeneity in contact patterns, but also time-of-day periodicity and strong location preference [10], [6]. The amount of "structure" observed implies high (statistical) predictability of these patterns. Nevertheless, the majority of trace analysis research has focused on the *inter-contact* and *contact duration* statistics [11], [12], [13], [14]. Inter-contact times and their distribution are an essential building block for most analytical models for DTN routing [15], [16], [17]. Debate is still ongoing as to whether these are in fact power-law distributed [12], have an exponential tail [13], [14], or have a qualitatively different behavior from one contact pair to another [11].

Despite their importance, inter-contact times are a *microscopic* property of human mobility. Analysis built around them becomes significantly involved when one departs from the exponential assumption [12], or a small amount of heterogeneity is introduced [18]. More importantly, human mobility and resulting contacts are driven by *intention* and *location*; *social* relations between nodes (e.g. friendship) guide a node to decide the destination and the timing of a mobility trip while *location* dictates the path followed. *This creates a rather intricate contact structure that is not readily observable at inter-contact level*. A more abstract, *macroscopic* view of mobility is needed that can better capture the set of node inter-relations, in a tractable manner.

Social Network Analysis (SNA) [19][1] offers such a natural, compact representation of the processes guiding human mobility contacts. SNA has been used with considerable success to model and analyze large networks ranging from citation networks and email chains, to Internet topology and online social networks like Facebook [19]. Recently, SNA has been successfully used to design opportunistic routing protocols for unicast and multicast [20], [21], [22], [23]. There, nodes and contacts between them are represented on a *contact graph*, where a link (or link weight) between two nodes indicates a measured "strong" relationship between them (e.g. frequent or long contacts [21], [24]). A variety of metrics and algorithms could then be used to characterize *node importance* on this graph (e.g. degree centrality), as well as to identify nodes belonging to the same "community" via implicit [20] or explicit [21] *community detection*. A "next hop" is then chosen based on its relative centrality and similarity values. These algorithms have been shown to outperform random [25] and utility-based protocols [26].

Even though these protocols aim to utilize "social" properties of human mobility, the latter have not been systematically studied. What kind of social structure characterizes the contact graphs of existing traces (e.g. scale-free, small world)? Are there strong communities, and how are these communities inter-connected? Is this structure consistent across the

[1] "Complex Network Analysis" or "Network Science" are other terms often used to describe the same body of research.

range of different environments measured? Can sophisticated synthetic mobility models also exhibit those characteristics? These questions are important for two reasons: *first*, to better understand the underlying structure governing human mobility and facilitate the design of improved mobility models; *second*, various processes of interest (e.g. distributed estimation [27], [28], routing [25], [26], [20], [21], etc.), taking place over an opportunistic network, can be modeled as a random process (e.g. random walk, diffusion) over the contact graph (and its embedded communities).

To our best knowledge, the research thread in [21] is the first work to have directly studied some of the social properties of traces, and is the closest one to the first part of our work. In [21], a community detection algorithm (k-clique [29]) is applied to different traces and results on the number of communities, community modularity, and pair-wise contact duration distributions are reported. In this paper, we take this work further along a number of dimensions. First, we use a more generic weight function for the contact graph and apply a different community detection algorithm as well as spectral analysis [30], [31]. Second, in addition to two traces also studied in [21] (for which our results are mostly in agreement) we analyze and compare the properties of two other traces and three synthetic mobility models. Third, we focus our attention to inter-community connections, and inter- and intra-community weight distributions. As it turns out, these not only reveal some limitations of state-of-the-art mobility models, but prove to be key for the performance of distributed algorithms over opportunistic networks.

We summarize our contributions here and outline the rest of this paper. We study the contact graph properties of four collected mobility traces [8], [12], [9], and three recently proposed synthetic mobility models [10], [32], [33] (Section II). We apply a state-of-the-art community detection algorithm [34] and spectral analysis techniques [30] to study community structure and modularity, the nature of inter-community links (e.g. bridging links, bridging nodes, community overlap), inter- and intra-community weight distributions, and other graph characteristics (Section III). Finally, we use our findings to (i) better understand the capabilities and limitations of state-of-the-art mobility models (Section III-E), and (ii) propose an analytical framework that links the performance of various distributed algorithms (namely estimation and routing) over a given opportunistic network to fundamental properties of the respective contact graph (Section IV).

II. Data Description

We define a *contact* as the period of time during which two devices are within radio transmission range of each other and can hence exchange data.

Contact Traces: In order to cover a broad range of mobility scenarios, we use four different contact traces with their characteristics summarized in Table II: the MIT *Reality Mining* [8] (MIT)[2], the iMotes Infocom 2005 (INFO) [7], the ETH [9] (ETH), and a new trace involving 150 people in an outdoor training scenario in the (SWISS) Alps[3].

Synthetic Mobility Models: We also analyze the contact properties of three recent synthetic mobility models [10], [32], [33]. Synthetic models allow one to create different scenarios at will, and examine their impact on measured properties. Furthermore, these three models have been shown to match various properties observed in traces (e.g. inter-contact time distributions), so we would like to assess whether they can also reproduce "social" properties observed in these traces. The models chosen are representative of two different trends in state-of-the-art mobility modeling, namely *location-driven* and *social network driven* models.

Time-variant Community Model (TVCM): In the TVCM model [10], each node is randomly assigned one or more home location areas ("communities") on the plane. Transitions in, out, and between home locations are governed by a simple 2-state Markov Chain as illustrated in Figure 1(a) i.e., with a probability $1-p$ of roaming outside the community and a probability p of staying or getting back in the home location. In any case, nodes move according to a RWP. By choosing different transition probabilities for each node, a large range of heterogeneous node behaviors can be reproduced. We use a simple TVCM scenario throughout our analysis, with only one community per node (for 100 "normal" nodes), and 4 more "gregarious" nodes covering each $\frac{1}{4}$ of the total area as its home community (see Figure 1(a))[4].

Ghost: Ghost [35] combines both microscopic (realistic displacements on a map) and macroscopic (preferred locations) features of mobility. We have used Ghost to reproduce our building's floor plan. Nodes follow a RWP model with waypoints being discrete locations (e.g., offices) and moving from one waypoint to another using the shortest path on the constraint layout. Waypoints and pause times are chosen among a node-specific ranked list of preferred destinations (according to the Zipf law). All nodes are assigned their office as the first ranked destination but for other ranks, nodes have specific destinations simulating social ties (work colleague or friend).

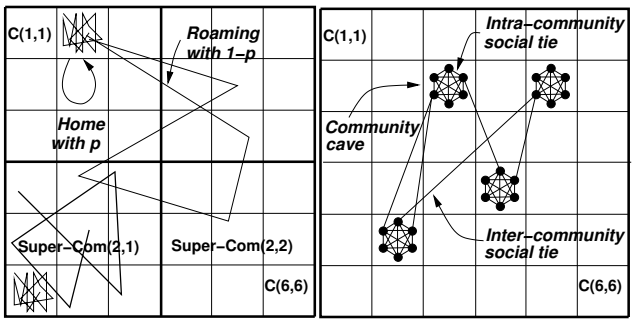
HCMM: The Community-based Mobility Model (CMM) [36] was the first mobility model directly driven by a social network. The Caveman model [19] is used to define a network with (social) communities and each community is assigned to a home location as shown in Figure 1(b). In contrast to TVCM, transition probabilities are directly linked to the weights on the overlay social network. Specifically, the probability that a node i performs a mobility trip towards a community C depends on his social ties (caveman graph weights) towards nodes currently in community C. HCMM [33] adds location-driven mobility to CMM. The transition probability to a location L no longer depends on nodes currently at that location but on the total weight of nodes assigned to L as their home location (i.e., irrespective of their current position).

For the three synthetic models, we log contacts. Some model

[2]Despite its long duration, a lot of short contacts were supposedly not logged in the MIT trace due to its time granularity of 5 minutes. We use 3 months of contacts between September 2004 and December 2004.

