GROUPS-NET: Group Meetings Aware Routing in Multi-Hop D2D Networks

Ivan O. Nunes, Clayson Celes, Pedro O. S. Vaz de Melo, Antonio A. F. Loureiro
Department of Computer Science
Federal University of Minas Gerais
Brazil
{ivanolive, claysonceles, olmo, loureiro}@dcc.ufmg.br

ABSTRACT
In the next generation cellular networks, device-to-device (D2D) communication is already considered a fundamental feature. A problem of multi-hop D2D networks is on how to define forwarding algorithms that achieve, at the same time, high delivery ratio and low network overhead. In this paper we aim to understand group meetings’ properties by looking at their structure and regularity with the final goal of applying such knowledge in the design of a forwarding algorithm for D2D multi-hop networks. We introduce a forwarding protocol, namely GROUPS-NET, which is aware of social group meetings and their evolution over time. Our algorithm is parameter-calibration free and does not require any knowledge about the social network structure of the system. In particular, different from the state of the art algorithms, GROUPS-NET does not need communities detection, which is a complex and expensive task. We validate our algorithm using different publicly available data-sources. In real large scale scenarios, our algorithm achieves approximately the same delivery ratio of the state-of-art solution with up to 40% less network overhead.

1. INTRODUCTION
In recent years, high data rate applications such as videos, songs, games, and social media have become increasingly popular in cellular networks. Device-to-Device (D2D) communication has been proposed to facilitate high data rate transmissions among near users offering higher throughput, efficient spectral usage, extended network coverage and improved energy efficiency.

D2D refers to the direct transmission of content between devices without the need for all the data to go through the base station, as in traditional cellular networks. The D2D transmission can be classified in two basic types: 1-hop transmission, in which a message goes directly from the source to the destination if they are close enough to each other; and multi-hop transmission, where the message must be opportunistically routed, device by device, from the source to the destination. This last one is more complex, since it depends on the intermittent communication structure of a mobile network and is suited for communication in which larger delivery times might be tolerated. This concept was firstly introduced in the context of Delay Tolerant Networks (DTN), but it has many applications to D2D networks as well, as discussed in [11]. A fundamental difference from D2D networks to pure DTNs is the possibility of a centralized control plane with a distributed data plan. In this case, the base station controls the forwarding policy while the data is transmitted from device-to-device until it reaches the destination.

Forwarding algorithms in multi-hop D2D networks have the goal of achieving cost-effective delivery, i.e., the highest possible delivery ratio with the lowest possible network overhead. In this case, the delivery ratio is measured as the percentage of the opportunistically routed messages that are successfully delivered to the destination. Successfully delivered messages are the ones that the base station will not need to deliver itself, enabling bandwidth offload. The network overhead is measured by the average number of times that the content will need to be D2D-transmitted for the message to get to its destination. A high number of transmissions may negatively impact the users’ experience by, for example, increasing the devices’ energy expenditure.

Considering these metrics, the most successful strategy for opportunistic cost-effective forwarding, Bubble Rap [9], relies on information about static social communities and nodes’ centrality (which can be approximated by the node popularity within the mobile network). The use of Bubble Rap in D2D Networks is proposed in [11]. However, communities have some problems. First, they are computationally expensive to detect [14]. Second, they are hard to detect in a distributed way since the individual nodes will not have information about the contact graph of the whole network. Existent distributed community detection algorithms have at most 85% precision in detected communities, as reported in [10]. Another problem of community detection algorithms is the parameter calibration. The most successful community detection algorithms depend on parameters that must be calibrated for each specific scenario [19]. In a real-time application, such as D2D communication, such calibration is not feasible. In addition to these mentioned problems, there is no established truth for community detection. Abrahao et al. [1] evaluate community detection schemes and show that,
for the same scenarios, different community detection algorithms yielded very different results for communities’ compositions. Finally, static communities detection does not account for the dynamism in humans’ social relationships, i.e., how they change over time.

Aiming to address these issues, we propose to look at social groups’ meetings, instead of detecting communities. A social group meeting is defined as a group of people who are together, in space and time, for some social reason or common goal. People in a bus, for example, are together because they share the same goal of getting to a given point of interest. Students in the classroom share the objective of learning the class’ subject content. Friends hanging out at a bar share the social motivation of being together to relax and talk to each other. All of these are examples of social group meetings. As human beings have regular schedules and routines, it is reasonable to expect social group meetings to present some regularity as well.

