Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing

Theus Hossmann, Thrasyvoulos Spyropoulos, and Franck Legendre
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
lastname@tik.ee.ethz.ch

***Abstract*—Delay Tolerant Networks (DTN) are networks of self-organizing wireless nodes, where end-to-end connectivity is intermittent. In these networks, forwarding decisions are generally made using locally collected knowledge about node behavior (e.g., past contacts between nodes) to predict future contact opportunities. The use of complex network analysis has been recently suggested to perform this prediction task and improve the performance of DTN routing. Contacts seen in the past are aggregated to a *social* graph, and a variety of metrics (e.g., centrality and similarity) or algorithms (e.g., community detection) have been proposed to assess the utility of a node to deliver a content or bring it closer to the destination.**

In this paper, we argue that it is not so much the choice or sophistication of social metrics and algorithms that bears the most weight on performance, but rather the *mapping* from the mobility process generating contacts to the aggregated social graph. We first study two well-known DTN routing algorithms – SimBet and BubbleRap – that rely on such complex network analysis, and show that their performance heavily depends on how the mapping (contact aggregation) is performed. What is more, for a range of synthetic mobility models and real traces, we show that improved performances (up to a factor of 4 in terms of delivery ratio) are consistently achieved for a relatively narrow range of aggregation levels only, where the aggregated graph most closely reflects the underlying mobility structure. To this end, we propose an online algorithm that uses concepts from unsupervised learning and spectral graph theory to infer this "correct" graph structure; this algorithm allows each node to locally identify and adjust to the optimal operating point, and achieves good performance in all scenarios considered.

I. Introduction

The Delay Tolerant Networking (DTN) paradigm has been proposed to support emerging wireless networking applications, where end-to-end connectivity cannot be assumed for technical reasons (e.g., propagation phenomena, and node mobility) or economical reasons (e.g., lack of infrastructure, low power nodes) [1], [2], [3]. To cope with this, *opportunistic* or *mobility-assisted* routing algorithms have been proposed [4], [5]: messages are forwarded one hop at a time, only when two nodes are in *contact* (i.e., move within transmission range); without full or any knowledge of future contact opportunities, a forwarding decision normally aims to simply increase the delivery probability at every step.

To combat the inherent uncertainty of future contact opportunities, many protocols forward in parallel multiple replicas of the same content [6] or resort to coding (network coding, erasure coding). Nevertheless, node mobility (and resulting contact opportunities) are not entirely random. Instead, weak or strong patterns are present. To this end, numerous *utility-based* routing schemes attempt to differentiate nodes that are more likely to deliver content or bring it closer to the destination [7].

Among them, a number of schemes implicitly assess the strength of ("social") ties between nodes. For example [8] uses time of last encounter, and [9] uses contact frequency as a hint on the *similarity* of mobility patterns. [10], [11] use instead a metric much akin to *degree centrality* to identify nodes that are highly mobile/social; the former scheme is reminiscent of search in scale-free networks [12], while the latter uses centrality to choose which relays to "spray" a limited budget of message replicas to. However, these simple metrics may only capture one facet of the underlying mobility process, which can hinder good contact predictions.

Complex network analysis [13] (CNA) has recently been proposed as a more generic and powerful tool to formulate and solve the problem of future contact prediction in DTNs. Past observed contacts between nodes are *aggregated* into a *social* graph, with graph edges representing (one or more) past meetings between the vertices. An edge in this graph conveys the information that two nodes often encounter each other either because they have a strong social tie (*friends*), or because they are frequently co-located without actually knowing each other (*familiar strangers*); thus, existence of an edge intends to have predictive capacity for future contacts.

Two recently proposed routing protocols, SimBet and BubbleRap [14], [15], make explicit use of CNA metrics and algorithms in order to highlight a node's position in the aggregated social graph, and assess its utility to act as a relay for messages destined to other nodes in the graph. Although the detailed mechanisms of the two protocols differ (see next Section), they are both based on the same principles: they assume that nodes naturally reside in mobility-related communities (e.g., class, work, home). Increasingly "central" or "well-connected" nodes in the graph are then chosen as carriers to relay content over different communities, until a node that shares many neighbors with the destination [14], (i.e., belongs to the destination's community [15], [16]) is reached. These protocols have been reported to often outperform well-known DTN routing schemes that are not explicitly "social".

Nevertheless, it is not well understood under *what* conditions these protocols and their individual components can achieve the suggested performance, nor is it *why*. What is more, it is actually not (just) the choice or sophistication of social metrics or algorithms that bears the most weight on performance, but rather *the mapping from the mobility process generating contacts to the aggregated social graph*. This mapping presents a tradeoff, where some information about

timing of contacts is lost[1]. As a simple example, one could create a link if at least one contact has occurred in the past between the two nodes [14], but this would result in an overly dense graph, after a certain network lifetime. Meaningful differentiation between nodes using complex network analysis will not be possible. Hence, the social graph created out of past contacts should best *reflect* the underlying (mobility or social) structure generating these contacts, so that nodes can be meaningfully differentiated and edges have predictive value.

In this paper, we demonstrate that CNA-based DTN routing can offer significant performance benefits *only* if applied to social graphs exhibiting these properties. Furthermore, we provide an efficient online algorithm to achieve this in a distributed fashion. We summarize below our contributions:

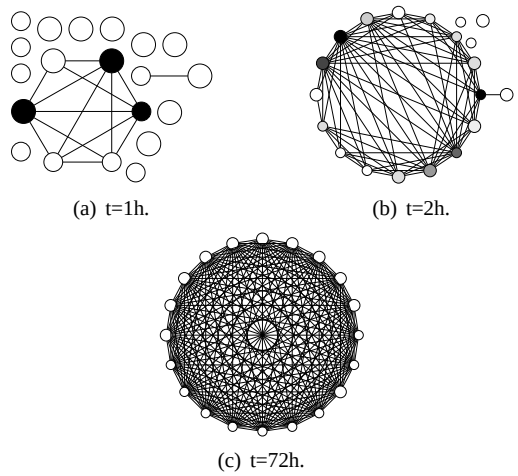
- We evaluate SimBet and BubbleRap under a range of synthetic contact generation models (i.e., Small-World and Caveman) and real mobility traces (i.e., MIT, iMotes Infocom, ETH). We show that good performance is consistently achieved only for a relatively narrow range of aggregation levels, where social graph structure closely reflects the underlying mobility structure (Section III).
- We investigate different methods to identify this optimal operating point "on the fly". Specifically, we use *clustering* techniques [18] to identify desirable patterns in observed node similarities, and then use concepts from *spectral graph theory* [19] to maximize the modularity of such clusters, and compare the behavior of various contact models under different aggregation methods and levels (Section IV).
- We propose a distributed online algorithm that can adjust its contact graph mapping to achieve optimal performance, *regardless of the mobility scenario or the specific routing protocol used* (Section IV).