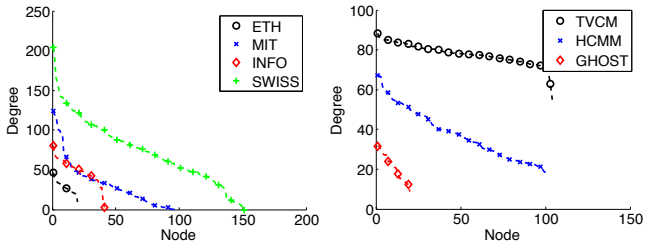
[3]This trace is not public.

[4]TVCM supports much additional complexity (see [10]). We choose here to use the minimum amount of complexity needed to create some non-trivial community structure.



(a) TVCM 6×6 grid with 3×3 super-communities. (b) HCMM grid with caveman social overlay.

Fig. 1. TCVM and HCMM models.



(a) Contact Traces. (b) Mobility Models.

Fig. 2. Node's degrees (ranked).

parameters and statistics are reported in Table I. GHOST is configured to closely match the situation of the ETH trace, whereas the parameters of TVCM and HCMM are chosen to roughly match the other traces in terms of number of nodes and average community size (see Section III).

	TVCM	GHOST	HCMM
Nodes	104	20	100
Speed	1-3 m/s	$\sim \mathcal{N}(1.3,.25)$ m/s	1-3 m/s
Structure	10 Nodes per Home-Location	1-3 Nodes per Office	10 Nodes per Community
Transm. Range	30m	5m	30m
# Contacts	100′000	9′756	100′000

TABLE I
MOBILITY MODEL PARAMETERS.

	Clustering Coefficient				Avg. Path Length			
	5%	**10%**	**25%**	**50%**	**5%**	**10%**	**25%**	**50%**
ETH	0.17	0.27	0.7	0.78	1.2	1.4	2.3	1.4
MIT	0.42	0.57	0.71	0.75	3.3	2.6	2.0	1.5
INFO	0.24	0.3	0.47	0.66	3.5	2.5	1.9	1.5
SWISS	0.27	0.37	0.5	1	3	2.2	2	1
TVCM	0.67	0.95	0.49	0.6	1.3	2.9	1.7	1.5
HCMM	0.67	0.82	0.62	1	1	4.4	1.8	1
GHOST	0.15	0.33	0.63	0.71	0.64	1.8	2.1	1.5

TABLE IV
CLUSTERING COEFFICIENTS AND AVERAGE PATH LENGTHS USING DIFFERENT WEIGHT THRESHOLDS.

A. Contact Graph and Tie Strength

To apply social network analysis tools to each contact dataset, we construct an appropriate graph, a *Contact Graph*, out of the sequence of contacts over time in a trace. We use a *weighted graph*, $\mathbf{W} = \{w_{ij}\}$, for this task. Each node is a vertex on this graph and a link weight w_{ij} represents the strength of the relationship ("tie") between nodes i and j.

A key question is how to assess the tie strength between two nodes, i.e. what w_{ij} should be, as a function of the contacts observed in a trace. Different metrics such as the age of last contact [37], [26], contact frequency [38], [21] or aggregate contact duration [21] have been used as tie strength indicators in DTN routing protocols. These can be seen as contact *features* whose importance depends on the scenario and application. We choose to consider here two features: contact *frequency* and aggregate contact *duration*. We need both dimensions in order to capture different kinds of relationships. *Friends* or *family* may have long meetings which can be frequent or rather rare, whereas *familiar strangers* are characterized by frequent short contacts. The two features are correlated to different degrees as shown in Table III.

ETH	**MIT**	**INFO**	**SWISS**	**TVCM**	**HCMM**	**GHOST**
0.5	0.81	0.6	0.55	0.96	0.97	0.33

TABLE III
PEARSON CORRELATION COEFFICIENTS OF DURATION AND FREQUENCY.

We first assign each pair of nodes $\{i, j\}$ a two-dimensional feature vector, $\mathbf{z_{ij}} = \left(\frac{f_{ij}-\bar{f}}{\sigma_f}, \frac{d_{ij}-\bar{d}}{\sigma_d}\right)$, where f_{ij} is the number of contacts in the trace between nodes i and j, and d_{ij} is the sum of the durations of all contacts between the two nodes. $\bar{f}$ and $\bar{d}$ are the respective empirical means and σ_f and σ_f the empirical standard deviations.

Since most social network analysis methods (e.g., state-of-the-art community detection) require one-dimensional tie strength metrics, we transform the two-dimensional feature vector to a scalar feature value: We use the *principal component* [39], i.e., the direction in which the data vector $Z = \{z_{ij}\}$ has the largest variance. This is the direction of the eigenvector $\mathbf{e_1}$ with the largest corresponding eigenvalue. We then define the tie strength between i and j as the projection of $\mathbf{z_{ij}}$ on the principal component

$$w_{ij} = \mathbf{e_1}^T \mathbf{z_{ij}} + w_0,$$

where we add $w_0 = \mathbf{e_1}^T \left(-\frac{\bar{f}}{\sigma_f}, -\frac{\bar{d}}{\sigma_d}\right)$ (the projection of the feature value for a pair without contacts) in order to have positive tie strengths. This is a more generic metric that combines the frequency and duration in a scalar value and better represents the heterogeneity of node pairs[5].

B. Structural Properties

In the following we use metrics from *complex (weighted) network analysis* to study some basic properties of the weighted contact graphs for each mobility scenario. Figure 2 plots the ranked node's degrees, where the weighted degree of a node i is $d_i = \sum_{j=1}^{N} w_{ij}$ (for N total nodes). In all mobility scenarios the degrees are heterogeneous, particularly in the MIT and SWISS traces.

[5]This framework implicitly assumes stationarity of the underlying process, something not always true in some traces. In practice (e.g., for protocol design), one would implement some sliding window mechanism (see e.g.[24]). An thorough time-dependent analysis of these traces can be found in [40].

	MIT	INFO	ETH	SWISS
Scale and context	92 campus students and staff	41 conference participants	20 lab students and staff	150 people
Period	3 months	3 days	5 days	9 hours
Scanning Interval	300s (Bluetooth)	120s (Bluetooth)	0.5s (Ad Hoc WiFi)	30s (GPS)
# Contacts total	$81'961$	$22'459$	$23'000$	$12'875$
# Contacts per dev.	890	547	$1'150$	85

TABLE II
MOBILITY TRACES CHARACTERISTICS.

We next examine our scenarios for small-world properties, which according to Watts & Strogatz [19] are expressed as a high clustering coefficient (tendency of relations to be transitive) *and* short paths between nodes (a property of random, Erdos-Renyi graphs). In order to compute these two metrics, we apply a weight threshold to the weighted contact graph and convert it to a *binary graph*[6]. For example, a threshold of 10% means that the strongest 10% of the weights are included in the binary graph. The clustering coefficient of node i is defined as (e.g., [19])

$$C_i = \frac{\text{number of triangles connected to } i}{\text{number of triples connected to } i},$$

and the average path length is the shortest path averaged over all pairs of nodes, between which there exists a path.

Table IV shows the average clustering coefficients for different weight thresholds. For a random graph (Erdos-Renyi), the clustering coefficient increases linearly from 0 to 1. Thus, in a graph where 10% of the node pairs are connected, the expected clustering coefficient is 0.1. The values show that all scenarios are considerably more clustered, strongly suggesting non-randomness. For TVCM, the clustering coefficient drops going from 10% to 25% as this adds many random inter-community links that form unclosed triples (the closed intra-community triples are already present before). Such behavior is not observed in the traces. Regarding the average path length, we see that there are consistently short paths between the nodes that are connected. In some cases, e.g., ETH and TVCM, the path length is increased in the steps from 5% to 10% due to disconnected clusters merging.

III. COMMUNITY STRUCTURE ANALYSIS

Having analyzed generic characteristics of our contact data, we now look at more "complex" structures, namely communities. Communities are (informally) defined as subsets of nodes with stronger connections between them than towards other nodes. They usually imply a social group (e.g. friends, co-workers). Although a high clustering co-efficient often correlates with existence of communities, communities often involve more than three nodes and their interpretation is more "subjective". As a result, their exact number, membership, and inter-connection may depend on the community detection algorithm and various thresholds. To ensure our observations are as generic as possible, we use two state-of-the-art community detection methods, namely the Louvain algorithm [34] and spectral clustering [31].