From the implementation point of view, a device can detect a group meeting of which it is part of by simply looking at the list of devices that remained nearby for more than a threshold time, for example, 10 minutes. This way, group meetings can be easily detected in a distributed fashion. Moreover, the group meeting detection method does not change depending on the scenario nor requires parameters calibration for each specific network, as in community detection schemes. In addition to those desirable characteristics, by looking only at recent group meetings or by giving higher importance to more recent meetings, it is possible to account for the dynamic nature of social relationships. All of these favorable characteristics motivated the study of group meetings to propose an opportunistic routing scheme that is better suited for D2D networks than the current social-aware proposals. Therefore, the contributions of this work are the following:

- A methodology for detecting and tracking group meetings from pairwise contact traces;
- A characterization of group meetings regularity properties and its modeling as a Poisson process, which enables to predict future meetings using the information about the most recent ones;
- An opportunistic forwarding algorithm for D2D networks, aware of group meetings, which does not need to detect communities nor calibrate parameters, and achieves better cost-effectiveness than the state of the art solution in real large scale scenarios.

This paper is organized as follows. Section 2 presents some of the related work and discusses the corresponding contributions for understanding human mobility and for leveraging cost-effective opportunistic routing. Section 3 formalizes our methodology to detect and track group meetings from pairwise contact traces. Section 4 describes the main properties of group meetings that makes them interesting in the design of a new forwarding strategy. Section 5 introduces GROUPS-NET, a group meetings-aware routing protocol. Section 6 comparatively evaluates GROUPS-NET, contrasting its performance with the state of art solution, Bubble Rap, in different network scales. Finally, Section 7 presents the final remarks and future work.

2. RELATED WORK

Human mobility. Some rules that govern human mobility have already been revealed. Studies have used diverse data sources to look into individual mobility patterns [6], impact of large scale events in urban mobility [3], typical transitions between points of interest in a city [20], and characterization and prediction of traffic conditions [13][14][21]. Some of the studies investigate how large scale events, like weather changes for example, may affect traffic conditions [3]. To the best of our knowledge, there is no such characterization related to group mobility and its applications to mobile networks, which is one of the contributions of this work.

Opportunistic Routing. Among several solutions proposed for opportunistic routing, the most successful ones were the probabilistic and social-aware routing. Probabilistic routing in DTNs was introduced by Lindgren et al. [12]. The central idea of this approach is that encounters that happened more recently are more likely to happen again in the near future. The PROPHET algorithm, which implements this idea, achieved great success, being years later outperformed by social-aware strategies. In this direction, Hui et al. [9] used the social community structure, detected from contact graphs of mobile networks, combined with network nodes’ centrality, to propose a forwarding algorithm named Bubble Rap. To the best of our knowledge, until now Bubble Rap was the most cost-effective forwarding algorithm in terms of high delivery ratio and low network overhead. The use of Bubble Rap in D2D networks is proposed in [14]. However, community detection has several problems which undermine its application in real networks. In the present work we introduce GROUPS-NET, a social-aware forwarding algorithm based on the tracking of social group meetings, which does not suffer from such problems and achieve better cost-effectiveness in real large scale scenarios.

3. GROUP MEETINGS DETECTION

Group meetings may be easily detected in a real scenario by looking at the list of near devices, for instance. To study group meetings properties it is necessary to detect group meetings from pairwise contacts traces, which are the typical data sources to study social-aware forwarding algorithms. In such traces, each pairwise contact is registered with the two nodes involved and the time when the contact happened. In this section, we describe the methodology we used to detect group meetings from pairwise contact traces. Notice that in a real distributed scenario these steps would not be necessary since group meetings detection is simple to perform distributively. However, this methodology must be applied to enable such study using pairwise contact traces.

In the present study, we used the MIT Reality Mining [5] and Dartmouth [9] traces, which are contact traces containing 80 and 1200 users respectively. In the MIT Reality Mining the monitored users reside in two university buildings and were monitored for several months. Contacts were registered when two users were less than 10 meters apart. Although the MIT Reality Mining trace consists of a specific and small scale scenario, we considered this trace in the present work since it is the original trace used to validate Bubble Rap in [9]. The Dartmouth trace registered contacts of all of the students in a university campus for two months. To the best of our knowledge, Dartmouth is the largest scale
discuss the ideal time size for proximity detection.

First, we slice the proximity dataset in time windows \( tw \). To detect group meetings, we propose a social graph model.

### 3.1 Modeling Contact Traces as Graphs

To detect group meetings, we propose a social graph model. First, we slice the proximity dataset in time windows \( tw \). We discuss the ideal time size for \( tw \) in Section 3.2. Contacts within the same slice \( tw \) will be aggregated in a contact graph \( Gc(V,E[tw = i]) \), in which \( V \) is the set of vertices representing entities in the data set (i.e., people) and \( E \) is the set of edges that represent proximity contacts between a pair of entities in \( V \). Thus, in our model, the trace processing will result in a set of subsequent, undirected, edge-weighted graphs: \( S = \{ Gc(V,E[tw = 0]), Gc(V,E[tw = 1]), \ldots, Gc(V,E[tw = n]) \} \). The weight of an edge \( (v,w) \in E \) is given by (i) the number of contacts registered between \( v \) and \( w \) during the time slice \( tw \); or (ii) the sum of \( (v,w) \) contacts duration, inside \( tw \) time slice, if contacts’ duration are available in the trace. Since in the MIT Reality Mining this information is not available we used the first option. For the Dartmouth trace, we used the second option (because contacts’ duration is available).

