From Contacts to Graphs: Pitfalls in Using Complex Network Analysis for DTN Routing

Theus Hossmann, Franck Legendre, and Thrasyvoulos Spyropoulos
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 made using locally collected knowledge about node behavior (e.g., past contacts between nodes) to predict which nodes are likely to deliver a content or bring it closer to the destination. One promising way of predicting future contact opportunities is to aggregate contacts seen in the past to a *social* graph and use metrics from complex network analysis (e.g., centrality and similarity) to assess the utility of a node to carry a piece of content. This aggregation presents an inherent tradeoff between the amount of time-related information lost during this mapping and the predictive capability of complex network analysis in this context. In this paper, we use two recent DTN routing algorithms that rely on such complex network analysis, to show that contact aggregation significantly affects the performance of these protocols. We then propose simple contact mapping algorithms that demonstrate improved performance up to a factor of 4 in delivery ratio, and robustness to various connectivity scenarios for both protocols.**

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 (e.g., propagation phenomena, and node mobility) or economical reasons (e.g., low power nodes) [1], [2], [3]. Due to the lack of end-to-end paths, traditional routing protocols perform poorly, and numerous *opportunistic* routing algorithms have been proposed instead [4], [5], [6], [7], [8], [9]. There, multiple replicas of the same content are often routed in parallel to combat the inherent uncertainty of future communication opportunities between nodes [4]. In order to carefully use the available resources (i.e., limit the number of content copies in the network) and still get a short delay, many protocols attempt to predict which nodes are likely to deliver content or bring it closer to the destination [6].

In most of the networks discussed, node mobility (and resulting communication opportunities) is not entirely random. Instead, weak or strong patterns are present, which a node can attempt to infer and use to predict future contact opportunities. To this end, numerous "utility-based" routing schemes have been proposed, where various contact properties such as time of last encounter between two nodes [5], frequency of past encounters [6], and mobility profiles [7] are maintained and analyzed to assess the probability of a given node to get closer to the destination. However, these schemes mostly rely on heuristics that may or may not correlate well with future delivery probability, depending on the scenario in hand.

Wireless devices (PDAs, cellphones, and laptops) are often carried by humans and communicate when in close proximity (e.g., PodNet [10], pocket switched networks [3]). Consequently, the social interactions of the carrying users will directly affect the communication patterns between the devices. Based on this observation, complex network analysis [11] has recently been proposed as an appropriate 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. A rich toolset from complex network analysis can then be used to assess the utility of a node to carry a piece of content (e.g., by considering its centrality) [8], [9].

A social graph offers a natural, compact representation of the resulting contact set over time. A graph link could mean that two nodes see each other frequently because they have a social connection (*friends*), or because they are frequently in the same place without actually knowing each other (*familiar strangers*); hence, a link is intended to have predictive value for future contacts. Nevertheless, the *aggregation* of contacts between nodes over time into a "static" social graph presents an inherent mapping tradeoff, where some information about timing of contacts is lost[1]. One could create a link if at least one contact has occurred in the past between the two nodes [8], but this would result in an overly dense graph, after a certain network lifetime. On the other hand, a past contact could represent a link only if occurred during a given time window [9]. However, if the sliding time window is too small, the resulting graph might be too sparse. In both cases, meaningful differentiation between nodes using complex network analysis may be rendered impossible. It is thus important to carefully design this mapping in order for links in the graph to maintain their predictive value.

In this paper, we address the issue of efficient contact aggregation. Section II presents related work focusing on complex network based forwarding schemes. In Section III, we describe and formalize the contact aggregation as a mapping

[1]*Time Expanded Graphs* [12] have been proposed to capture timing 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.

problem, and propose various mappings. Finally, in Section IV, we demonstrate that mappings used currently in popular DTN routing schemes may result in significantly worse performance than predicted; then, we show that alternate mappings of similar complexity can considerably improve the performance of the analyzed forwarding schemes (up to a factor of 4 for the delivery ratio). Our evaluation is based on simulations relying on synthetic graph models (i.e., Scale-Free and Small-World) and real-world traces (i.e., MIT, ETH).