As a final note, although we focus on unicast routing in this paper, we believe that the observations made and methodology proposed are more widely applicable to most content dissemination algorithms for opportunistic networks.

II. Contact Aggregation: Preliminaries

In this section, we describe our two case study protocols, SimBet and Bubble Rap, in more detail, and formulate the graph aggregation problem.

SimBet [14] assesses *similarity*[2] to detect nodes that are part of the same community, and *betweenness centrality*[3] to identify bridging nodes, that could carry a message from one community to another. The decision to forward a message depends on the similarity and centrality values of the newly encountered node, relative to the current one: If the former node has a higher similarity with the destination, the message is forwarded to it; otherwise, the message stays with the most central node. The goal is to first use increasingly central nodes to carry the message between communities, and then use similarity to "home in" to the destination's community.



(a) t=1h. (b) t=2h. (c) t=72h.

Fig. 1. Aggregated contacts for the ETH trace at different time instants.

Bubble Rap [15] uses a similar approach. Again, *betweenness centrality* is used to find bridging nodes until the content reaches the destination community. Communities here are explicitly identified by a community detection algorithm, instead of implicitly by using similarity. Once in the right community, content is only forwarded to other nodes of that community: a *local centrality* metric is used to find increasingly better relay nodes *within* the community.

A. Time Window Based Aggregation

State-of-the-art algorithms tend to aggregate contacts using a time window.

- **Growing Time Window:** In the original SimBet [14], betweenness and similarity are calculated over a social graph, where there is an edge between two nodes if there has been *at least one* contact between them at *any time* in the past.
- **Sliding Time Window:** A limited time window is used for the two centrality value calculations in Bubble Rap [15], where time is split into 6h *time windows*, and only contacts in the last 6h window form edges of the graph. Yet, this window length is only empirically determined.

Clearly, for both SimBet and Bubble Rap (and CNA-based approaches in general) to function properly, social structures which drive node mobility, such as communities and bridges, must be correctly reflected in the social graph. We argue that this heavily depends on the way this graph is constructed out of observed contacts (*contact aggregation*).

We illustrate this using a real trace of contacts, collected at ETH (see Section III and Table I for details about the trace). Figure 1 shows that an aggregation over the whole history of the network is problematic since the social graph gets more and more meshed. As a consequence, heterogeneity of the nodes, with respect to the above social network metrics, is no longer reflected after long network lifetime. The same holds for aggregation in very short time windows. With nodes shaded according to their betweenness centralities, we see that after a short network lifetime (e.g., after 1 hour) most nodes have the

[1]Other, less compact representations such as *Time Expanded Graphs* [17] have been proposed to include time-related information in a dynamic graph. However, considerable scalability issues quickly arise, as one would essentially need to store a graph for every time instant in the past.

[2]Similarity of two nodes is defined as the number of neighbors these nodes have in common (see e.g., [20]).

[3]Betweenness centrality of a node is defined as the fraction of shortest paths between each possible pair of nodes going through this node (see e.g., [21]).

same color since they did not have any contacts yet and thus their betweenness centrality is not defined (the same holds for similarity). After 2 hours, enough contacts have occurred to differentiate many nodes. However, after 72 hours of running time, all nodes have seen each other and the nodes have again the same betweenness centrality (and similarity). What is more, the time window values at which these transitions occur will differ from scenario to scenario. Consequently, it is easy to see that time window based aggregation can result to forwarding decisions that degenerate to random, significantly affecting the performance of the two protocols, as we shall see in Section III.

B. Density Based Aggregation

Both the choice of how many contacts and the choice of which ones to include in the graph affect its quality. In order for our social graph to be useful for prediction, all edges included must correspond to "regular" contacts (whose past occurrence is predictive of a future occurrence) and none to "random" incidental ones. In that case, there is an *optimal* density for the graph that contains mostly regular nodes. Note that this optimal density depends on the scenario.

We do *not* imply here that random incidental contacts are not useful to the routing process. A random (unpredicted) contact with a node that, e.g., has higher betweenness or belongs to the destination's community can be a very fortuitous event (analogous to the "strength of weak ties" [22]) that a good routing algorithm should seek to exploit. What we do argue for is that the social graph on which such betweenness and communities are calculated should mostly comprise edges corresponding to real, "regular" relationships between vertices.

Let us define a *contact* as the period of time during which two nodes are able to communicate and assume that time is slotted[4]. Let us further denote the ordered sequence of contacts from time 0 to time n as $C_{0,n}$. We can define an aggregation mapping f at time n as

$$f : C_{0,n} \longmapsto G_n(V, E_n).$$

G_n is the output social graph at time n, consisting of all network nodes V, and a subset of edges E_n, among the set of all possible node pairs in the edge set E of the complete graph. We argue that a more useful and robust approach than time-based aggregation, would be to choose the aggregation function such that the resulting social graph has a given *density*. We define the density d of the aggregated graph G_n as the fraction of aggregated edges, $|E_n|$, over all possible edges (i.e., all pairs of nodes) $|E| = \frac{V \cdot (V-1)}{2}$

$$d(G_n) = \frac{|E_n|}{|E|}.$$

If we want to operate the social graph at a certain density, say, $d(G_n) = 0.2$, we choose the "best" edges, according to some criterion, such that E_n will have the desired cardinality. Next, we discuss two methods of picking the edges to fill the graph.