### 3.2 Data Characterization

Considering a pre-defined \( tw \) size, for instance 30 minutes, a single contact within the whole time slice does not necessarily mean that entities are socially interacting. Single contacts might be random encounters, even if nodes share a social bond. It might be caused by intersections between individual trajectories and should not necessarily mean a social interaction. We are interested in defining a threshold \( w_{th} \), for the minimum edge weight, which should be enough to consider that the entities are in fact together inside the time slice \( tw \). It is clear that to properly define both the size of slice \( tw \) and the edge weight threshold \( w_{th} \) we need to analyze data set properties (e.g., sampling rate). In Section 3.2 we perform a characterization of the studied dataset to be able to properly define \( tw \) and \( w_{th} \). Throughout this section, due to space limitation, we only present the characterization of the MIT Reality Mining trace, but the same methodology can be applied to the Dartmouth trace and to other contact datasets as well.

First, we analyze the time between pair re-encounters, i.e., once a pair has met, what is the distribution of the time until the next meeting. Figure 2 shows that the re-encounter behavior is very periodical, with peaks around every five minutes (red dashed lines). This behavior indicates that the deployed system for data acquisition acts every five minutes most of the times, but for some reason it can also actuate in shorter periods. Looking at the CDF of the re-encounter probability, approximately 95% of the re-encounters can be captured with a \( tw \) of one hour. Thus, we set the duration of the time window \( tw \) to one hour.

Next, we analyze the frequency distribution of contact pairs within one hour to define \( w_{th} \). Figure 1 shows that 27% of pairs that meet in a given hour only meet once. We assume these one-time meetings as coincidence meetings. For meeting frequencies from 2 to 12, the graphic shows values between 5% and 10%. For frequencies higher than 12, the probability becomes very low, which is consistent with the assumption that the data acquisition occurs mostly in periods of five minutes, but rarely less than five. From 2 to 12 encounters per hour, we have similar values in the CDF when compared to \( P(X = 1) \). For this reason, \( w_{th} = 2 \) for the MIT Reality Mining dataset.

In summary, through the dataset characterization, we were able to define that two or more contacts within an hour are enough to be considered social interaction in the MIT Reality Mining trace. Through similar analysis we defined \( tw = 1h \) and \( w_{th} = 10 \) minutes for the Dartmouth trace.

### 3.3 Detection and Tracking

After defining values for \( tw \) and \( w_{th} \), we define a social group as follows:

- **Definition of social group meeting**: A group meeting is a community detected in \( Gc(V,E[tw = i]) \), i.e., the graph generated from the \( i^{th} \) time slice of the trace \( S \), after eliminating edges between pairs with weight below the threshold \( w_{th} \).

So far, we have established a model to represent social interactions that consists of graphs generated from peer contacts in traces’ time slices. Following the above group definition, we must be able to detect communities (social groups represented by more densely interconnected parts within a graph of social links) in such graphs in order to track social groups.

There are several community detection algorithms, such as CPM. From the existing algorithms, in this work we, use the Clique Percolation Method (CPM). The main reasons for using CPM are twofold: (1) community members can be reached through well connected subsets of nodes, and (2) communities may overlap (share nodes with each other). This latter property is essential, as most social graphs are characterized by overlapping and nested communities. For each time-slice graph \( Gc(V,E[tw = i]) \), we compute CPM.

In CPM, a community is defined as a union of all \( k \)-cliques (complete sub-graphs of size \( k \)) that can be reached from each other through a series of adjacent \( k \)-cliques (where adjacency means sharing \( k - 1 \) nodes). The CPM parameter \( k \) is determined during the community detection process. This method allows for the detection of communities even in networks with overlapping nodes, where traditional methods might struggle.
Figure 2: Probability function of the time $x$ (in seconds) until the next meeting. Red dashed lines show a fixed inter-measurement time of 318 seconds in which re-encounter peaks happen. This means that most of the trace proximity records were acquired in fixed periods of 318 seconds. The probability of a pair of nodes meeting again has approximately exponential distribution and 95% of the re-encounters happen in less than one hour.

Figures 3: Group detection with CPM, in the MIT proximity trace, with $tw = 1h$ in three consecutive time windows, at February 5th of 2009. Only edges with $w_{th} \geq 2$ are represented.

limits the minimum size of detected communities. CPM has remarked itself as one of the most effective methods once fed with correct parameters [19]. We set $k = 3$ to detect social groups, i.e., we consider groups of three or more people. Figure 3 shows groups detected with CPM in the MIT proximity trace [5] at February 5th of 2009, at 7, 8 and 9am, which are subsequent time slices of one-hour size.