II. Related Work

A number of proposed DTN forwarding schemes rely implicitly on assessing the strength of social connections between nodes. In [5], the time elapsed since the last encounter with a node is taken as a predictive value for future contacts. Along the same line, [13] uses the contact frequency as a utility metric. This is an inherently social metric since nodes with strong connections will see each other frequently. In [14], the minimum estimated expected delay for each link is inferred from contacts observed in the past. Similarly, [6] defines a delivery predictability based on how frequently two nodes meet. Finally, a metric much akin to *degree centrality* is used in [15] to identify candidate relays in spray routing.

Two recently proposed forwarding schemes, SimBet [8] and Bubble Rap [9], use the social structure of the network more explicitly and apply complex network analysis to assess the utility of a node for forwarding. Both are based on the idea that nodes (or rather the users carrying the devices) are clustered to communities of highly connected nodes, and some nodes form bridges between such communities. Further, they assume that this structure will be reflected in the social graph they construct. Although preliminary results for SimBet and Bubble Rap demonstrate promising performance [8], [9], we will show that this performance heavily depends on the way contact aggregation is done. Below, we give a brief description of the two protocols. They will serve as our case studies.

SimBet [8] 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. In the original algorithm [8], 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.

[2]Similarity of two nodes is defined as the number of neighbors these nodes have in common (see e.g., [16]).

[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., [17]).

Bubble Rap [9] 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. Regarding contact aggregation, it is performed at two points. First, for community detection where, as in SimBet, *all* contacts in the past are considered. Second, for the two centrality values the time is split into 6h *time windows*, instead. All contacts in such a 6h window form edges of the graph.

III. Contact Aggregation

For both SimBet and Bubble Rap to function properly, social structures which drive node mobility, such as communities and bridges, must be correctly reflected in the social graph. Here, we argue that this heavily depends on the way this graph is constructed out of observed contacts (i.e., contact aggregation). We illustrate and define the contact aggregation problem and propose simple aggregation mappings.

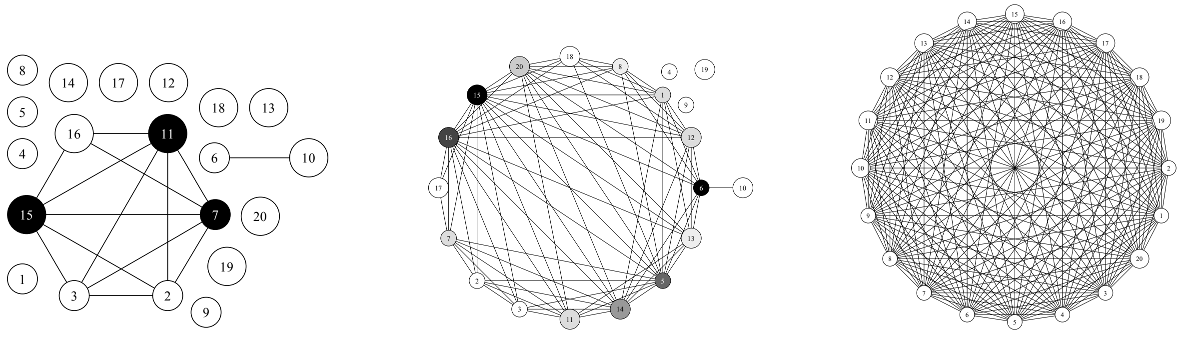
A. Contact Aggregation Problem

Using a real trace of node contacts, collected at ETH (see Section IV-A), 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 in Figure 1 that after a short network lifetime (e.g., after 1 hour) most nodes have the same color since they did not have any contacts yet and thus their betweenness centrality is not defined. 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. In this case, forwarding decisions gradually degenerate to random, significantly affecting the performance of the two protocols, as we shall see in Section IV.