- **Most Recent Contacts (MR)**: In many scenarios, it is reasonable to assume that very old contacts may not have the same predictive power as more recent ones (see e.g., [8]). In that case, each edge $\{u, v\}$ in the graph is labeled with the last time of appearance, timestamp $t_{\{u,v\}}$. Further, a time variable $t_{\text{oldest},n}$ is maintained that keeps track of the oldest edge in G_n. For all contacts $\{u, v\}$ included in the graph, it holds that $t_{\{u,v\}} > t_{\text{oldest},n}$[5].
- **Most Frequent Contacts (MF)**: Another option is to aggregate only the set of most frequent contacts in E_n, since a more frequent contact might reflect a stronger social link. We now maintain, for each pair of nodes, a counter $c_{\{u,v\}}$ indicating how often a contact was seen in the past. Further, the least frequent contact ID in E_n and the respective number of times it was seen $c_{\text{least},n}$ is maintained. If a contact $\{u, v\}$ not included in the graph is observed frequently enough such that $c_{\{u,v\}} > c_{\text{least},n}$, then this new edge is included and the least frequent is deleted (to maintain the chosen density).

It is important to note that a large number of different and more sophisticated mappings are possible, such as weighted graphs [23]. Our goal here is not to derive an optimal aggregation function, but rather to demonstrate, that even with simple aggregation functions, one can considerably influence the performance of DTN routing schemes that utilize complex network analysis.

III. Performance Sensitivity to Aggregation

We first analyze the dependence of performance of CNA-based DTN routing schemes on the parameters of the aggregation mapping (i.e., mapping function and density). Since it is not clear how to *directly* assess the quality of the aggregated social graph, we do so indirectly, by extensively simulating different routing schemes with different aggregation parameters.

A. Network Scenarios

We consider different network scenarios by using five contact generators, two of which are synthetic contact processes and three are real mobility traces. The results from these simulations give strong empirical evidence that our conjecture – that performance heavily depends on how well the underlying network structure is captured by the aggregation – holds for a wide range of different network scenarios.

Synthetic contact processes. Our assumption is that, in most networks of interest, there is some social structure between the participating nodes, often modeled as a graph or *complex network*. In our scenarios, this graph governs the probability of two nodes coming into contact at any time, regardless of the actual mobility model underlying the process (see [24] for an example of a mobility model with a social overlay). As it has been consistently observed (e.g., [25]), such social graphs often exhibit *small-world* behavior. Our two synthetic models described below aim to capture this. The

[4]We consider that each contact lasts one time slot and all messages can be transmitted during this slot as bandwidth issues and contact duration are mostly orthogonal to the type of issues we attempt to expose here. Although contact duration has been proposed as an indicator of link strength, we choose to not consider it here, for simplicity.

[5]Notice that this scheme is similar to the Sliding Time Window, with the difference that the window is now defined directly in the number of past contacts. Although the same question remains, "how to choose the right window value", we believe that the density-based approach offers a more flexible (and natural) way to answer this. Hence, throughout the rest of the paper, we consider MR equivalent to sliding time window.

	MIT	INFO	ETH
Scale and context	97 campus students and staff	41 conference participants	20 lab students and staff
Period	9 months	3 days	5 days
Periodicity	300s (Bluetooth)	120s (Bluetooth)	0.5s (WiFi)
# Contacts			
Total	$100'000$	$22'459$	$23'000$
Per dev.	$1'030$	547	$1'150$

TABLE I
REAL MOBILITY TRACES CHARACTERISTICS.

small world process (**SW**) is inspired by the Watts and Strogatz small world graph model [25]. We number all N nodes sequentially, conceptually arranging them to a ring, and let them have "strong" links to k of their neighbors, i.e. *friends* (e.g., for $k = 4$, node 5 has strong link to nodes $3, 4, 6$ and 7). Then, each of these links is *rewired* with probability p to random nodes outside of the k neighbors to model *familiar strangers*. The resulting graph has strongly connected neighbors, as well as *shortcuts* that capture the small world characteristic [25]. Finally, to generate the sequence of contacts, we select the first node uniformly at random; with probability $1 - q$, we select the peer uniformly at random from the set of its strong links (*friends* and *familiar strangers*) and with probability q, we select its from the set of all nodes (random contacts).

The *caveman* process (**CAVE**) is similar to the SW process, with a different underlying graph model [25]. The N nodes of the network are grouped in cliques (caves) of size k (i.e., $k-1$ neighbors). Thus, unlike the SW model where communities are overlapping, here communities are distinct. Rewiring of some links is used again to create shortcuts, and the next contact is picked among all graph edges similar to the SW model.[6]

Real mobility traces. The MIT *Reality Mining* [26] (**MIT**), the iMotes Infocom 2005 (**INFO**) [3], and the ETH [27] (**ETH**) traces, spanning different network and mobile environments, are used to further support our analysis and findings. Their characteristics are summarized in Table I. Note that in the MIT trace, despite its long duration, a lot of short contacts were supposedly not logged due to its time granularity of 5 minutes. For our simulations we cut the trace at both ends and used $100'000$ contacts reported between September 2004 and March 2005. Note that this time period contains holidays and semester breaks and thus still captures varying user behavior. The ETH trace contains more than $23'000$ reported contacts and is unique in terms of time granularity and reliability. Although its mesurement period spans a considerably shorter time than MIT, we have on average more than 1000 reported contacts per device. This is roughly the number of contacts per device we also have for the MIT trace.

For all three traces, we ignore logged timing information and just order the reported contacts according to their start times (i.e., slotted contacts).

B. Performance at Different Aggregation Densities

We vary the density parameter to investigate how sensitive these protocols are to different aggregations. Using the *most frequent* and the *most recent* aggregation mappings discussed in Section II, we fix the density to values between 0 and 1 (in steps of 0.01). For the density under evaluation, each node creates a message for each other node (i.e., $|E|$ messages). We check how many messages are delivered before the expiry of an empirically determined TTL, chosen such that we get delivery ratios larger than 0.5 for the respective networks[7]. Simulations with different TTLs give qualitatively similar results. As a performance measure, we compute the delivery ratio (DR) of SimBet and Bubble Rap[8], relative to *Direct Transmission* (performance factor thereafter), where the source keeps the message until it meets the destination. Direct Transmission serves as a lower bound on delivery ratio, any smarter scheme should outperform it.