[6]Although average path length and clustering coefficient metrics exist for weighted graphs, they are not as easily interpretable.

A. Community Detection Methodology

Louvain Community Detection. Finding the optimal allocation of nodes to communities is a computationally hard problem, and therefore, state-of-the-art algorithms use heuristics. The Louvain [34] algorithm starts with assigning each node its own community. It then iteratively – until no further improvement is possible – goes through all nodes and moves it to one of the existing communities, such that the gain in modularity is maximal (the Q function [41] is used as a measure for modularity, see Section III-B). In a second step, the communities are merged, if merging increases modularity. These two phases (moving nodes and merging communities) are iteratively repeated until no further improvement is possible. The algorithm is fast and was reported to find communities that are as good or better than other algorithms for a number of different graphs [34].

Spectral Clustering. Spectral Graph Theory [30] studies the structural properties and invariants of the weighted graph defined by $\mathbf{W}$, using eigenvalue decomposition of the Laplacian $\mathbf{L}$. Let us define the normalized *Laplacian* of the weight matrix $\mathbf{W}$ as

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}, \tag{1}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{D}$ is the diagonal matrix whose (i,i)-element $d_{ii} = \sum_j w_{ij}$ (i.e., is the *degree* of vertex i on the matrix $\mathbf{W}$). If $\mathbf{W}$ is block-diagonal (ideal case) i.e., it consists of k connected components with all weights 0 between blocks, the eigenvalues $\lambda_i, i = 1 \dots n$ of $\mathbf{L}$ are:

$$\lambda_1 = \cdots = \lambda_k = 0 < \lambda_{k+1} \cdots \leq \lambda_n. \tag{2}$$

In the non-ideal case, $\mathbf{W}$ is not block diagonal (i.e., is connected with lower weights between clusters), and only the first eigenvalue of $\mathbf{L}$ is 0. Matrix perturbation theory [31] suggests, however, that if the clusters are compact and modular (in other words, identifiable by a human), the eigenvalues corresponding to these clusters will still be small. Spectral clustering [31] uses this to identify k strongly connected components in W. It projects the n points into the eigenspace of L consisting of L's first k eigenvectors, and uses common clustering techniques like k-means [39] on this projection, to infer clusters.

B. Community Detection Results

We apply the two community detection algorithms to the four traces and three synthetic mobility scenarios. The number of identified communities by each method is shown in Table V. In addition to the number of communities, we are interested in the *modularity* of the resulting partition of nodes to communities. High modularity implies strong community structure,

and high potential for node cooperation [42] and community-based trust mechanisms [43]. Yet, it may also imply high convergence times for distributed algorithms, as we shall see in Section IV.

We use the following two metrics to assess modularity. The widely used Q *function*, as introduced by Newman [41]:

$$Q = \frac{1}{2m} \sum_{ij} \left(w_{i,j} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j),$$

where $k_i = \sum_j w_{i,j}$ is the strength of node i and $m = \frac{1}{2} \sum_j k_j$ is the total weight in the network. c_i denotes the community of node i thus, the Kronecker delta function $\delta(c_i, c_j)$ is one if nodes i and j share the community and zero otherwise. $Q = 0$ is the expected quality of a random community assignment and [41] reports modularities of above $Q = 0.3$ for different networks (social, biological, etc.) for state-of-the-art community detection algorithms. As our other modularity metric, we use the second eigenvalue λ_2 of the normalized Laplacian of $\mathbf{W}$ (often called Algebraic Connectivity [30]), which measures how well connected a graph is.

The Q function and algebraic connectivity values for all contact traces and mobility models are also given in Table V. A first observation is that the two clustering algorithms find different number of communities in some cases[7] although modularities are almost identical. A visual inspection reveals that community membership is consistent for both algorithms. Therefore, in the rest of this section, we present all results for the Louvain algorithm since the same conclusions hold also for Spectral Clustering. We will go back to spectral properties and algebraic connectivity in Section IV.

Trace/Model	Alg. Conn.	# Comm.	Q
ETH	0.49	2/2	0.22/0.21
MIT	0.081	6/6	0.52/0.5
INFO	0.66	6/4	0.12/0.11
SWISS	0.094	7/6	0.22/0.22
TVCM	0.4	10/10	0.48/0.48
HCMM	0.11	8/10	0.6/0.59
GHOST	0.42	3/2	0.21/0.2

TABLE V
ALGEBRAIC CONNECTIVITY, NUMBER OF COMMUNITIES AND MODULARITY (Q) (THE FIRST VALUE IS USING LOUVAIN, THE SECOND FOR SPECTRAL CLUSTERING).

A second observation is that modularity varies broadly among the traces: the MIT trace is highly modular while ETH and SWISS have a lower modularity and INFO a very low modularity. Similar values for other community detection algorithms (K-Clique and Newman), different traces and other strength metrics (total contact duration) have already been reported in [21], thus our findings are in agreement. Note also the high modularity of the synthetic scenarios. This is a first evidence that existing models can emulate highly modular community structure with simple scenarios.

Finally, the algebraic connectivity values correlate with the modularities (i.e., high modularities mean small algebraic connectivities) for the traces. However, the algebraic connectivity of the TVCM model does not. We believe this is due to the following: while the MIT and SWISS traces are modular (i.e. tight communities) they have very weak inter-community connections. This results in a small algebraic connectivity. On the other hand, the TVCM scenario is also modular, but the 4 gregarious nodes make sure that most groups of nodes are not very badly connected to the rest. This will be confirmed in Section III-D, where we investigate more closely how the inter-community weights are distributed. In other words, algebraic connectivity and modularity sometimes capture different qualitative characteristics.

[7]In the case of SWISS there are 2 nodes that form communities on their own. We excluded these from the analysis.

C. Intra-Community Ties

High modularity implies that a community's interior nodes are "well connected". However, it does not say anything about *how* these are connected. Hence we look at the distribution of nodes' intra-community weighted degree. For a community C_A and a node $i \in C_A$, its intra-community weighted degree or "internal degree" is:

$$d_{int}(i) = \sum_{j \in C_A} w_{ij}$$

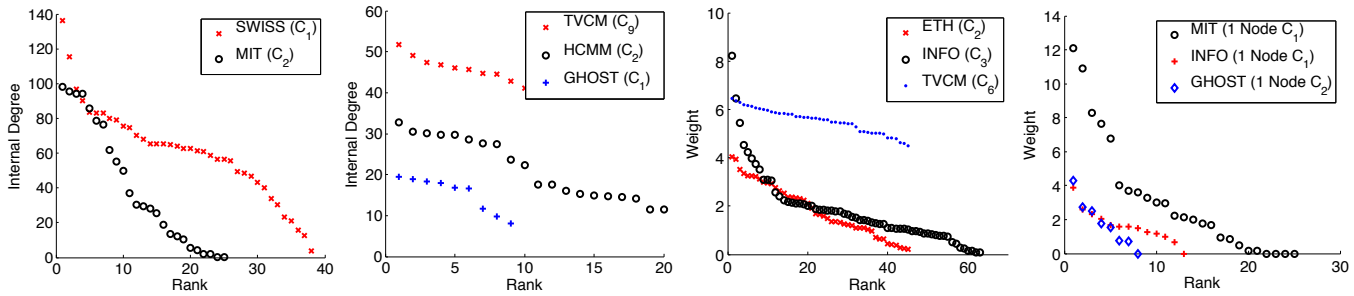
For each community, we rank and plot the distribution of internal degrees over all nodes of a community as shown in Figures 3(a) and 3(b). We see that internal degrees are skewed for the traces and slightly decreasing for models.