In order to formalize the problem of contact aggregation we define a *contact* as the period of time during which two nodes are able to communicate and assume that time is slotted. Further, without loss of generality, we consider that each contact lasts one time slot and all messages can be transmitted during this slot[4]. If we denote the ordered sequence of contacts from time 0 to time n as $C_{0,n}$, then we can define an aggregation mapping f at time n as

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

[4]Bandwidth issues and contact duration are mostly orthogonal to the type of issues we attempt to expose here. Although contact duration has been in some cases used as an indicator of link strength, we choose to not consider it here, for simplicity.



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

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

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 $e_{u,v}$ in the edge set E of the complete graph. Therefore, a mapping can be written as an indicator function for each edge $\mathbb{1}_{e_{u,v}} : E \longmapsto \{0,1\}$, indicating whether this edge is part of the subset E_n under the particular mapping or not.

B. Contact Aggregation Mappings

There are different ways to define this indicator function, resulting in different aggregation mappings.

Growing Time Window: As we saw in Section II, some state-of-the-art algorithms [8] use what we will refer to as *growing time window* mapping, that is,

$$\mathbb{1}_{e_{u,v}} = \begin{cases} 1 & \text{, if } \{u,v\} \in C_{0,n} \\ 0 & \text{, otherwise.} \end{cases} \tag{1}$$

As explained earlier, the problem with this aggregation is that for large n the graph gets fully meshed with all nodes appearing equal with respect to social metrics (e.g., Figure 1.c).

Sliding Time Window: A generalization of the *time window* aggregation is to aggregate over a fixed time window instead of the whole lifetime of the network. Denoting the time window length as T, we write

$$\mathbb{1}_{e_{u,v}} = \begin{cases} 1 & \text{, if } \{u,v\} \in C_{n-T,n} \\ 0 & \text{, otherwise.} \end{cases} \tag{2}$$

The crucial question here is how large the fixed time window should be for an optimal aggregation. For example, Bubble Rap uses a 6h window for the centrality aggregation.

A fixed size time window improves the situation. However, it can have different implications depending on the scenario in hand. In a smaller network, such as in an office or conference, the graph can be fully meshed after few days or hours already, as shown by Figure 1. On the other hand, in a larger network such as a campus or a city-wide scenario, a window of few hours may result in a very sparse graph.

Growing and sliding (fixed duration) time window are the aggregation functions used in one way or another in current protocols. We argue that a more general and robust approach would be to choose the aggregation function such that the resulting social graph has a constant *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 edge indicator function such that E_n will have the desired cardinality. In the following, we discuss two indicator functions that fulfill this condition.

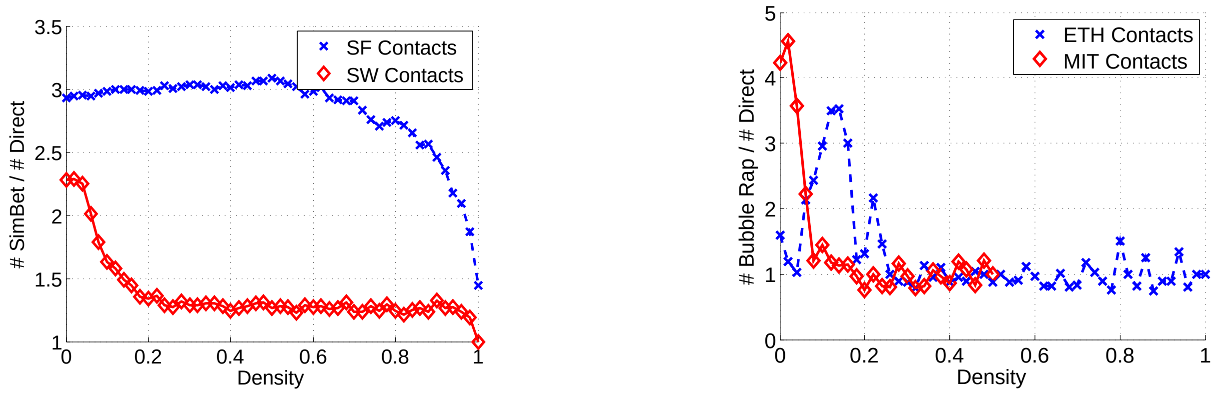
Most Recent Contacts: In many scenarios, it is reasonable to assume that very old contacts may not have the same predictive power as more recent ones. Hence, we could aggregate only the E_n most recent edges to construct the social graph. To achieve this, each edge in the graph is labeled with the last time of appearance in order to know which edge to replace in case of a new contact. Specifically, we assign each pair of nodes $\{u,v\}$ a timestamp $t_{\{u,v\}}$, and maintain a time variable $t_{\text{oldest},n}$ that keeps track of the oldest edge in G_n. We can then write our indicator function at time n as