Figures 2(a) and 2(b) show the performance factor of Bubble Rap with SW and CAVE (averaged over 10 simulation runs), and SimBet with ETH and MIT contacts (averaged over 10 different starting times in the trace, at which the messages are created). The first thing to note is that for aggregation densities close to 0 or 1 (i.e., the graph is either empty or complete) the performance factors are 1, that is, the performance is the same as for direct transmission. This has direct implications for the growing time window aggregation, especially if the graph gets complete quickly (see Figure 1). Note that for the MIT contacts, the graph only reaches a density of about 0.4 during the trace duration. We believe that this is due to the large time granularity, which potentially does not capture many short random co-locations.

For SW, CAVE and MIT, we observe clear performance peaks at small densities (around 0.1) and clear performance drops at densities of about 0.2 and higher. For the INFO trace (not shown in the figures) and for ETH, the performance peak is a bit less pointy and at a higher density of around 0.4 for ETH (around 0.65 for INFO). We suspect the reason for this is that both traces are smaller than MIT in terms of nodes and geographical extent.

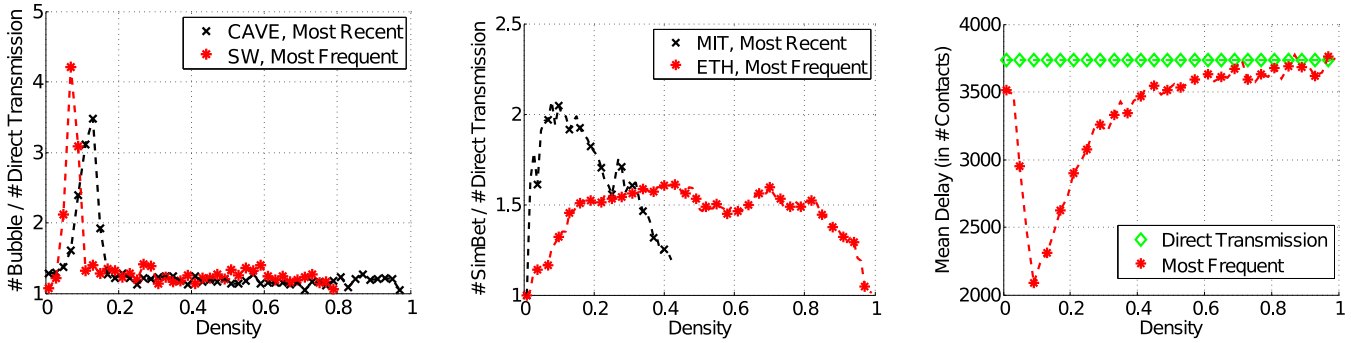
Table II summarizes the results of all combinations of the two protocols with the two mapping functions and with the five contact processes, in terms of density at which the peak performance occurs and height of the peak performance factor. One important observation from the table is that the peak densities differ from scenario to scenario, but are consistent within a scenario (e.g., peak densities for SW are all along the same value). *This indicates that the optimal point of aggregation does not depend on the forwarding metric, but rather on the contact process.* One exception is the ETH trace, where the peak density for SimBet with MR mapping is at a quite smaller value. One might deduce from this that the few very recent contacts have much more predictive power than slightly older ones. We plan to investigate this further in future work.

Another observation from the table is, that the performance peaks of SimBet and Bubble Rap are of similar height within a scenario. However, the peak height differs from scenario to

[6]Note that for both SW and CAVE, draws are with replacement meaning that an existing contact can be picked again. For both synthetic processes, we set the parameters to $k = 10$, $p = 0.1$ and $q = 0.5$, to match the properties of the contact traces. $N = 100$ in both scenarios.

[7]The TTL values are 2000 for CAVE and SW. 5000 for MIT, 1500 for INFO and 250 for ETH

[8]We use a slight modification of the Bubble Rap protocol: In order to compare it to SimBet, we operate Bubble Rap as a single-copy instead of a multi-copy protocol.



(a) Bubble Rap DR performance increase (b) SimBet DR performance increase (c) SimBet Delivery Delay (CAVE)

Fig. 2. Delivered messages and delivery delay for SimBet/Bubble Rap vs. Direct Transmission.

Protocol	SW	CAVE	MIT	ETH	INFO
SimBet MF	.09/4.3	.09/3.3	.03/1.8	.43/1.6	.64/1.3
SimBet MR	.09/3.7	.12/3.5	.09/2.1	.02/1.5	.71/1.4
Bubble MF	.08/4.2	.09/4.5	.08/2.5	.44/1.5	.67/1.4
Bubble MR	.09/3.3	.09/3.9	.14/2.5	.33/1.4	.67/1.4

TABLE II
OPTIMAL AGGREGATION DENSITIES. THE FIRST NUMBER IS THE DENSITY, THE SECOND NUMBER IS THE PERFORMANCE INCREASE FACTOR. MF = MOST FREQUENT, MR = MOST RECENT.

scenario. This indicates that different networks show different degrees of structure, which the routing protocols can use to increase their performance compared to direct transmission. Also, the peak heights depend on the TTL we set for the messages. Our analysis shows that higher TTL flattens the peaks, since the larger delay of direct transmission in that case is reflected to a lesser degree in the performance factor. In order to have results independent of TTL, we analyzed the delivery delay. Figure 2(c) shows results for SimBet with CAVE contacts. It shows similar properties, i.e., low delays compared to the baseline, at only a narrow range of small densities.

We have also evaluated the protocols' CNA components (i.e., similarity, community detection, centrality) individually, with respect to graph density, and observed similar sensitivity to density. We omit these plots due to space limitations.