As Figure 3 shows, group nodes composition may change over time, but some of the detected groups are present throughout several consecutive time windows $tw$. Since we are interested in investigating the regularity in group meetings, there must be a way of tracking them, i.e., a criterion for considering that two groups in consecutive time slices are in fact the same group. With that goal in mind, we introduce the Group Correlation Coefficient $\rho(G1, G2)$:

$$\rho(G1, G2) = \frac{|V(G1) \cap V(G2)|}{|V(G1) \cup V(G2)|}$$ (1)

where $|V(G1) \cap V(G2)|$ is the number of common nodes in groups $G1$ and $G2$ and $|V(G1) \cup V(G2)|$ is the total number of different nodes present in the union set of both groups. The coefficient $\rho$ assumes values from 0 to 1, where 0 means no correlation, i.e., no node belongs to both groups and 1, which means that $G1$ and $G2$ have the exact node composition. Group correlation coefficient is a measurement of the stability in groups’ composition. We consider two time separated groups $G(tw = i)$ and $G(tw = j)$ to be the same if $\rho(G(tw = i), G(tw = j)) > 0.5$, i.e., if at least 50% of the group members remain the same. Thus, a $\rho$ value greater than 0.5 is the necessary condition to map two times each separated group as the same. If $\rho < 0.5$ for two separated
groups, then one of these groups could be mapped to two different groups with less than half of its node composition in a different time window, adding complexity to the group tracking. At the same time a $\rho$ threshold of 0.5 allows high volatility in group composition, making it possible to better analyze the groups’ evolution.

4. SOCIAL GROUP MEETINGS PROPERTIES

In the previous section, we showed how we are able to detect group meetings from proximity traces. This section presents the main properties of such group meetings that make it interesting to use them to perform D2D Routing. Further characterization of other group mobility properties may be found in [16].

Figure 4(a) presents the frequency of group re-encounters for the MIT Reality Mining, i.e., given the fact that a group first met at time $t = 0$, how group re-encounters are distributed along the next hours ($t$ hours after the first meeting). The result reveals that the probability mass is concentrated around peaks of 24-hour periods (represented by red dotted lines). This means that group meetings are highly periodic in a daily fashion. One may also notice higher peaks marked with green dashed lines. Green dashed lines represent periods of seven days, meaning that group meetings also present weekly periodicity. This periodicity makes sense since people have schedules and routines. This result motivated our next experiment, which tries to answer the question: is it possible to use past group meetings to predict future ones?

In our next experiment, we select a node as the origin of a message and simulate an epidemic message transmission, i.e., every time a node with a message meets a node that does not have it yet, the message is propagated. We simulate the message propagation selecting each node of the MIT dataset as origin and divide the rest of the nodes into two classes: nodes that have belonged to a group together with the origin in the past 30 days and nodes that have not. Then, we compute the delivery ratios of the two classes of nodes. We consider that the message is delivered to node $N$ if node $N$ receives the message within seven days after the start of the dissemination.

As presented in Figure 4(b), for different months, the delivery ratios to nodes that have been in groups with the origins are over two times higher than of the other groups. Around 90% of them received the message sent by the origin within one week (result omitted). On the other hand, the delivery ratio to nodes that have not been in a group together with the origin is around 40%. This result conforms with the periodical behavior presented in Figure 4(a) (a group that has met in recent past is likely to meet again soon) and is a key insight on how group meetings could and should be used to better design opportunistic routing protocols. One may notice that in June the delivery ratios for both classes significantly drop. This behavior is explained by the fact that the trace was collected in a university campus and, in June, most students in the US start to leave the campus for summer vacation.

To use group meetings to design a forwarding policy, there must be a representative statistical model for group meetings regularity. Due to group meetings’ periodicity, presented in Figure 4, it makes sense to model such behavior as a Poisson process. In a Poisson process, the CDF of the number of occurrences along the time must be well approximated by a straight line with slope $\lambda$. To verify the goodness of fit of group meetings to a Poisson process, for each group in the trace, we perform a linear regression of the number of meetings over time. Then, we compute the $R^2$ value of each group, which measures how well the linear model fits to the number of meetings. Figure 5 exemplifies such regression for different values of $R^2$. Figure 4(c), which presents the frequency distribution of $R^2$ for all groups in the trace, shows that group meetings have a good fit to a Poisson process, most of them with $R^2$ values of 0.85 or higher. We use this Poisson process model to design our forwarding algorithm, as discussed in Section 5.