$$\mathbb{1}_{e_{u,v}} = \begin{cases} 1 & \text{, if } t_{\{u,v\}} > t_{\text{oldest},n} \\ 0 & \text{, otherwise.} \end{cases} \tag{3}$$

Most Frequent Contacts: Another option is to aggregate only the set of most frequent contacts in E_n. In many scenarios, a more frequent contact has a higher probability of occurring again soon, and thus reflects a stronger social link. Instead of a timestamp of last appearance, 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. Then, the indicator function corresponding to the above mapping is:

$$\mathbb{1}_{e_{u,v}} = \begin{cases} 1 & \text{, if } c_{\{u,v\}} > c_{\text{least},n} \\ 0 & \text{, otherwise.} \end{cases} \tag{4}$$

It is important to note that a large number of different and more sophisticated mappings are possible such as, for example, weighted graphs [18]. 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.



(a) SimBet.

(b) Bubble Rap.

Fig. 2. Number of delivered messages for SimBet/Bubble Rap relative to number of delivered messages by direct transmission.

IV. Evaluation

In this section, we evaluate the performance of different versions of SimBet and Bubble Rap, using the discussed aggregation functions. We use the number of messages delivered within a certain time interval as the performance metric.

A. Contact Processes

For our simulations, we have used four contact generators, two of which are synthetic contact processes and two are real mobility traces. For the synthetic contact processes, we assume that there is a (social) graph structure between the participating nodes that governs the probability of two nodes coming into contact at any time, regardless of the actual mobility model underlying the process. As it has been consistently observed (e.g., [11]), such social graphs often exhibit *scale-free* or *small-world* behavior, which is what our two synthetic models aim to capture. For both models, we simulate 100 nodes and $500'000$ contacts according to the procedure described below. The two mobility traces, collected from real network environments, are used to further corroborate our analysis and findings.

SF: Initially in the SF (*scale-free*) contact process, each node is assigned a popularity according to a power law (with exponent 3). Then, successive endpoints of a contact (i.e., two nodes) are randomly and independently choosen according to their respective popularity. Note that a network of 100 nodes may be too small to call it scale-free, however, we certainly achieve to model a skewed degree distribution. It is important to note that for this, and any of the other contact processes, the social graph resulting from aggregating contacts over time will eventually become fully meshed. The particular degree distribution or small world behavior is supposed to be reflected at the operating level of aggregation (see e.g. Figure 1.(b) or Section III).

SW: The *small world* contact process is inspired by the Watts and Strogatz small world graph model [19]. We number all nodes sequentially, conceptually arranging them to a ring, and let them have “strong” links to four of their neighbors (e.g., node 4 has strong link to nodes $2, 3, 5$ and 6). For a new contact, we select the first node uniformly at random. With probability $p = 0.1$, we select its peer uniformly at random from the set of all other nodes (corresponds to the *shortcut* links of [19]). With probability $1 - p = 0.9$, we select the peer uniformly at random from the set of its strong links (the *regular* links of [19]). This model aims to capture that each node has a number of strong connections, but also gets in proximity of other people[5].

For both SW and SF, draws are with replacement meaning that an existing contact can be picked again.