IV. Inferring the Optimal Aggregation Density Using an Online Algorithm

There is clearly an optimal density (or rather a narrow range of densities) for the aggregated social graph, outside of which the performance of CNA-based DTN routing protocols significantly degrades. Although density-based aggregation can directly operate the social graph around a given density point, an interesting question arises: *How can nodes find an optimal aggregation density in real time, without any prior knowledge (e.g., mobility context) and using local information only, to optimize their performance?*

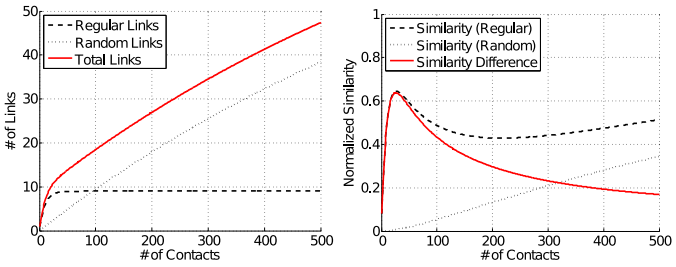
We conjecture that this optimal density lies at the point where the underlying (social) structure guiding mobility best correlates with the social structure that can be observed on the aggregated contact graph. In other words, a strong "relationship" in the contact process (e.g., a community/regular link in SW or CAVE) should also be an observable relationship (edge) in the aggregated graph; at the same time, the number of additional graph edges (e.g., random links) should be as few as possible, as these will most probably be the result of incidental past meetings. There is solid evidence that real life mobility is predominantly small-world, with salient features such as communities [16] and a high clustering coefficient [28], even though the exact *actual* structure is still to be investigated further. We hence believe that the above observation is accurate for the mobility traces as well, something that is further supported by our trace-related results.

The goal of this section is to devise an algorithm that observes the aggregated social graph *online* (i.e., as new contacts arrive), and tries to assess the density at which this graph has a structure that best reflects the above intuition about connectivity/contact *patterns*. This is a difficult *unsupervised learning* problem. We first give hints on how to tackle this problem through an analytical treatment of the CAVE model. Then, we propose two clustering-related methods of identifying distinguishable similarity patterns: one based on spectral graph theory [19] and the other based on established methods of evaluating the quality of graph partitioning [29]. Both can be used at the core of our algorithm. Eventually, we evaluate the performance of the DTN routing protocols operating with a graph obtained from our online algorithm, and show that they perform as well as with the optimal density found in Section III (see Table II).

A. Optimal community structure of the social graph

One way to distinguish regular neighbors from random neighbors is by their similarity values. *Each node will see a set of nodes to which it is highly similar (i.e., many shared regular neighbors in the graph) and another set of nodes to which it is less similar (i.e., random neighbors).* Our problem can now be cast as maximizing the similarity to regular neighbors, while minimizing the similarity to random ones. In order to motivate this approach, we use the CAVE model (without rewiring, see also Section III), to derive the expected number of regular neighbors and the expected number of random neighbors of a node in the aggregated graph. From these, we get the expected values of similarity that would be observed online, as a function of time. Note that the same methodology can be applied to CAVE *with* rewiring and to the SW model.

Expected Number of Regular Links: Let $n_{reg}(c)$ denote the number of *regular* links of a node in the aggregated social graph, after c contacts with other nodes. Then, the expected



(a) # Links vs. # Contacts (b) Similarity vs. # Contacts

Fig. 3. Number of links and similarity values as a function of time.

number of regular links is:

$$E[n_{reg}(c)] = (k-1)\cdot\left(1-\left(1-\frac{1-q}{k-1}\right)^c\right). \quad (1)$$

$1-q$ is the probability that the contact is regular – see Section III – and there are $k-1$ regular links for a node, thus, the probability of a contact to be a specific node u is $\frac{1-q}{k-1}$. The probability that we observe at least one contact to u is one minus the probability that we do not see the node, thus, $1-\left(1-\frac{1-q}{k-1}\right)^c$. This is valid for each of the $k-1$ nodes independently, so, the expected number of filled regular links after c contacts is given by Eq. 1. Clearly, this number is bounded above by $k-1$.

Expected Number of Random Links: The same way we can easily derive the expected number of random links of a node after c contacts, $E[n_{rnd}(c)]$, as

$$E[n_{rnd}(c)] = (N-k)\cdot\left(1-\left(1-\frac{q}{N-k}\right)^c\right). \quad (2)$$

In Figure 3(a), we plot the expected number of regular, random, and total links as a function of contacts seen by a specific node (this is an implicit measure of time as well) for the parameters $N=100, k=10, q=0.1$. This figure shows that the point when all (or most) regular links are in the graph is reached quickly. In this scenario, this is the right point to stop the aggregation and fix the density. After that point, only random links are added to the graph. These have little predictive value and erroneously increase nodes' similarity. Next, we derive expected similarity values explicitly. Denote the similarity of a node u to an encountered node v as $sim(u,v) = |N(u) \cap N(v)|$, where $N(u)$ is the set of neighbors of node u.

Expected Similarity of Regular Neighbors: Let $E[sim_{reg}(c)]$ denote the expected similarity (number of common neighbors) of a node u to a *regular* neighbor v, after it has experienced c contacts with other nodes. Such common neighbors can be from: (i) the $k-2$ remaining nodes of their community, or (ii), the $N-k$ nodes of all other communities. We denote the respective numbers as $E[sim_{reg-reg}(c)]$ and $E[sim_{reg-rnd}(c)]$.

$$E[sim_{reg}(c)] = E[sim_{reg-reg}(c)] + E[sim_{reg-rnd}(c)]. \quad (3)$$

Each of the remaining $k-2$ nodes of the respective community are independently of each other in the neighbor set of u and v, thus,

$$E[sim_{reg-reg}(c)] = (k-2)\cdot\left(1-\left(1-\frac{1-q}{k-1}\right)^c\right)^2. \quad (4)$$

This similarity quantity has predictive value, as the existence of the edges accounted for in it, implies that a future contact between the vertices at the ends of each of these edges is quite probable.

The expected number of common *random* contacts is

$$E[sim_{reg-rnd}(c)] = (N-k)\left(1-\left(1-\frac{q}{N-k}\right)^c\right)^2. \quad (5)$$

Expected Similarity of Random Neighbors: If u and v are not in the same community, we have to distinguish the following two cases: (i) random contacts of node v happen to occur with regular neighbors of node u and vice versa; we denote the number of such common neighbors as $E[sim_{rnd-reg}(c)]$; (ii) random contacts of node u happen to coincide with random contacts of node v, denoted as $E[sim_{rnd-rnd}(c)]$. Then,

$$E[sim_{rnd-reg}(c)] =$$
$$2(k-1)\left(1-\left(1-\frac{1-q}{k-1}\right)^c\right)\left(1-\left(1-\frac{q}{N-k}\right)^c\right),$$

and,

$$E[sim_{rnd-rnd}(c)] = (N-2k)\left(1-\left(1-\frac{q}{N-2k}\right)^c\right)^2.$$