5. GROUPS-NET: GROUP MEETINGS AWARE ROUTING

Considering the group meetings properties revealed in Section 4, our algorithm, GROUPS-NET (Group ROUTing in Pocket Switched-NETworks), works by forwarding the messages from the origin node to the destination node through the most probable group-to-group route. To model the probability of group-to-group paths, GROUPS-NET uses a probabilistic graph model in which each group detected in the recent past is represented as node and the edges between two nodes represent the probability of a message being propagated from one group to another. To assign a probability for an edge that links two groups, for instance groups $A$ and $B$, GROUPS-NET considers (i) the probability of groups $A$ and $B$ meeting again in the near future, and (ii) the probability of a message being carried from a group $A$ to a group $B$ by a person who is member of both groups. To compute such probability, GROUPS-NET relies on two main properties:

- **Meetings regularity**: Each group is assigned with a probability of meeting again soon, which is based on the number of times that the group has met in the recent past. We show, in Section 4, that it follows a Poisson process. The regularity property comes from the group meetings periodical behavior, depicted in Figure 4(b). The key insight is that the higher the number of meetings of a group in the recent past, the higher the probability that group meeting again in the near future. By only considering meetings in the recent past, the meetings regularity property accounts for the social dynamism of human relationships.

- **Shared group members**: In a group meeting, a message can be propagated for all nodes involved in the meeting. However, the message must be propagated forward to the next group and so on, until it reaches a group that the destination node is member of. This group-to-group propagation is made by nodes that belong to both groups linked by an edge. If two groups have a higher number of member nodes in common, there is a greater probability for the message to be carried from group $A$ to group $B$, for instance. Thus, higher probabilities should be assigned to edges between groups that have more shared members.

To combine both of the aforementioned properties, GROUPS-NET assigns nodes’ probabilities as the product of the proba-
Figure 4: a) Probability of a given group re-meeting $t$ hours after its first meeting. Red dotted lines represent 24-hour periods and green dashed lines 7-day periods. b) Average delivery ratio, for different origin nodes, from November of 2008 to June of 2009. c) R-squared distribution for Poisson distribution fits of each group of the trace.

Figure 5: Poisson process fit for different values of $R^2$.

The $\lambda$ value of a group in the Poisson process is the inverse of the group’s average inter-meeting time. Thus, given a fixed-time window size of length $L$, which is the considered time to look back in past (e.g., 3 weeks), the $\lambda$ of a group can be estimated by:

$$\lambda = \frac{\text{number of meetings}}{L}. \quad (2)$$

Since group meetings follow a Poisson process (as we show in Section 4), the probability of a given group to meet $K$
times in the \( t \)-time interval is given by the expression:
\[
P[N(t) = K] = \frac{e^{-\lambda t}(\lambda t)^K}{K!}.
\] (3)

For our opportunistic routing algorithm, we are interested in the chance of a group to meet again at least one time during the considered time interval \( t \):
\[
P[N(t) \geq 1] = 1 - P[N(t) = 0] = 1 - e^{-\lambda t}.
\] (4)

Equation 4 shows that the frequency in group meetings can be used to compute the probability of a group meeting to happen at least once in the near future time \( t \). Thus, GROUPS-NET sets nodes’ probabilities according to Equation 4. The time \( t \) should be set according to the messages’ TTLs of the network.

To consider the probability of the message being propagated between two different groups by common members of both, the algorithm computes the overlap in groups’ members composition as:
\[
P(m : G1 \rightarrow G2) = \frac{|V(G1) \cap V(G2)|}{|V(G1) \cup V(G2)|}.
\] (5)

After setting the edges probabilities, the algorithm re-computes each edge weight as the product of each of the groups’ re-meeting probabilities (computed with Equation 3) multiplied by the groups’ composition overlap, as in Equation 6.
\[
W(E_{G1,G2}) = P(G1 \rightarrow G2) \times P_{G1}[N(t) \geq 1] \times P_{G2}[N(t) \geq 1]
\] (6)

Therefore, with edges’ probabilities set, the most probable group-to-group probability can be computed by the product of each edge in its path (Equation 7). By exploiting the logarithm-likelihood property described in Equation 8, the most probable path can be simply computed by a shortest path algorithm, such as Dijkstra, after setting each edge weight \( W(E_{i,j}) \) to \(-\log(W(E_{i,j}))\).
\[
P(R) = \prod_{E_{i,j} \in R} W(E_{i,j}),
\] (7)
\[
\arg\max_{R} \left( \prod_{E_{i,j} \in R} W(E_{i,j}) \right) = \arg\max_{R} \left( \log(\prod_{E_{i,j} \in R} W(E_{i,j})) \right)
\] (8)

Using this modeling, we propose GROUPS-NET to compute the most probable group-to-group path and forward a message opportunistically to nodes that belong to such route. GROUPS-NET is formalized in Algorithm 1.