MIT: The MIT contacts were collected in the *Reality Mining* [21] project, where 97 students and employees of MIT were equipped with mobile phones scanning every 5 minutes for Bluetooth devices in proximity during 9 months. These traces are unique in terms of number of experimental devices and duration, however, with the time granularity of 5 minutes a lot of short contacts were supposedly not logged. 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.

ETH: For the ETH trace [22], 20 students and staff working on the same floor of a building of ETH Zurich were carrying 802.11 enabled devices for 5 days. At an interval of 0.5s, each device sent out a beacon message, the reception of which was logged by all devices in 802.11 radio proximity. This shorter trace contains more than $23'000$ reported contacts and is unique in terms of time granularity and reliability. Although the ETH trace measurement period spans a considerable shorter time, 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. In that sense, the two traces are comparable. For both, MIT and ETH trace, we ignore logged timing information and just ordered the reported contacts according to their start times (i.e., slotted contacts).

[5]Note that for both, SW and SF, *communities* are not well defined. In order to have a model with clear communities, we also used a contact process inspired by the *caveman* model [20]. The results, which we do not present here due to space limitation, are similar to the SW model.

B. Degrading message delivery performance

As a first step, we seek to confirm the intuition that with the *growing time window* aggregation the performance of SimBet and Bubble Rap degrades after a long enough network lifetime. Moreover, we want to investigate how the performance varies with graph density to get an estimate for suitable graph densities for the constant density aggregation mappings.

To do so, we let the contact processes run until the aggregation using the growing time window results in a fully meshed graph (i.e., $d(G_n) = 1$). At the same time, we create one message for all possible pairs of nodes (i.e., $|E|$) at each 0.02 increase in density. Eventually, we log the number of them that gets delivered. Each message has an empirically determined TTL value (2000 for SF, SW and MIT, and 50 for ETH). These values are the minimum that give reasonable delivery ratios for the respective networks (we have also performed simulations with other TTL values, with similar findings). We compare the performance of SimBet and Bubble Rap to *Direct Transmission*, where the source keeps the message until it meets the destination. Direct Transmission serves as a lower bound on performance, so any "smarter" scheme should be able to outperform it.

Figure 2(a) shows the ratio of messages delivered by SimBet versus direct transmission with SF and SW contacts, averaged over 10 simulation runs. Figure 2(b) shows the same performance metric for Bubble Rap using the MIT and ETH contact traces[6]. The first important observation is that, as time increases (i.e., $density \to 1$), the performance of both protocols degenerates to almost as bad as that of direct transmission, in all scenarios. This confirms our intuition that, after the network is operational for a long enough time, SimBet and BubbleRap make increasingly random forwarding decisions. A second observation is that performance seems to be more sensitive to density values in the SW scenario compared to the SF one. With the former, there is a performance peak at very low density with low and steady values for the rest of the densities, whereas, with the latter, performance is high and quite stable for densities up to 0.7. A possible explanation is that in the SF scenario, the aggregated graph remains scale-free (and thus heterogeneous) for a much higher number of edges compared to SW. In the SW case, after a small number of edges, most of the regular edges have been "filled" and the additional ones are shortcut edges turning the graph more random. Finally, performance for both traces seems to behave similarly to the SW model, which is not surprising since social networks are believed to be small-world.[7]

Summarizing, in all scenarios considered, running SimBet and Bubble Rap at small graph densities outperforms the same algorithms at very dense graphs by a factor of 2.3 to 4.5.

[6]Note that we have also performed simulations for SimBet with trace contacts, and BubbleRap with SW and SF contacts; The performance is similar in both cases, so we only depict two plots per graph for clarity.

[7]Note that the density of the MIT graph goes only up to 0.5 despite its long duration. This might be due to the relatively large sampling interval of 5 minutes, resulting in many short coincidental collocations not being logged. In contrast, in the ETH trace the graph gets fully meshed very quickly.