In total, the expected similarity of random neighbors is

$$E[sim_{rnd}(c)] = E[sim_{rnd-reg}(c)] + E[sim_{rnd-rnd}(c)]. \quad (6)$$

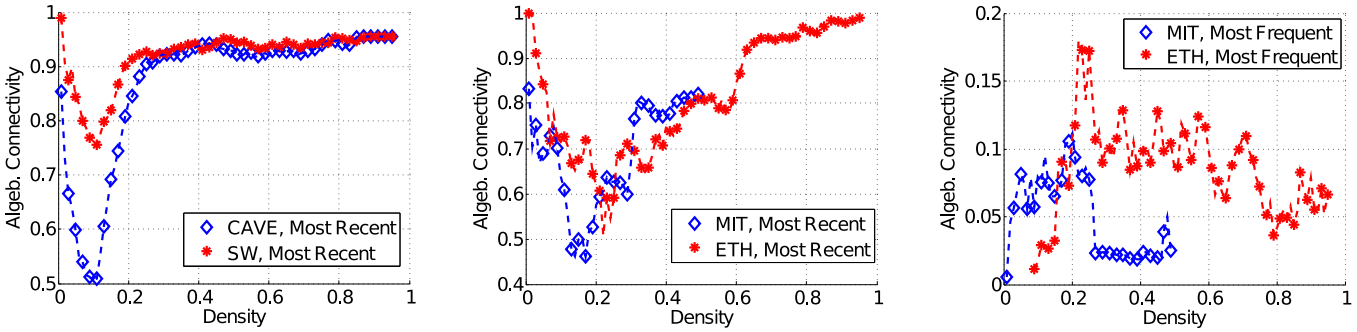
This similarity quantity has little predictive value, as it accounts for incidental contacts that do not imply anything about the same contact re-occurring.

To maximize the predictive capacity of the aggregated graph we want to maximize $E[sim_{reg}(c)]$, while at the same time minimizing $E[sim_{rnd}(c)]$. In practice, nodes do not have any a priori knowledge about community membership and size, or the total number of nodes. Thus, it is more sensible to use normalized similarities and divide similarities by the expected node degree $E[n_{reg}(c)] + E[n_{rnd}(c)]$. Consequently, in order for our algorithm to automatically adjust the aggregation window to the optimal aggregation density, it seems reasonable to solve the following maximization problem:

$$\underset{c}{\text{maximize}}\left(\frac{E[sim_{reg}(c)] - E[sim_{rnd}(c)]}{E[n_{reg}(c)] + E[n_{rnd}(c)]}\right). \quad (7)$$

Figure 3(b) depicts the normalized similarities and their difference. It is clear from this plot, that the difference is maximized around the point where enough contacts per node have occurred to fill in most regular links, but only few random links have been instantiated. We will show later that this maximum correlates well with the aggregation density at which the performance of SimBet and Bubble Rap is optimal (see Section III).

Nevertheless, regular similarity $E[sim_{reg}(c)]$ is not directly observable without knowing which links are regular and which are not. In fact, in order to apply this maximization, nodes need to first be able to distinguish between these two classes of links. When a node encounters another, it only knows and logs down its similarity value to this node. Out of the contacts observed over time, it can create a histogram of similarity values observed. One way to assign "labels" to



(a) Algebraic Connectivity (Models) (b) Algebraic Connectivity (Traces) (c) Q Function (Traces)

Fig. 4. Modularity metrics of similarity values depending on the density of the aggregated graph.

each past contact ("regular" vs. "random") is to perform *clustering* on the set of normalized similarity values. If the aggregated density is appropriate, a *2-means* [18] algorithm should produce two clusters of *similar* normalized similarity values, one for similar regular nodes and another for similar random nodes. Note, however, that at low (close to 0) and high densities (close to 1) only one cluster appears since all nodes have *similar* similarities close to 0 (no similarity) and 1 (all nodes similar), respectively. Eq. (7) then becomes equivalent to maximizing the distance between the two cluster centroids of low and high densities.

B. Robust Clustering of Similarity Values

Although the above discussion provides useful intuition, in practice, the structure guiding mobility (and thus contacts) cannot be captured in a straightforward way using the 2-means clustering approach based on normalized similarity. First, as explained earlier, there might not be 2 clear cluster centers at low and high densities, in which case the result returned by the simple 2-means approach is unreliable. Second, the real world traces, albeit sharing basic structure with the synthetic models, exhibit more heterogeneity: nodes might belong to more than one community, degree distributions might be skewed, communities might be overlapping, and the line between regular and random contacts becomes blurred. As a result, the two classes of similarities might not be easily distinguishable even around the "right" aggregation point. Thus, the simple clustering algorithm sketched above, based only on cluster center distance, might draw deceiving conclusions.

To cope with the difficulty of identifying a contact as random or regular, we use two approaches to assess how distinguishable the two clusters returned by the clustering algorithm are. One is *spectral analysis* of a pre-processed similarity matrix and the study of the matrix' *algebraic connectivity* [19]. The second is the *Q function* of cluster modularity that has been found to correlate with cluster quality in many examples [29].

Spectral Analysis: Let us assume that a node u has collected a set of n contacts c_i ($i = 1 \ldots n$) during a time period (according to one of the methods in Section II). Node u uses these contacts to build its view of the social graph. Each contact c_i observed is assigned a real number s_i measuring the *normalized similarity* between u and the node encountered v:

$$s_i = \frac{|N(u) \cap N(v)|}{min\{|N(u)|, |N(v)|\}} \quad (s_i \in [0,1]), \tag{8}$$

where $N(x)$ is the set of neighbors of node x in the aggregated social graph. In other words, each node has now a vector $\mathbf{s}$ of n real-valued entries in $[0,1]$ representing the various similarity values observed thus far. According to the previous discussion, when the right amount of contacts (n) has been observed, the values in this vector should cluster around small and high values.