The GROUPS-NET algorithm has an upper bound defined by the computation of the shortest path in a graph, which has time complexity of \( O(V^2 \log(V)) \), where \( V \) is the number of different groups in the network, i.e., vertices in the groups graph \( G[V, E] \) of Algorithm 1.

Notice that to compute the most probable group-to-group path, it is necessary to centralize the information about recent group meetings at some point. Such computation is made possible by the D2D architecture, which defines a centralized control plane and a decentralized data plane. This is the reason why GROUPS-NET properly fits applications in the D2D networks, but it is not necessarily feasible in purely distributed DTNs. To centralize the information about groups meetings, each device must periodically (e.g., once a day) update the base station with its recent group meetings.

When a given origin device wishes to send a content to a destination, it sends a request to the base station, which computes the most probable group-to-group path and sends it back to the origin device. Next, the forwarding policy proceeds as follows: starting by the origin device, each device will make the decision of forwarding or not the content to a new encountered device based on the condition that the encountered device must be a member of at least one of the groups that belong to the most probable group-to-group path.

6. COMPARATIVE ANALYSIS

To validate the performance of GROUPS-NET, we compare it to the forwarding algorithm that achieved the most cost-effective performance for D2D networks: Bubble Rap [9].

6.1 Bubble Rap Algorithm

The Bubble Rap algorithm identifies static social communities by looking at densely interconnected nodes in the aggregated contact graph during the whole trace using the Clique Percolation Method [18]. Therefore, each node in the network must belong to at least one community. Nodes that
do not belong to any community are assigned to a pseudo-community of one node. This is necessary for the forwarding algorithm operation. Moreover, each node gets a measure of its global popularity in the network (GlobalRank) and a local measurement of popularity, which is valid within that node’s community (LocalRank). Using these parameters, the forwarding strategy works as follows:

- At each encounter, a given node transmits its content if the encountered node has a higher GlobalRank, or if the encountered node belongs to a community of which the final destination is a member.
- Once the message is inside the final destination’s community, the forwarding process occurs if the LocalRank of the encountered node is higher than the LocalRank of the node that has the message. This procedure goes on until the message reaches the final destination.

With the purpose of having a fair comparison, in this work we implemented Bubble Rap using the community detection pre-calibrated parameters reported in [7], which provided the best results in terms of cost-effective content delivery. Also, the GlobalRank and the LocalRank were calculated using the C-Window technique that better approximated the node centrality metric in their experiments [9].

6.2 Performance Evaluation

With the goal of evaluating GROUPS-NET and Bubble Rap, we used the following metrics:

- **Delivery ratio**: Evaluates the percentage of successfully delivered messages along the time.
- **Number of transmissions**: Measures the network overhead, i.e., the number of D2D transmissions that each algorithm performs along the time.

For each evaluated trace (MIT and Dartmouth), an (origin, destination) pair is randomly selected among the users of the trace with uniform probability. Moreover, the time \( t \) in which the message transmission starts at the origin node is also randomly selected within the trace duration. Both protocols were executed with 500 randomly generated (origin, destination, time). This process was repeated 8 times with different seeds for random number generation, to obtain 95% confidence intervals.

This way, we aim to capture diversified behavior patterns throughout the trace, conferring generality to the tests. Together with GROUPS-NET’s and Bubble Rap’s results, we also plot the results for a flooding transmission. In the flooding scheme, the message is always propagated whenever a node that has a message encounters a node that does not have it yet. The flooding establishes the upper bound for the delivery ratio and for the network overhead. In our tests, the recent past period used by GROUPS-NET to predict future group meetings was of three weeks, since it is enough to capture both, daily and weekly periodicity.

Figure 6 presents the comparative results in terms of delivery ratio and network transmissions overhead along the time. The results for the MIT Reality Mining trace, presented in Figures 6(a) and 6(b), show that in the first hours, after the beginning of a transmission, GROUPS-NET has a slightly higher delivery ratio and, in the final hours, Bubble Rap overcomes it, successfully delivering a small percentage more. Throughout the whole message propagation time, GROUPS-NET presented a slightly higher network overhead. This happens because the MIT Reality Mining trace is a very particular, since all monitored users reside in the same university buildings. For this reason, they are expected to have stronger social bonds than regular nodes in D2D scenarios. This characteristic benefits an algorithm such as Bubble Rap, which uses the static social structure in its forwarding policy.

This fact motivated the study of a large scale and more general trace, such as Dartmouth, to evaluate the forwarding tests. However, notice that even in this specific scenario, GROUPS-NET presented a competitive result when compared to Bubble Rap, with the advantage of not requiring community detection and parameters’ calibration, which are hard or unfeasible in a practical real-time scenario.