We further rank and plot individual link weight distributions: for all links in a community (Fig. 3(c)) and for links of a *given* node inside its community (Fig. 3(d)). We observe that the weights are strongly skewed for the traces. Although few examples are shown, due to space limitations, this observation has been remarkably consistent across all traces and communities. We conclude that *a community can thus not be thought of as a homogeneous group of strongly connected nodes (like a mesh).* Instead, there is strong heterogeneity even within a community. Synthetic models reproduce this skewed distribution to various degrees. We discuss this further in Section III-E.

D. Inter-Community Ties

We now focus on the interface *between* communities. Table VI shows how the total weight in the network is distributed within and between the communities, for all datasets. Note that the inter-connections of communities are weak in many cases. For instance, in the MIT trace, communities 1 and 2 together contain more than 50% of the weights and 50% of the nodes. However, between them there is only 2% of the weight. How "thick" or "thin" inter-community cuts are has an important impact on the amount of information that can be exchanged between two communities, and how fast this can be done (see Section IV).

In addition to the *total* weight between communities, it is important to know how this weight is distributed among nodes and links connecting two communities. We aim to identify the type of interface as either (i) bridging links, (ii) bridging nodes (overlap), or (iii) hierarchical communities, defined as follows:



(a) Trace Intra-Degrees (b) Models Intra-Degrees (c) Intra-Weights (d) Intra-Weights per Node

Fig. 3. Ranked Intra-Community Weights and Node Degrees for both traces and synthetic models.

(a) ETH

	1(10)	**2**(10)
1(10)	**45%**	26%
2(10)	26%	**30%**

(b) MIT

	1(24)	**2**(23)	**3**(16)	**4**(16)	**5**(7)	**6**(6)
1(24)	**25%**	2.3%	0.8%	1.1%	0.034%	0.097%
2(23)	2.3%	**27%**	3.6%	7.6%	0.45%	1.4%
3(16)	0.8%	3.6%	**9.2%**	3.9%	0.19%	0.96%
4(16)	1.1%	7.6%	3.9%	**9.7%**	0.29%	0.94%
5(7)	0.034%	0.45%	0.19%	0.29%	**3.1%**	0.17%
6(6)	0.097%	1.4%	0.96%	0.94%	0.17%	**2.2%**

(c) INFO

	1(13)	**2**(12)	**3**(6)	**4**(5)	**5**(3)	**6**(2)
1(13)	**14%**	18%	8.4%	4.3%	4.1%	2.1%
2(12)	18%	**12%**	8.6%	4.5%	3.7%	2.3%
3(6)	8.4%	8.6%	**4%**	2.2%	1.8%	1.3%
4(5)	4.3%	4.5%	2.2%	**1.3%**	1%	0.64%
5(3)	4.1%	3.7%	1.8%	1%	**3.4%**	0.6%
6(2)	2.1%	2.3%	1.3%	0.64%	0.6%	**1.4%**

(d) GHOST

	1(9)	**2**(8)	**3**(3)
1(9)	**33%**	24%	6.4%
2(8)	24%	**21%**	9%
3(3)	6.4%	9%	**6.3%**

(e) TVCM (5 rand. selected communities)

	1(11)	**3**(11)	**4**(11)	**6**(10)	**9**(10)
1(11)	**6%**	1%	1%	0.94%	0.75%
3(11)	1%	**6.3%**	1.4%	0.85%	1.3%
4(11)	1%	1.4%	**5.9%**	0.95%	1.1%
6(10)	1%	0.85%	0.95%	**6%**	0.85%
9(10)	1.1%	1.3%	1.1%	0.85%	**5.7%**

(f) SWISS

	1(41)	**2**(41)	**3**(38)	**4**(19)	**5**(11)
1(41)	**24%**	8.1%	12%	7.5%	0.064%
2(41)	8.1%	**14%**	11%	3.5%	0.025%
3(38)	12%	11%	**11%**	3.2%	0.027%
4(19)	7.5%	3.5%	3.2%	**5.4%**	0.04%
5(11)	0.064%	0.025%	0.027%	0.04%	**0.96%**

TABLE VI

PERCENTAGES OF TOTAL WEIGHT WITHIN AND BETWEEN COMMUNITIES. THE ROWS AND COLUMNS ARE COMMUNITY INDICES WITH THE NUMBER OF NODES IN THE RESPECTIVE COMMUNITY SHOWN IN BRACKETS.

Definition 3.1: Let us look at two communities C_A and C_B and identify the cut $\partial(C_A, C_B) = \{(i,j) : i \in C_A, j \in C_B\}$. Then, we define the following three types of inter-community connections, based on the relative weight of different links and different nodes in $\partial(C_A, C_B)$.

- *(Bridging link)* A link (i,j) is a bridging link if
$$w_{ij} \gg median\{w_{kl} \in \partial(C_A, C_B)\}.$$
- *(Bridging node)* Define $w_{i,C_j} = \sum_{j \in C_B} w_{ij}$. A node $i \in C_A$ is a bridging node if
$$w_{i,C_B} \gg \underset{k \in C_A}{median}\{w_{k,C_B}\}.$$
- *(Hierarchy)* Communities A and B form a hierarchy if there is no bridging node or bridging link between them and
$$\frac{vol(\partial(C_A, C_B))}{\min\{|C_A|, |C_B|\}} \gg \underset{C_j}{median}\{\frac{vol(\partial(C_A, C_j))}{\min\{|C_A|, |C_j|\}}\}.$$

In other words, a bridging link implies that a node in one community has a particularly strong link to a node in another community (but to no other nodes in that community). A bridging node on the other hand "knows" many nodes in *both* communities (making it more useful to carry information across communities). Finally, a hierarchy implies that most nodes in the two communities have some ties with each other, but not perhaps as strong as inside their own community (e.g. two groups in the same division).

Note also that this distinction is somewhat subjective. For example, if there are more than one bridging node we can say that the two communities *partially overlap* (but do not form a hierarchy). Furthermore, certain community detection algorithms such as k-clique inherently identify some of these interfaces. However, neither of them provides a distinction between all the three types of inter-connection. We will characterize any connection not falling into these categories as "flat".

There is a number of reasons why one might care about how inter-community weight is distributed. First, it is related to the number of links or nodes that need to be removed in order to "disconnect" the network [44]. Even if the total inter-community weight is large, if this is concentrated on a very small number of nodes (links), losing these few nodes (links) would suffice to severely hamper the capacity for sharing information or content across communities in an opportunistic network. This would be the case, for example, if the bridging node runs out of battery (something not unlikely given that this node will be overused by smart SNA-based algorithms) or decides to not forward traffic.

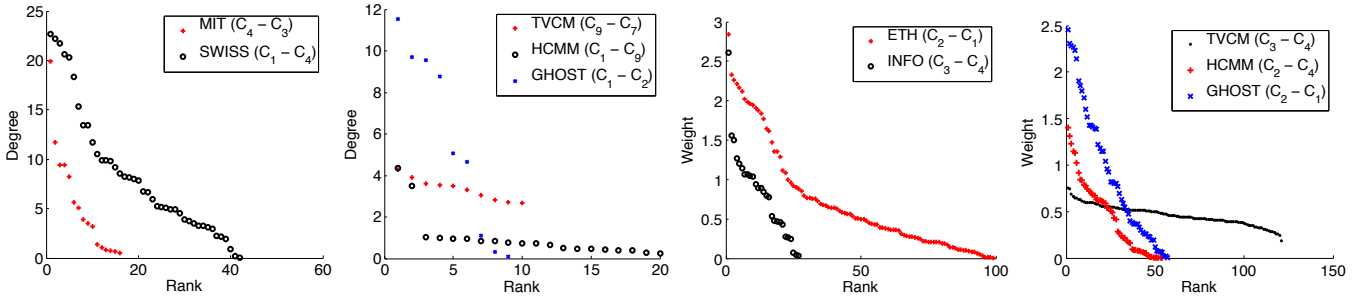
In order to identify the type of inter-community conductance we test the cut between communities according to Definition 3.1. Table VII shows for all traces the results for the 5 strongest inter-community connections ("cuts")[8], i.e., the type of interface and – in case of bridging nodes/links – how much of the cut strength is concentrated around them. Note that not all community inter-connections must have a type according to Def. 3.1. We see that different traces show

[8]Stronger cuts are more interesting since these contain most of the "capacity" (e.g. to carry information) between communities.