In order to formally measure this spectral clustering [19] converts the vector $\mathbf{s}$ into an $n \times n$ *affinity matrix* $\mathbf{W}$.

$$\begin{aligned} \mathbf{W} &= \{w_{ij}\}, \\ w_{ij} &= exp(-\frac{\|s_i - s_j\|^2}{2\sigma^2}), \text{if } i \neq j \text{ , and } w_{ii} = 1, \end{aligned} \tag{9}$$

with $\sigma \in [0,1]$ (threshold value).
Let us further define the *Laplacian* of $\mathbf{W}$ as

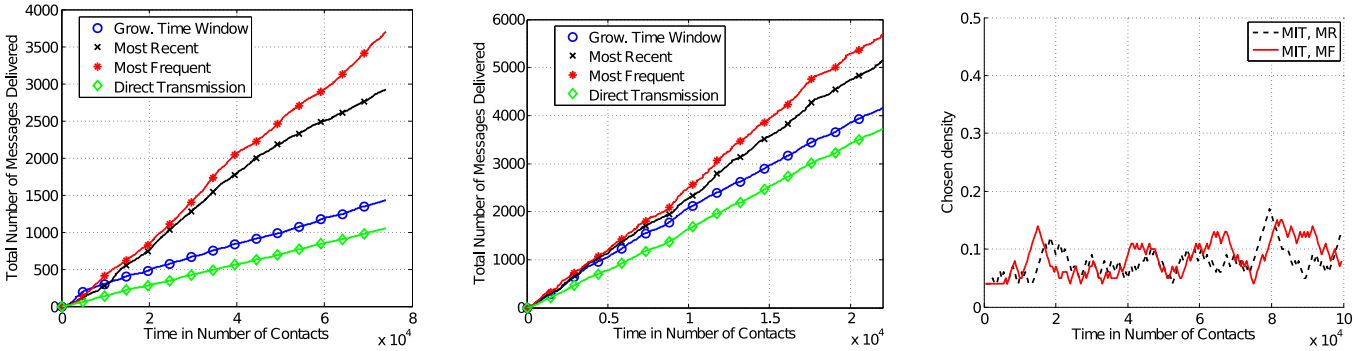
$$L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \tag{10}$$

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 $\mathbf{W}$). *Spectral Graph Theory* studies the structural properties and invariants of the weighted graph defined by $\mathbf{W}$, using eigenvalue decomposition of the Laplacian $\mathbf{L}$. Spectral clustering uses this theory to identify k strongly connected components in W with few weak links between them, by projecting the n points into the eigenspace of L consisting of L's first k eigenvectors. In fact, spectral clustering methods solve a relaxation of the *normalized cut* problem (a known NP-hard problem), which is a min cut problem under the constraint that the two cut-sets are balanced.

In the ideal case where $\mathbf{W}$ is block-diagonal, that is, it consists of k connected components with all weights 0 between blocks, the eigenvalues $\lambda_i, i = 1 \ldots n$ of $\mathbf{L}$ are:

$$\lambda_1 = \cdots = \lambda_k = 0 < \lambda_{k+1} \cdots \leq \lambda_n. \tag{11}$$

This means, it has exactly k eigenvalues equal to 0, and all other eigenvalues bounded away from and larger than 0. In that case, spectral clustering algorithms are guaranteed to identify clusters correctly even when they form non-convex (non-linearly separable) regions. In the non-ideal case, $\mathbf{W}$ is not



(a) Bubble Rap with CAVE contact trace. (b) SimBet with ETH contact trace. (c) Density chosen by the adaptive algorithm.

Fig. 5. SimBet and Bubble Rap performance with different aggregation functions compared to direct transmission.

block diagonal (i.e., is connected with lower weights between clusters), and only the first eigenvalue of **L** is 0. Nevertheless, matrix perturbation theory [30] suggests 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.

This is the basis of our algorithm. We expect to see either of two things: Either two separable (even if noisy) similarity-value clusters in the optimal density range, or only one (or two non-easily separable ones) if density is too high or too low. Our algorithm then seeks to locally minimize the second eigenvalue λ_2 of the Laplacian (known as the *Algebraic Connecitivty*) of the similarity vector **s** observed over time.

Modularity Function Q: A different approach often used for evaluating community structure in complex networks is the use of an appropriate *modularity* function Q [29]:

$$Q(\mathcal{P}_k) = \sum_{c=1}^{k} \left[\frac{\mathcal{A}(V_c, V_c)}{\mathcal{A}(V, V)} - \left(\frac{\mathcal{A}(V_c, V)}{\mathcal{A}(V, V)} \right)^2 \right], \qquad (12)$$

where $\mathcal{P}_k$ is a partition of the vertices into k groups and where $\mathcal{A}(V', V'') = \sum_{i \in V', j \in V''} w_{ij}$, with $\mathbf{W} = \{w_{ij}\}$ defined as above. This function attempts to evaluate a particular graph partitioning by measuring the ratios of intra-community links to inter-community links. The more the intra-community link occurrence diverges from what one would expect for a random network, the more clearly separable the communities, and the higher the value of Q.

Although the modularity function approach and the spectral clustering approach are not completely different (in fact, spectral methods can be used to maximize Q [31]), these have particular strengths and weaknesses (as we have also observed for our datasets), as well as differences in implementation overhead. For this reason, we are going to present results for both approaches and compare their performance in various scenarios.

Online optimal density tracking algorithm: The idea is to find the density at which the *Algebraic Connectivity* of observed similarity values is minimal, or alternatively, the Q function is maximal. While collecting and logging n contacts[9], we compute similarity values, using the aggregated graph at different densities. After n contacts, we use these similarities to determine the gradient of the Algebraic Connectivity (or Q Function) and adapt the density of operation accordingly, towards the optimum. In order not to get stuck in local extrema, we use occasional random lookaheads.

[9]We set the update interval n empirically to 100 contacts, as a tradeoff between having enough similarity samples, and reacting sufficiently fast to changes in the network.

Protocol	SW	CAVE	MIT	ETH	INFO
SimBet MF	4.1/3.3	3.0/3.0	1.8/1.7	1.5/1.5	1.2/1.2
SimBet MR	3.6/3.2	2.8/2.5	2.1/2.0	1.4/1.5	1.3/1.2
Bubble MF	2.9/2.7	3.6/3.8	2.1/1.7	1.5/1.5	1.3/1.2
Bubble MR	3.2/3.3	3.4/3.2	1.8/1.3	1.4/1.4	1.1/1.2

TABLE III
PERFORMANCE FACTORS USING THE DENSITY ADAPTION ALGORITHM. THE FIRST NUMBER IS FOR ALGEBRAIC CONNECTIVITY, THE SECOND NUMBER FOR THE MOST Q MODULARITY.