Figures 6(c) and 6(d) present the results for the same experiment performed in the Dartmouth trace, which is more general and has a larger scale. In this scenario, GROUPS-NET achieved a considerably better performance than Bubble Rap. In the period from 24 to 96 hours after the start of the message propagation, Bubble Rap obtains higher delivery ratio, but, after that, being outperformed by GROUPS-NET until the end of the three weeks’ transmission period. With respect to the network overhead, after the sixth hour, Bubble Rap starts to transmit much more messages than GROUPS-NET, presenting an average overhead 50% higher in the following hours. For 1000 different (origin, destination) messages, this represents an economy of 60000 D2D-transmissions in the network.

Defining a benefit-cost metric as the ratio between successful delivery and network overhead, throughout the duration of the transmissions, GROUPS-NET presents a better benefit-cost than Bubble Rap. As depicted in Figures 7(a) (in log-scale) and 7(b) (in regular-scale), after the first hours of transmission, GROUPS-NET reaches two times Bubble Rap’s benefit-cost. As mentioned before, GROUPS-NET also has the advantage of not depending on community detection schemes nor needing parameter calibration, being for these reasons a viable practical solution.

6.3 Discussion

By forwarding messages through the most probable group-to-group path, GROUPS-NET achieves a high delivery ratio, which is comparable to the flooding upper bound and to Bubble Rap in both, small-scale and large-scale scenarios. This conforms to the result presented in Figure 4(b). However, by looking at the results presented in Figure 6, one question that arises is: Why does GROUPS-NET present a much lower overhead in large-scale scenarios, but not in small-scale ones? The answer to this question rests in the nature of each algorithm.

Bubble Rap works by forwarding messages to nodes that have a higher popularity. Therefore, the maximum number of transmissions is limited by the nodes in the network that have higher popularity than the origin. When the origin is randomly selected, the expected number of nodes that are more popular than the origin is directly proportional to the
total number of nodes in the trace. For this reason, when the number of nodes in the network is increased 15 times, from 80 (in MIT) to 1200 (in Dartmouth) Bubble Rap’s overhead also increases by approximately 15 times, as presented in Figures 6(b) and 6(d).

GROUPS-NET maximum overhead, on the other hand, is limited by the size of the most probable group-to-group path (in number of groups) multiplied by the average number of members in the groups of such path. Figure 6(a) shows the distribution of (for most probable group-to-group) path sizes in terms of the number of groups involved in the path. In both traces, for randomly selected (origin, destination) pairs the distribution of the paths’ sizes is not proportional to the number of nodes involved in the network. In fact, the average size does not change for different scales. Since the addition of each new group to the route involves a multiplication by the probability of a new edge (which can significantly reduce a path probability), the most probable paths tend to have few hops. For this reason, GROUPS-NET presents significantly lower overhead in large scales.

To investigate how the network overhead evolves with the increase of the number of nodes in the network, we generate subsets of 200, 400, 600, 800 and 1000 nodes from the original Dartmouth dataset. This subset generation happens using a Snowball algorithm, which firstly assembles a social-contact graph and selects the node with the highest centrality. Next, it adds to the subset of nodes the neighbors of the central node. Then, it adds the neighbors of such neighbors and so on, until the desired number of nodes for the subset is reached. This way, the social structure of the
network is preserved even for small subsets of 200 and 400 nodes, considering a total of 1200 nodes.

Using these subsets, we evaluate the overhead of both algorithms with the goal of analyzing their evolution with the increase of the number of nodes in the network. Figure 8(b) shows the average overhead per message with different network sizes for both algorithms. Figure 8(c) presents the statistical distribution of such messages’ overheads for 1000 transmissions in each scenario. These experiments confirm the expected behavior for the overhead. Bubble Rap overhead presents a linear increase with the size of the network. GROUPS-NET overhead, on the other hand, remains stable for networks with 400 nodes or more. Since real cellular networks often have thousands (or even millions) of nodes, we claim that GROUPS-NET is better suited for such applications.

7. CONCLUSION AND FUTURE WORK

In this work, we introduce the use of social group meetings awareness to leverage cost-effective message transmissions in multi-hop D2D Networks. Firstly, we propose a methodology for detecting group meetings from contact traces. Using the detected groups, we build a probabilistic graph, which is used to compute the most probable group-to-group path for a message to be forwarded. Our approach has the advantage of not requiring community detection and of being parameter-calibration free. Our experiments show that, in large-scale scenarios, this strategy is more cost-effective than previous state-of-art strategy (which is based on static social communities) with respect to delivery ratio and network overhead.
The results show that the group meetings approach is a promising strategy. Based on this idea, one can propose forwarding strategies for several different applications. For example, GROUPS-NET can be modified to consider different types of forwarding such as Single-Source-Multiple-Destinations or Multiple-Source-Multiple-Destinations. This could be simply achieved by computing the union of all pairwise most probable group-to-group paths in the groups graph. Examples of applications that could benefit from Single-Source-Multiple-Destinations and Multiple-Source-Multiple-Destinations are those with high download demand and in which timely delivery is not essential, such as smartphones’ system updates or video advertisement.