MIT		ETH		INFO		SWISS		TVCM		HCMM		GHOST	
2 ↔ 3	Hierarchy	1 ↔ 2	flat	1 ↔ 2	flat	1 ↔ 3	Node (10.1%)	3 ↔ 4	flat	1 ↔ 7	Node (21.8%)	1 ↔ 2	flat
2 ↔ 4	Hierarchy			2 ↔ 3	flat	1 ↔ 3	Node (7.4%)	1 ↔ 10	flat	2 ↔ 8	flat	2 ↔ 3	flat
3 ↔ 4	Link (16.8%)			1 ↔ 3	flat	2 ↔ 3	flat	3 ↔ 9	flat	1 ↔ 2	flat	1 ↔ 3	flat
1 ↔ 2	Link (27.1%)			2 ↔ 4	flat	1 ↔ 2	flat	2 ↔ 6	flat	3 ↔ 7	Node (31.3%)		
1 ↔ 3	Link (44.1%)			1 ↔ 4	flat	1 ↔ 4	flat	4 ↔ 9	flat	4 ↔ 5	Node (46.0%)		
1 ↔ 3	Node (44.8%)					2 ↔ 4	Node (12.4%)						

TABLE VII

BRIDGE TYPES. THE NUMBER IN PARENTHESIS IS THE PERCENTAGE OF THE TOTAL AMOUNT OF THE CUT THAT IS CONCENTRATED IN THE BRIDGE. WE USE "FLAT" TO DENOTE NO PARTICULAR IDENTIFIED STRUCTURE.



(a) Trace Inter-Degrees (b) Models Inter-Degrees (c) Trace Inter-Weights (d) Models Inter-Weights

Fig. 4. Ranked Inter-Community Weights and Node Degrees for both traces and mobility models.

different behavior. For instance in MIT, there is a strong tendency for communities to be linked by bridging links, whereas the SWISS trace tends to have bridging nodes. As we have already observed, the INFO trace shows relatively less structure and Def. 3.1 does not assign any type to the community pairs. In case of the mobility models, we do not observe any bridging links. We discuss this in Section III-E.

Figure 4 plots the ranked inter-community weighted node degree and inter-community weights for the traces and synthetic models. For the distribution of inter-community weighted degree per node (i.e., all weights of a given node $i \in C_i$ towards any node in some other C_j) we see two examples in Figures 4(a) and 4(b). Again, we observe very skewed distributions of weights for the traces with degrees strongly concentrated in a few nodes (actually ETH and INFO trace degrees). For synthetic models, behaviors are mixed ranging from uniform to highly skewed.

Figure 4(c) and 4(d) plot the inter-community link weights for the traces and models. For the traces, although the distributions differ, in almost all cases weights show a lot of variation and distributions are skewed (a subset of plots are only shown, due to space limitations). For the models, again, the behavior differs.

E. Implications for Mobility Modeling

From the range of results presented, it is evident that there are a number of similarities between the traces and synthetic mobility models, but also a number of important differences.

The most important finding is that not all synthetic models can easily reproduce the skewed distributions for inter- and intra-community link weights (prevalent among traces). All synthetic models tend to also produce more uniform internal degree distributions. Finally, we did not observe a bridging link in any of the synthetic scenarios. We present here a possible interpretation.

In TVCM, destinations are chosen uniformly inside a home community (and outside). This results in relatively uniform weights as, in the long run, every node will see every other node in a given "class" (e.g. "nodes with same home location", "nodes only met randomly") equally often. Multi-tier communities can be introduced [10] with skewed transition probabilities to emulate the skewed location preference observed in WLAN traces. These could also be tuned to emulate observed weight distributions, yet only through detailed "model fitting". Ghost on the other hand uses a detailed map and allows defining destinations with a finer granularity. This, together with the use of a skewed destination preference (Zipf) better reproduces skewed distributions at the intra and inter-community levels. Although the skewed location preference is well-motivated and matches our intuition, it is essentially hard-wired, with no clear process driving it per node. Nevertheless, *both TVCM and Ghost are location-driven models and thus destinations can only be locations.*

HCMM takes a different approach. Although it is ultimately location-driven, as TVCM, individual behaviors (i.e. transition probabilities between communities) are more naturally derived, through an overlay social network. Without explicitly reverse-fitting any trace, this simple transformation of social weights to transition probabilities [33] seems to be equally successful at capturing microscopic (e.g. weight distributions) and macroscopic (e.g. modularity) behaviors. Nevertheless, closer observation reveals that HCMM also falls short. Internal degree distributions are also more uniform than traces. In fact, since destinations are ultimately home/remote communities *most nodes in a visited remote community are expected to become familiar strangers*. Much of the difference observed,

is only due to the varying amount of time each node stays in its community.

Further consideration of the mechanisms behind the three models suggests that they also all suffer from the same limitation. Although they are more or less able to capture contact related phenomena arising from *location-driven* mobility actions they are less ready to capture mobility driven by explicitly *social intentions*. All three models ultimately see *locations as destinations*, fixed and decided at the beginning. They do not see other *nodes as destinations*. Experience tells us that our intention to see node X, will guide us to go *to the location where X is, at the time that X is there.* [36] *does* seem to want to capture this, but fails in the implementation [33]. We suspect that social intention is one of the reasons behind the consistently skewed weight distributions in traces, as nodes go to meet other nodes selectively, not necessarily contacting other nodes in the destination's home community[9]. Furthermore, *meetings between nodes with social relations do not necessarily occur in one's home community.*

Concluding, we believe that models should put more emphasis on the social aspects of mobility. Capturing *both* location-driven mobility decisions and social relationship driven ones seems necessary to accurately reproduce the social properties observed in real traces. Furthermore, it seems that group mobility primitives (i.e. biasing the decision of node X based on what his friend Y is doing, perhaps in a time-dependent manner) could still be relevant.

IV. Distributed Algorithms

Community structure, strength, and type of community inter-connections are also important for various distributed algorithms, e.g. distributed estimation and gossip-based diffusion [27], [45], [28], load balancing and congestion control [46], [47], and DTN routing [21], [20], [25], that could run over opportunistic networks. In this section, we show that the performance of these processes is directly linked to properties of the *contact graph*, such as its *algebraic connectivity*, the *conductance* of the cut between communities, and the distribution of weights across a cut, studied in Section III.

A. Distributed Estimation

Distributed estimation of network parameters or global network state is a key component for the correct and efficient operation of numerous DTN algorithms. For example, in [25], an estimate of the network size is needed in order to tune the number of copies used by the protocol. Furthermore, optimal buffer management protocols such as [48], [47] require an estimate of the total number of existing message replicas. [28] argues for the systematic study of distributed estimation problems in DTNs, and examines pair-wise and population methods. Nevertheless, no analytical treatment of the convergence of these algorithms is available.

[9]A note is due here: Eventually traces themselves are also limited in capturing "everything going on" since (i) WiFi and bluetooth based logging have their own pitfalls, and (ii) the resulting contact traces using these technologies jointly aggregate the effect of two processes, mobility and radio propagation.

Distributed estimation and gossip-based diffusion, on the other hand, have been extensively studied in the context of distributed databases and systems [27], [45]. There, a connected network is considered, such as a social network, a peer-to-peer network, or a (connected) ad hoc network, with applications being rumor spreading, estimation, distributed data fusion, etc. Time is counted in rounds, and in each round, one or more links are chosen, with some probability, over which content is shared/fused to estimate some aggregate parameter (e.g. average, cardinality, min, max). This probability matrix defines a *randomized gossip algorithm*. The number of rounds needed for convergence is found to be linked to the second largest eigenvalue of the expectation of this matrix.

Here, we will use the framework of [27] as it can treat non-complete graphs and asynchronous rounds. However, in our case, the network is not connected and *link activation depends only the mobility model*. We need to modify the proposed framework accordingly. Specifically, we assume that *the clock ticks at consecutive contact intervals* (i.e. a contact between *any* nodes). We discuss how this translates into real time, later in this section.