C. Performance Evaluation

Using this simple tracking algorithm, we evaluate how close we get to the optimal performance (see Section III). Figure 5(a) shows the cumulative number of messages Bubble Rap delivers, when we create messages between random sender and destination throughout the simulation. We choose the density with the online algorithm using the Algebraic Connectivity metric, and compare it to the original offline algorithm with growing time window and to direct transmission. We observe that after a short initial period, the growing time window graph becomes too dense to be useful, and the performance degrades compared to that of direct transmission (i.e., the two lines are parallel). On the other hand, our online algorithm keeps the good performance throughout the simulation with both – the most recent and most frequent – mappings. Most frequent performs a bit better in this case, as community links are more frequent than random ones in the CAVE model by design.

A similar result is shown in Figure 5(b), where we use SimBet and the ETH trace with the same simulation setup. However, in this case the performance increase is a bit smaller, since the trace is less structured, as reported in Section III. Figure 5(c) shows the evolution of the chosen density by our online algorithm in the course of trace time (MIT). Our algorithm moves around a density of 0.09, which indeed coincides with the best performance benefit calculated offline

(cf. Table II). We note here that, in the case of the traces, we have also observed some small variance in the density value that achieves the best performance at different time slices of the trace, indicating some non-stationarity in the process. Nonetheless, our online algorithm seems to be capable, in most cases, to track a close to optimal density value. We intend to look deeper into such trace properties in future work.

Table III summarizes the performance factor for all combinations of protocols, aggregation functions, and contact processes. Most values are close to the ones of Section III. In some cases, for instance SimBet with the MIT trace, we even reach the reported optimal values. Note also, that the performance of the Algebraic Connectivity and the Q Function versions are very similar. In some cases, Algebraic Connectivity is slightly better. However, the computation of the Q values is significantly less complex (i.e., less similarity preprocessing). The choice between the two is thus a tradeoff between complexity and a slightly better performance.

V. Conclusions

In this paper, we have established the predominant importance of efficient mappings of mobility contacts to an aggregated social graph, which DTN algorithms using complex network analysis (CNA), can utilize to optimize forwarding decisions. Specifically, this aggregated social graph exhibits an optimal density where it best reflects the underlying social mobility and where performance benefits are maximized. Contrary to this, specific metrics and algorithms (e.g., for community detection, etc.) used by different CNA-based schemes seem to have a less prominent effect on performance. Finally, by mapping the problem to that of unsupervised clustering of observed node similarity values (online), we have shown that methods based on algebraic connectivity and cluster modularity can capture this optimal point in a robust manner both for synthetic models and real world traces. Using an algorithm based on these methods we can track this optimal point and achieve closed to offline performance, without prior knowledge. We believe that our preliminary findings and proposed solutions have a wider applicability for a large range of DTN data dissemination protocols based on social networks.

In future work, we intend to look deeper into the various traces, as well as into the connectivity graphs resulting from state-of-the-art, trace-based mobility models (e.g., [28]), in order to better understand their underlying structure and similarities. We intend to use both traditional graph metrics (e.g., degree distribution, etc.) as well as spectral graph theory to uncover interesting invariants. Furthermore, we are interested in exploring more sophisticated mappings, e.g., appropriate weighted graphs, as these can potentially capture more information and spectral methods are still applicable. Finally, we are interested in the scaling behavior of such CNA-based approaches. Specifically, although these schemes have demonstrated some performance benefits, we would like to investigate the navigability properties of the respective contact graphs, and more importantly, whether these properties can indeed result in efficient and scalable DTN routing solutions as network size increases.

Acknowledgment

This work was funded by the ANA project (EU FP6-IST-27489).

References

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," in *ACM ASPLOS-X*, 2002.
[2] P. Basu and T. D. C. Little, "Networked parking spaces: architecture and applications," in *IEEE VTC*, 2002.
[3] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *ACM WDTN*, 2005.
[4] E. P. Jones and P. A. Ward, "Routing strategies for delay-tolerant networks," *Submitted to ACM CCR*, 2006.
[5] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, November 2006.
[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *ACM WDTN*, 2005.
[7] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, July 2003.
[8] 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.
[9] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, "Diversity of forwarding paths in pocket switched networks," in *IMC*, 2007.
[10] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *ACM MobiHoc*, 2008.
[11] T. Spyropoulos, T. Turletti, and K. Obrazcka, "Routing in delay tolerant networks comprising heterogeneous populations of nodes," *IEEE TMC*, 2009.
[12] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks," *Phys. Rev. E*, vol. 64, no. 4, Sep 2001.
[13] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, March 2003.
[14] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *ACM MobiHoc*, 2007.
[15] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *ACM MobiHoc*, 2008.
[16] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed Community Detection in Delay Tolerant Networks," in *ACM MobiArch*, 2007.
[17] L. Ford and D. Fulkerson, *Flows in networks*. Princeton University Press, 1962.
[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.
[19] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.
[20] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.
[21] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, 1979.
[22] M. Granovetter, "The strength of weak ties: A network theory revisited," *Sociological Theory*, vol. 1, pp. 201–233, 1983.
[23] M. E. J. Newman, "Analysis of weighted networks," *Physical Review E*, vol. 70, p. 056131, 2004.
[24] M. Musolesi, S. Hailes, and C. Mascolo, "An ad hoc mobility model founded on social network theory," in *ACM MSWiM*, 2004.
[25] D. J. Watts, *Small Worlds : The Dynamics of Networks between Order and Randomness*. Princeton University Press, November 2003.
[26] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, May 2006.
[27] "Measurements from an 802.11b mobile ad hoc network," in *IEEE ExponWireless*, 2006.
[28] A. Scherrer, P. Borgnat, E. Fleury, J. L. Guillaume, and C. Robardet, "Description and simulation of dynamic mobility networks," *Elsevier Computer Networks*, vol. 52, no. 15, 2008.
[29] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, p. 026113, 2004.
[30] 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, pp. 849–856.
[31] S. White and P. Smyth, "A spectral clustering approach to finding communities in graph." in *SDM*, 2005.