	SF	**SW**	**ETH**	**MIT**
SimBet Most Recent	2.04	2.67	1.57	1.24
SimBet Most Frequent	2.46	2.31	1.42	1.23
Bubble Rap Most Recent	1.24	4.10	1.62	1.70
Bubble Rap Most Frequent	1.07	1.77	1.08	1.80

TABLE I
PERFORMANCE INCREASE WHEN USING CONSTANT DENSITY AGGREGATION RELATIVE TO GROWING TIME WINDOW AGGREGATION.

C. Constant density message delivery performance

Based on the previous observations, we will show next that by keeping the aggregated graph density constant, we can maintain good performance throughout the whole network lifetime. We fix the density to 0.05 using Equations 3 and 4, and compare these aggregations to growing time window.

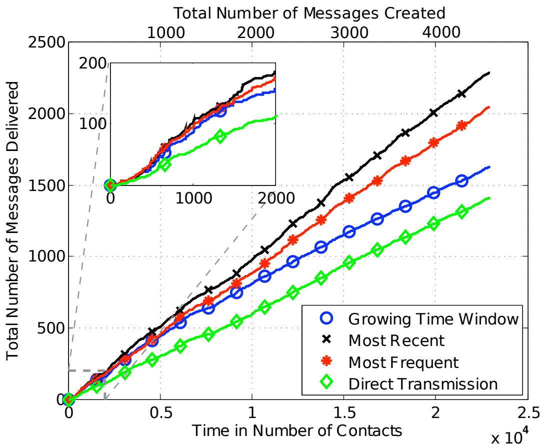
Figure 3 shows the cumulative number of delivered messages (before the TTL expires), when we keep creating messages throughout the simulation (CBR traffic). Figure 3(a) shows the performance of SimBet using the ETH trace. We observe in the subplot highlighting the beginning of the simulation that for low graph density, growing time window aggregation performs as well as constant density functions. After $7'000$ total contacts, the graph is complete and performance of growing time window decreases to that of direct transmission (same slope). On the other hand, the "most recent" and "most frequent" schemes keep delivering about 57% and 42% more messages per time interval, respectively. Figure 3(b) shows the result for Bubble Rap when using the MIT trace.

Table I summarizes the performance increase in terms of number of delivered messages in a certain time interval of the constant density aggregations versus growing time window aggregation. We observe a significant performance increase in all cases. However, the values vary largely, and the results are not easily comparable across different contact generators, due to perhaps an inherent difference in difficulty in doing DTN routing for one type of network over another. Thus, only direct comparisons along the same column are valid.

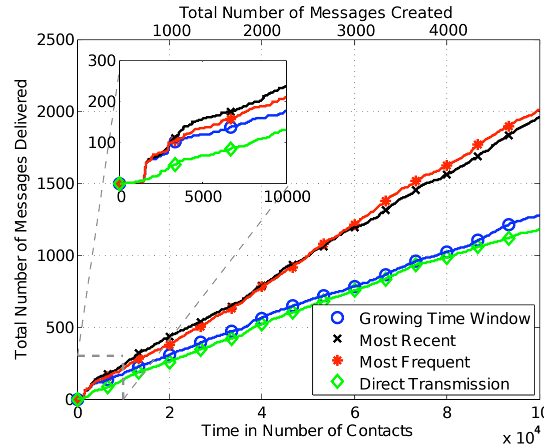
To conclude, the performance of both routing protocols can be significantly improved (up to a factor of 4 in the considered scenarios) by using simple density-based aggregation schemes. We believe that this finding should also hold for any other DTN data dissemination algorithm that uses social network metrics.

D. Discussion & Future Directions

Throughout our earlier discussion and simulation evaluation, we make the simplifying assumption that all nodes have global knowledge to construct their view of the social graph. In reality, not having global knowledge is a fundamental property of DTN networks, and the discussed protocols take measures to calculate global metrics such as betweeness centrality using only local information (e.g., the ego-view in [8]). Our goal in this paper was to decouple these two problems and demonstrate that efficient aggregation is important even with full knowledge. Still, preliminary results we have performed using only local information indicate similar behaviors.