Let us assume that each node i possesses a value x_i and we want all nodes to calculate the average value over all x_i (this process can be used to calculate other aggregate parameters such as cardinality, min/max, etc.). If $\mathbf{x}(k) = \{x_1(k), \ldots, x_i(k), \ldots, x_N(k)\}$ is the current (N-dimensional) vector value at clock tick k, then in the next tick

$$\mathbf{x}(k+1) = \mathbf{R}\mathbf{x}(k),$$

where $\mathbf{R}$ is an $N \times N$ random matrix. If the next contact is between nodes i and j (with some probability p_{ij}), then the matrix $\mathbf{R}$ is equal to

$$R_{ij} = I - \frac{(e_i - e_j)(e_i - e_j)^T}{2}, \tag{3}$$

where e_i is an $N \times 1$ vector with the i^{th} component equal to 1, and all other components zero.

We therefore need to define these probabilities p_{ij} in our context. In lack of other information, it is reasonable to assume that the probability p_{ij} that the next contact is between nodes i and j is proportional to the edge weight w_{ij} in the contact graph $\mathbf{W}$[10]:

$$p_{ij} = \frac{w_{ij}}{\sum_{(k,l)} w_{kl}} \tag{4}$$

Let us define the volume of a subset of vertices $S \subseteq G$ as $vol(S) = \sum_{i \in S} d_i = \sum_{i \in S} w_{ij}$. It is easy to see then that

$$E[\mathbf{R}] = \frac{2\mathbf{W}}{vol(\mathbf{W})}. \tag{5}$$

We are now ready to connect the convergence results of [27] to the contact graph of a given mobility scenario. Let us denote the target vector $\mathbf{x}(k)$ as $x_{ave}\mathbf{1}$, where $x_{ave} = \sum_i x_i(0)$. We define the ϵ-averaging time ($0 < \epsilon < 1$), as in [27], with the difference that, instead of an asynchronous gossip algorithm

[10]See also [22], [23] for more rigorous arguments.

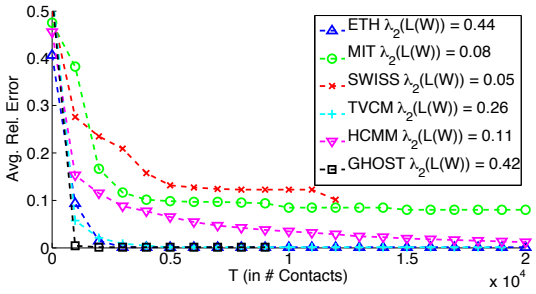


Fig. 5. Average relative estimation error.

$\mathcal{A}(\mathcal{P})$, we now have an opportunistic network with weighted contact graph $\mathbf{W}$. Then,

$$T_{ave}(\epsilon, W) = \sup_{x(0)} \inf \left\{ k : Pr\left(\frac{\|\mathbf{x}(k) - x_{ave}\mathbf{1}\|}{\|\mathbf{x}(0)\|} \geq \epsilon \right) \leq \epsilon \right\}. \tag{6}$$

Hence, $T_{ave}(\epsilon, W)$ denotes the time at which all nodes in our opportunistic network will have converged close to the desired average value, with high probability. The following Lemma is the direct application of Theorem.3 of [27] to our modified framework, and is presented without proof[11].

Lemma 4.1: The averaging time $T_{ave}(\epsilon, W)$ over an opportunistic network with weighted contact graph W (in terms of number of contacts) is bounded as follows:

$$\frac{0.5 \log \epsilon^{-1}}{\log\left(\frac{vol(W)}{2}\lambda_2(W)^{-1}\right)} \leq T_{ave}(\epsilon, W) \leq \frac{3 \log \epsilon^{-1}}{\log\left(\frac{vol(W)}{2}\lambda_2(W)^{-1}\right)}, \tag{7}$$

where $\lambda_2(W)$ is the 2nd largest eigenvalue of W.

The eigenvalues of $\mathbf{W}$ and of the Laplacian $L(W)$, calculated in Section III, are connected through Eq.(1).

To validate this result, we run an averaging process over the real and synthetic mobility traces studied in Section III. Fig. 5 relates the convergence time for the averaging process on each scenario and the respective algebraic connectivity. As can be seen there, scenarios like MIT and SWISS with small $\lambda_2(L(W))$ (i.e., large $\lambda_2(W)$ in Lemma 4.1), also take more time to converge.

Cheeger's inequality [30] further connects the speed of the averaging algorithm with inter-community cut weights, measured in Section III. To see this, let us isolate two communities of graph $\mathbf{W}$, say C_A and C_B, and consider the subgraph $W_{A,B} = \{V_{A,B}, E_{A,B}\}$, where $V_{A,B} = C_A \cup C_B$ and $E_{A,B} = \{(i,j) : i,j \in C_A \cup C_B\}$. Let us further define the *conductance* of the cut $\partial(A,B)$[12], as

$$\phi_{A,B} = \frac{vol(\partial(C_A, C_B))}{\min\{vol(C_A), vol(C_B)\}}.$$

Then, we can use Cheeger's inequality to connect the weight of the inter-community cut and the second eigenvalue of the Laplacian of $W_{A,B}$, $\lambda_2(\mathcal{L}(\mathcal{W}_{\mathcal{A},\mathcal{B}}))$ as follows:

Lemma 4.2: If $W_{A,B}$ denotes the subgraph defined by two communities C_A and C_B, and $\phi_{A,B}$ is the conductance of the inter-community cut, then

$$\frac{\phi_{A,B}^2}{2} \leq \lambda_2(\mathcal{L}(\mathcal{W}_{\mathcal{A},\mathcal{B}})) \leq \in \phi_{\mathcal{A},\mathcal{B}}. \tag{8}$$

Lemmas 4.1 and 4.2 together imply that the speed by which two communities in a given opportunistic network can share information or fuse data, spread in both communities, strongly depends on the conductance (i.e. the percentage of total contact weight) of the cut between the communities.

1) From Contact Times to Link Weights and Back: We would like here to discuss some of our assumptions and their implications. Through Eq.(4), we are essentially assuming that the number of contacts until (i,j) happens (starting from the stationary distribution) is geometrically distributed with parameter p_{ij}. The geometric assumption is in accordance with [27], yet one might question to what extent this is applicable for the opportunistic networks at hand.

In this direction [22] assumes that individual contact pairs can be modeled as independent Poisson processes with rates equal to the respective link weight (it easy to check that this assumption maps to our discrete, contact-driven framework). The authors there validate this assumption on two popular traces (INFO and MIT, also studied here), and find it to hold for a significant percentage of contact pairs. [49] also studies *inter-contact time* distributions of traces and reports that a subset of them are exponentially distributed (in accordance with [22]) but most of them can also be fit with a lognormal (perhaps somewhat contradicting [22]).

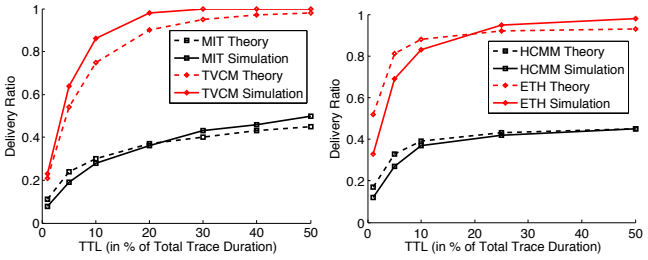
Finally, it is important to note that, in our framework, we are only interested in *first contact times* and not *inter-contact times*. These are more commonly referred to, in the theory of Random Walks on Graphs [50], as *hitting times* and *first return times*, respectively. While the latter are often not memoryless (this is easy to see for a random walk on a 2D lattice), hitting times on subsets of graph vertices often have an exponential tail [50], [13], [14]. We believe that these observations provide sufficient support for the applicability of this framework in Sections IV-A and IV-B. Results in Section IV-B give further validation.