In the GROUPS-NET strategy, described in Section 5, only the most probable path is considered for forwarding the message. However, more than one path can be considered for forwarding a message. An interesting future work would be to propose an overhead constrained version of GROUPS-NET, which would add a maximum-overhead parameter to the algorithm. This way, GROUPS-NET would forward a message through the N most probable redundant group-to-group paths that involve at most the maximum tolerated number of nodes. The higher the maximum-overhead parameter is, the more similar to a flooding the forwarding will behave, since the flooding strategy forwards messages through all possible paths. In the case that the single most probable path involves more nodes than the maximum tolerated overhead, then the most probable path with less nodes than the maximum overhead would be chosen instead. This strategy would allow to decrease the base station bandwidth demand and, at the same time, control the D2D network overhead, a feature that is not possible in previous forwarding strategies.

Other interesting future work would be the evaluation of GROUPS-NET performance with different proposals of group meetings detection methods. Also, it would be interesting to evaluate the algorithm in a real scenario, in which groups would be distributed detected by the devices themselves, as in a real D2D network implementation.

8. REFERENCES

[1] B. Abrahao, S. Soundarajan, J. Hopcroft, and R. Kleinberg. On the separability of structural classes of communities. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 624–632. ACM, 2012.
[2] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano. Road traffic congestion detection through cooperative vehicle-to-vehicle communications. In Local Computer Networks (LCN), 2010 IEEE 35th Conference on, pages 606–612. IEEE, 2010.
[3] F. Calabrese, F. C. Pereira, G. Di Lorenzo, L. Liu, and C. Ratti. The geography of taste: analyzing cell-phone mobility and social events. In Pervasive computing, pages 22–37. Springer, 2010.
[4] S. Dunne and B. Ghosh. Weather adaptive traffic prediction using neuralwavelet models. Intelligent Transportation Systems, IEEE Transactions on, 14(1):370–379, 2013.
[5] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. Personal and ubiquitous computing, 10(4):255–268, 2006.
[6] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. Nature, 453(7196):779–782, 2008.
[7] S. Gregory. Finding overlapping communities in networks by label propagation. New Journal of Physics, 12(10):103018, 2010.
[8] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. Computer Networks, 52(14):2690–2712, 2008.
[9] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. Mobile Computing, IEEE Transactions on, 10(11):1576–1589, 2011.
[10] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. In Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture, page 7. ACM, 2007.
[11] Y. Li, T. Wu, P. Hui, D. Jin, and S. Chen. Social-aware d2d communications: qualitative insights and quantitative analysis. Communications Magazine, IEEE, 52(6):150–158, 2014.
[12] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. ACM SIGMOBILE mobile computing and communications review, 7(3):19–20, 2003.
[13] J. Lu and L. Cao. Congestion evaluation from traffic flow information based on fuzzy logic. In Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE, volume 1, pages 50–53. IEEE, 2003.
[14] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai. Overlapping communities in dynamic networks: their detection and mobile applications. In Proceedings of the 17th annual international conference on Mobile computing and networking, pages 85–96. ACM, 2011.
[15] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai. Overlapping communities in dynamic networks: their detection and mobile applications. In Proceedings of the 17th annual international conference on Mobile computing and networking, pages 85–96. ACM, 2011.
[16] I. O. Nunes, P. Vaz de Melo, and A. A. F. Loureiro. Group mobility: Detection, tracking and characterization. In IEEE ICC 2016 International Conference on Communications (ICC’16 SAC-8 SN), Kuala Lumpur, Malaysia, May 2016.
[17] G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. Nature, 446(7136):664–667, 2007.
[18] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435(7043):814–818, 2005.
[19] L. Peel. Estimating network parameters for selecting community detection algorithms. In Information Fusion (FUSION), 2010 13th Conference on, pages 1–8. IEEE, 2010.
[20] R. H. Silva, P. O. Vaz de Melo, J. M. Almeida, J. Salles, and A. A. Loureiro. Revealing the city that we cannot see. ACM Transactions on Internet Technology (TOIT), 14(4):26, 2014.
[21] F. Tereso-Sáenz, M. Valdés-Vela, C. Sotomayor-Martínez, R. Toledo-Moreo, and A. F. Gómez-Skarmeta. A cooperative approach to traffic congestion detection with complex event processing and vanet. Intelligent Transportation Systems, IEEE Transactions on, 13(2):914–929, 2012.
[22] H. Xu, Y. Hu, Z. Wang, J. Ma, and W. Xiao. Core-based dynamic community detection in mobile social networks. Entropy, 15(12):5419–5438, 2013.