(a) SimBet with ETH contact trace.



(b) Bubble Rap with MIT contact trace.

Fig. 3. SimBet and Bubble Rap performance with different aggregation functions compared to direct transmission.

The presented results demonstrate that the aggregation function used should aim to maintain the aggregated social graph density to a desired range of values, in order for routing protocols that utilize social metrics to perform well. The plots provide some hints as to how to choose the target density in advance, but it is still an open question how to identify a desired density range *online* and in a distributed fashion, and to ensure that nodes can self adapt to an aggregation level that allows efficient calculation of social metrics. One possible direction is to use theoretical analysis to explicitly identify the desired density range for a specific connectivity scenario, and then use an online algorithm that estimates (e.g., by sampling meeting counts and meeting times) when this desired density is reached. Alternatively, one could try to implicitly identify the desired operating point, using calculated social metric values and comparisons thereof as a feedback signal to increase or reduce aggregation. We intend to further investigate this issue.

Finally, although the comparison metric for all our results is the number of delivered messages, one could also infer from the depicted plots, that the delay of both schemes using density based aggregation is smaller than the original protocol. However, due to the interplay of delay and TTL, we defer further investigation of delay properties for future work.

V. Conclusions

In this paper, we established the importance of efficient mapping between the set of “contacts” over time between nodes, and the static “social” graph used by state-of-the-art DTN routing schemes. We showed that current aggregation strategies used by social network based routing protocols such as SimBet (*ever* growing time window) and Bubble Rap (fixed time window) may lead to inefficient forwarding decisions that quickly degrade performance. Then, we showed that simple aggregation strategies that aim to maintain a constant density for the aggregated graph improve the performances up to a factor of $4\times$ in terms of number of messages delivered. 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.

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 ASPLOS-X, 2002.
[2] P. Basu and T. D. C. Little, “Networked parking spaces: architecture and applications,” in Vehicular Technology Conference, 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 WDTN, 2005.
[4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in WDTN, 2005.
[5] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, “Age matters: efficient route discovery in mobile ad hoc networks using encounter ages,” in MobiHoc, 2003.
[6] 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.
[7] J. Leguay, T. Friedman, and V. Conan, “Evaluating mobility pattern space routing for dtns,” in INFOCOM, 2006.
[8] E. M. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in MobiHoc, 2007.
[9] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay tolerant networks,” in MobiHoc, May 2008.
[10] V. Lenders, G. Karlsson, and M. May, “Wireless ad hoc podcasting,” in SECON, 2007.
[11] M. E. J. Newman, “The structure and function of complex networks,” March 2003.
[12] L. Ford and D. Fulkerson, Flows in networks, 1962.
[13] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, “Diversity of forwarding paths in pocket switched networks,” in IMC, 2007.
[14] E. P. C. Jones, L. Li, and J. K. Schmidtke, “Practical routing in delay-tolerant networks,” IEEE Transactions on Mobile Computing, 2007.
[15] T. Spyropoulos, T. Turletti, and K. Obrazcka, “Routing in delay tolerant networks comprising heterogeneous populations of nodes,” IEEE Transactions on Mobile Computing, 2009.
[16] 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.
[17] L. C. Freeman, “A set of measures of centrality based on betweenness,” Sociometry, vol. 40, no. 1, pp. 35–41, 1977.
[18] M. E. J. Newman, “Analysis of weighted networks,” 2004.
[19] D. J. Watts and S. H. Strogatz, “Collective dynamics of ’small-world’ networks.” Nature, vol. 393, no. 6684, pp. 440–442, June 1998.
[20] D. Watts, Small Worlds : The Dynamics of Networks between Order and Randomness (Princeton Studies in Complexity), 2003.
[21] N. Eagle and A. Pentland, “Reality mining: sensing complex social systems,” Personal and Ubiquitous Computing, May 2006.
[22] “Measurements from an 802.11b mobile ad hoc network,” in EXPONWIRELESS, 2006.