As a final step, we discuss how one could go back from "contact clock ticks" to actual time. The authors of [27] assume exponential clock ticks and use the law of large numbers to show that the exact time of the n-th tick is highly concentrated around its average. A slightly more generic statement can be maded based on Renewal Theory and *Wald's equation* [51]. Let the times of consecutive contact events be renewals. If time is counted at renewal times (contact times), it is easy to see that k in Eq.(6) and delay quantities derived in Section IV-B, are *stopping times*. We can apply Wald's equation [51] to get the following Lemma.

Lemma 4.3: Let the average time between consecutive contact events be $E[C]$. Let further T_{st} denote the delay or convergence time of a given process over an opportunistic contact graph $\mathbf{W}$, in terms of number of contacts. Then, the expected delay of this process is equal to $E[C] \cdot E[T_{st}]$.

[11] In [27] both the random averaging matrix and its expectation are denoted as W. To avoid confusion, we denote the former as R.

[12] Often called "sparsity", with conductance being the minimum sparsity for the whole graph. In the case of $W_{A,B}$ it coincides.



(a) MIT, TVCM (b) HCMM, ETH

Fig. 6. Avg. Delivery Ratio as a function of TTLs (in % of trace duration) according to Eq. 9.

B. Random and SNA-based DTN Routing

In this section, we are interested in predicting the performance of DTN routing over different opportunistic networks *as a function of the properties of its weighted contact graph*. Due to space limitations, we only provide sketches of proofs wherever needed. In light of Lemma 4.3, all time quantities are measured in number of contacts.

1) Performance of Direct Transmission: In Direct Transmission the source holds on to a message until it encounters the destination itself [52]. Its performance has been studied for simple synthetic mobility models [17]. Theorem 4.4 derives the delivery probability for Direct Transmission on a generic opportunistic network[13], as a function of the weighted graph matrix $\mathbf{W}$.

Theorem 4.4: Let T_{ttl} denote the time-to-live for a packet. Then, the expected delivery ratio for direct transmission is

$$P_{deliver} = 1 - \frac{1}{N^2}\sum_{i,j}\left(1 - \frac{2w_{ij}}{vol(W)}\right)^{(T_{ttl}+1)}. \quad (9)$$

Proof: Let us pick uniformly a random source destination pair (i, j). Then, according to Eq.(4) and Eq.(5) the number of contacts until i and j meet for the first time is geometrically distributed with probability $p_{ij} = \frac{2w_{ij}}{vol(W)}$. Thus, the probability of delivery for this pair is

$$P_{deliver}(i,j) = 1 - \left(1 - \frac{2w_{ij}}{vol(W)}\right)^{T_{ttl}+1}.$$

Averaging over all source-destination pairs gives us Eq.9. ■

Fig. 6 compares analytical and simulation results for the delivery delay of Direct Transmission on some of the mobility scenarios. As can be seen there, despite the simplicity of the framework and the assumptions discussed in Section IV-A1 the analytical formula of Eq.(9) follows the observed simulation values well.

2) Delay of Spray and Wait: Spray and Wait [25] is a "random" DTN routing scheme that distributes a limited number of message copies to the first few nodes encountered. When all copies are distributed, each of the relays with one performs Direct Transmission. Spray and Wait has been found to perform favorably compared to epidemic-based DTN routing schemes (in terms of the delay-resources tradeoff), but only in homogeneous scenarios. Its delay for environments with skewed node degree distributions is studied in [18]. Theorem 4.5 provides a closed-form lower bound for the basic Spray and Wait scheme in a generic opportunistic network.

Theorem 4.5: Let M be the number of copies used by Spray and Wait. Consider two communities, C_i and C_j, and let $M \leq |C_i|$. Then, the expected delay of Spray and Wait between nodes $i \in C_A$ and $j \in C_j$ is

$$ET_{SW}(C_i \to C_j) \geq \frac{vol(W)|C_i||C_j|}{2M}\frac{1}{vol(\partial(C_i, C_j))}, \quad (10)$$

Proof: (*This is a sketch of proof*) The delay of Spray and Wait consists of two components, the delay of the spray phase, ET_{spray}, plus the delay of the wait phase, ET_{wait}. Since the destination is in a different community and $M \leq |C_i|$, the probability that the destination is found during the spray phase is negligible [25]. Consequently,

$$ET_{SW}(C_i \to C_j) = ET_{spray} + ET_{wait} \geq ET_{wait}.$$

If a contact graph is highly modular and community cuts small, then the above bound is tight.

Let's denote the set of relays with a copy, after the spray phase finishes, as $L_i = \{i_1, i_2, \ldots, i_L\}$. Then, the expected remaining delay, $ET_{wait|L_i}$ is

$$ET_{wait|L_i} = \frac{\frac{1}{2}vol(W)}{w_{i_1 j} + w_{i_2 j} + \cdots + w_{i_L j}}. \quad (11)$$

Since the set L_i is random, we need to calculate the expectation over all L_i, $ET_{wait} = E[ET_{wait}|L_i]$.

$ET_{wait|L_i}$ is a convex function of $w_{i_1 j}, w_{i_2 j}, \ldots$. We can thus use Jensen's inequality to get

$$E\left[\frac{\frac{1}{2}vol(W)}{w_{i_1 j} + w_{i_2 j} + \cdots + w_{i_L j}}\right] \geq \frac{\frac{1}{2}vol(W)}{M \cdot E[w_{ij}|i \in C_i, j \in C_j]}.$$

This bound holds independent of the choice of node i. Furthermore, $E[w_{ij}|i \in C_i, j \in C_j]$ can be written as $\frac{vol(\partial(C_i,\{j\}))}{|C_i|}$.

Finally, we average over all destinations $j \in C_j$, using the same argument (Jensen's inequality and the convexity of the function with respect to the choice of j) to get

$$ET_{SW}(C_i \to C_j) \geq \frac{\frac{1}{2}vol(W)}{M\frac{vol(\partial(C_i,C_j))}{|C_i||C_j|}}.$$

■

It is clear that the delay of Spray and Wait between two communities is inversely proportional to the volume of the cut between the communities. Furthermore, the above bound is tighter the more evenly distributed the total cut weight is among nodes in the two communities. In the presence of strong bridging nodes it becomes less tight. Fig. 7 shows the computed bound and the simulated delay, for inter-connections of communities with at least M nodes. Observe that the bound is followed more closely for traces that have less strong bridging elements (i.e., TVCM and SWISS).

3) Delay of Social Routing Schemes: In this last part, we analyze a simple generic SNA-based routing scheme in order to get a feel of their performance advantage in the presence of strong social structure[14]. Let us consider again the random

[13]The observed skewed weight distributions make an average deliver delay expression less useful.

[14]The complexity of full-fledged SNA-based DTN protocols like [21], [20] does not permit us their detailed analytical treatment here, which we defer for future work.

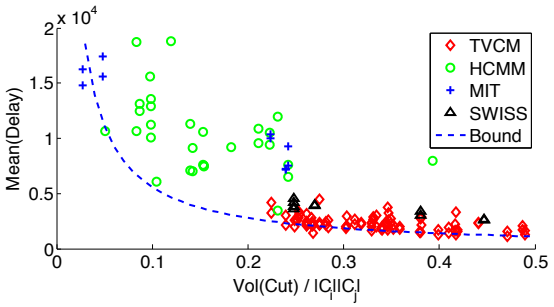


Fig. 7. Avg. Delivery Delay of Spray & Wait ($M = 5$) for different community pairs related to Eq. 10.

spraying scheme of Theorem 4.5, but now assume that a relay in the wait phase is able to recognize a node belonging to the same community as the destination; the relay then hands over its copy to that node as it has a higher delivery probability. This could be done, for example, with a community detection algorithm as in [21] or using similarity as in [20]. It is a hybrid scheme using random replication at the beginning, but allowing some SNA-based forwarding in the "wait" phase (similar to the multi-copy scheme proposed in [53], without the centrality part).

Theorem 4.6: Let M be the number of copies used by Spray and Wait with Community Detection (CD). Consider two communities, C_i and C_j, and let $M \leq |C_i|$. Then, the expected delay of Spray and Wait (CD) between nodes $i \in C_A$ and $j \in C_j$ is

$$ET_{SW-CD}(C_i \to C_j) \geq \frac{vol(W)|C_i|}{2M} \frac{1}{vol(\partial(C_i, C_j))}. \quad (12)$$

Proof: (*This is a sketch of proof*) We can perform the same analysis as in Theorem 4.5. The only difference is that in the wait phase, we only need to encounter any node in community C_j, instead of a specific node j as in the Spray and Wait case. Thus,

$$ET_{wait} \geq \frac{vol(W)|C_i|}{2M} \frac{1}{vol(\partial(C_i, C_j))}.$$

The last leg of the delivery process, from the relay found in community C_j to the destination, incurs additional delay. This delay doesn't change the direction of the inequality. ■

Eq.(10) and Eq.(12) imply that Spray and Wait (CD) can be up to a factor of $|C_j|$ faster than simple Spray and Wait. Clearly, we are comparing bounds here, and such conclusions should be considered only as hints and not solid evidence. Although it is not difficult to derive more detailed expression to account for all components of the the above delays, this is beyond the scope of this paper. Table 4.6 show the actual performance gain between the two schemes for different scenarios, as a function of message copies used.

V. Conclusions

In this paper, we study the "social" properties of four mobility traces and three synthetic mobility models. We use Social Network Analysis and Spectral Analysis to compare various properties of interest with a special focus on community structure and inter-community connections. Our findings suggest not only that collected traces differ qualitatively from each other, but also that state-of-the-art synthetic models fail to capture some behaviors consistently appearing in traces. Finally, we propose a framework to analyze the performance of various distributed algorithms (such as distributed estimation and routing) over an opportunistic network. We prove that this performance strongly depends on key properties of the contact graph such as its algebraic connectivity and conductance.

In future work, we plan to look deeper into modeling various distributed processes of interest over opportunistic networks as a random process over the respective contact graph. We also plan to explore how to adapt or amend existing mobility models to be able to emulate social properties more realistically. Finally, the presence of bridging links and bridging nodes raise some interesting issues related to congestion control, power management, and incentives.

	TVCM	HCMM	GHOST	MIT	ETH
$M = 2$	2.80	1.77	2.36	1.63	1.55
$M = 4$	1.97	2.06	1.72	1.53	1.28
$M = 8$	1.48	1.79	-	1.39	1.02

TABLE VIII
Average Performance Gain of Spray & Wait (CD) vs. Spray & Wait. GHOST has no community pair with more than $M = 8$ nodes in the communities.

References

[1] The Aka Aki Network. http://www.aka-aki.com/. [Online]. Available: http://www.aka-aki.com/

[2] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn, "The rise of people-centric sensing," *IEEE Internet Computing*, vol. 4, no. 12, pp. 12–21, July 2008.

[3] G. Karlsson, V. Lenders, and M. May, "Delay-Tolerant Broadcasting," *IEEE Transactions on Broadcasting*, vol. 51, no. 1, March 2007.

[4] "Delay tolerant networking research group," *http://www.dtnrg.org*.

[5] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *ACM MOBICOM*, 2004.

[6] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, June 2008.

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

[8] N. Eagle, A. Pentland, and D. Lazer, "Inferring Social Network Structure using Mobile Phone Data," *PNAS*, 2007.

[9] V. Lenders, J. Wagner, and M. May, "Measurements from an 802.11b mobile ad hoc network," in *IEEE EXPONWIRELESS*, 2006.

[10] W. J. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *IEEE INFOCOM*, 2007.

[11] V. Conan, J. Leguay, and T. Friedman, "Characterizing pairwise inter-contact patterns in delay tolerant networks," in *ACM Autonomics*, October 2007.

[12] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *IEEE INFOCOM*, 2006.

[13] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnovic, "Power law and exponential decay of inter contact times between mobile devices," in *ACM MobiCom*, 2007.

[14] H. Cai and D. Y. Eun, "Crossing over the bounded domain: from exponential to power-law inter-meeting time in manet," in *ACM MobiCom*, 2007.

[15] Z. J. Haas and T. Small, "A new networking model for biological applications of ad hoc sensor networks." *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, 2006.

[16] R.Groenevelt, G. Koole, and P. Nain, "Message delay in MANET (extended abstract)," in *ACM Sigmetrics*, 2005.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *ACM MOBIHOC*, 2006.

[18] T. Spyropoulos, T. Turletti, and K. Obrazcka, "Routing in delay tolerant networks comprising heterogeneous populations of nodes," *IEEE Transactions on Mobile Computing*, 2009.

[19] M. E. J. Newman, "The structure and function of complex networks," March 2003.

[20] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *ACM MobiHoc*, 2007.

[21] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble Rap: Social-based forwarding in delay tolerant networks," in *ACM MobiHoc*, 2008.

[22] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *ACM MOBIHOC*, 2009.

[23] A. Picu and T. Spyropoulos, "Minimum expected *-cast time in DTNs," in *Bionetics*, 2009.

[24] T. Hossmann, T. Spyropoulos, and F. Legendre, "Know thy neighbor: Towards optimal mapping of contacts to social graphs for DTN routing," in *IEEE Infocom 2010*, March 2010.

[25] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, 2008.

[26] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE MCCR*, July 2003.

[27] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, 2006.

[28] A. Guerrieri, A. Montresor, I. Carreras, F. D. Pellegrini, and D. Miorandi, "Distributed estimation of global parameters in delay-tolerant networks," in *IEEE AOC*, 2009.

[29] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, June 2005.

[30] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[31] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[32] V. Borrel, F. Legendre, M. D. de Amorim, and S. Fdida, "Simps: Using sociology for personal mobility," *ACM/IEEE Transactions on Networking*, Jun 2009.

[33] C. Boldrini, M. Conti, and A. Passarella, "Users mobility models for opportunistic networks: the role of physical locations," in *WRECOM*, 2007.

[34] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J.STAT.MECH.*, 2008.

[35] F. Legendre, V. Borrel, M. D. de Amorim, and S. Fdida, "Revisiting Mobility Modeling," *IEEE Networks*, vol. 00, no. 0, March 2008.

[36] M. Musolesi and C. Mascolo, "A community based mobility model for ad hoc network research," in *ACM REALMAN*, 2006.

[37] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *ACM MobiHoc*, 2003.

[38] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," in *IEEE INFOCOM*, 2006.

[39] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[40] A. Scherrer, P. Borgnat, E. Fleury, J. L. Guillaume, and C. Robardet, "Description and simulation of dynamic mobility networks," *Elsevier Computer Networks*, 2008.

[41] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, 2006.

[42] C. Boldrini, M. Conti, and A. Passarella, "Contentplace: social-aware data dissemination in opportunistic networks," in *ACM MSWiM*, 2008.

[43] L. A. Cutillo, R. Molva, and T. Strufe, "Safebook : a privacy preserving online social network leveraging on real-life trust," *IEEE Communications Magazine*, vol. 42, no. 12, 2009.

[44] M. E. J. Newman, "Analysis of weighted networks," 2004.

[45] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003.

[46] Y. Rabani, A. Sinclair, and R. Wanka, "Local divergence of markov chains and the analysis of iterative load-balancing schemes," in *IEEE Symposium on Foundations of Computer Science (FOCS)*, 1998.

[47] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *IEEE SECON*, 2008.

[48] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *ACM SIGCOMM*, 2007.

[49] V. Conan, J. Leguay, and T. Friedman, "Characterizing pairwise inter-contact patterns in delay tolerant networks," in *ACM Autonomics*, 2007.

[50] D. Aldous and J. Fill, "Reversible markov chains and random walks on graphs. (monograph in preparation.)," *http://stat-www.berkeley.edu/users/aldous/RWG/book.html*.

[51] S. M. Ross, *Stochastic Processes*. Wiley, 1995.

[52] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, 2002.

[53] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant manets," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, 2009.