

Using Hierarchical Community Structure to Improve Community-Based Message Routing

Matthew Stabeler and Conrad Lee and Graham Williamson and Pádraig Cunningham

School of Computer Science and Informatics
Complex and Adaptive Systems Laboratory
University College Dublin
Dublin 4, Ireland

Abstract

Information about community structure can be useful in a variety of mobile web applications. For instance, it has been shown that community-based methods can be more effective than alternatives for routing messages in delay-tolerant networks. In this paper we present initial research that shows that information on hierarchical structures in communities can further improve the effectiveness of message routing. This is interesting because the examination of hierarchical structure in communities is a little-explored area in social network analysis.

Introduction

It is well established that knowledge of community structure is useful for information routing in mobile social networks (Nazir, Ma, and Seneviratne 2009; Hui, Crowcroft, and Yoneki 2010; Vallina-Rodriguez, Hui, and Crowcroft 2009). Information about the community to which the intended recipient of the message belongs provides a larger *target* to guide routing. In this paper we explore the idea that if communities have a hierarchical organization then information about that hierarchical structure can further help message routing. We describe an algorithm that exploits hierarchical community structure for message routing and present a preliminary evaluation that suggests that this algorithm improves routing efficiency.

While earlier research on community-based methods for message routing has emphasized distributed rather than centralized algorithms for community discovery and centrality (Hui, Crowcroft, and Yoneki 2010; Vallina-Rodriguez, Hui, and Crowcroft 2009) we feel that a combined approach is possible, as modern mobile devices have the storage and processing capacity to run the community finding algorithms that have been considered, if a mechanism for sharing the network structure exists.

In the next section we provide an overview of existing research on community-based message routing and present our algorithm that exploits hierarchical community structure. Next, we present the algorithm we use to uncover hierarchical community structure and then present an evaluation of the performance of that algorithm.

Message Routing

Research into Delay Tolerant Networking (DTN) protocols is concerned with the delivery of messages between nodes within some dynamical, possibly mobile network, in a way that is tolerant to intermittent connections, disconnections and failures. This could be within networks of animals, humans, sensors, satellites or any other system of interacting devices, objects, beings or things. Such systems are often characterised by a sparse network of connections between individuals that change over time. Pocket Switched Networks (PSN) (Hui et al. 2005) which are a sub-set of DTNs, best describe our interests, PSN deal with human networks, specifically, networks created by mobile devices carried by people, for example, mobile phones. In this case, humans interact with each other over time, and their mobile phones communicate via radio antennae (e.g. Bluetooth or WiFi).

There are a number of approaches to delivering messages within DTNs; flooding based approaches, flood the network with copies of messages between any nodes that meet; epidemic-like protocols behave similarly, for example Epidemic Routing (Vahdat and Becker 2000) transmits messages to other nodes with some probability, but limits messages by hop count to reduce overhead. These are perhaps the most effective, but the large number of message copies generated and sent, mean a large overhead in message transmission between nodes. More conservative approaches include Spray and Wait (Spyropoulos, Psounis, and Raghavendra 2005), where a limited number of messages are distributed before a phase where nodes keep the messages until meeting the destination. Other approaches involve probabilistic, or opportunistic mechanisms to predict future interactions, such as PROPHET (Lindgren, Doria, and Schelén 2003) and Context-Aware Adaptive Routing (CAR) (Muñolesi and Mascolo 2009) which use knowledge about previous contacts to fuel predictions about co-locations which are used to decide next-hop routes.

The measures for success of a DTN algorithm can be encapsulated by three metrics; most often the goal is to have high *delivery ratio*, a low overhead *cost* and low *delivery latency* (Crowcroft et al. 2008).

Community-Based Routing

The BUBBLERap (Hui, Crowcroft, and Yoneki 2010) protocol uses community structure to inform routing decisions,

```

On node  $N$  connected with encountered node  $P$ 
for all messages  $M$  held by  $N$  for destination  $D$  do
  Identify bridging community  $BC$  {the smallest community containing  $N$  and  $D$ }
  if  $P == D$  then
     $P \leftarrow M$ 
  else if  $P$  shares a community with  $D$  that is smaller than  $BC$  then
     $P \leftarrow M$ 
  else if  $P$  has a higher local rank for  $BC$  than  $N$  then
     $P \leftarrow M$ 
  else
     $N$  keeps  $M$ 
  end if
end for

```

Figure 1: BubbleH Algorithm

it uses K-CLIQUE clustering (Palla et al. 2005) to generate overlapping community structure from a graph formed from contact between nodes. An edge between two nodes is formed when a contact occurs, and the edge weight is incremented with the duration of the contact every subsequent meeting. Each node is given a global rank, and a rank within each community to which they belong. Ranks are based on their betweenness centrality globally and within each community. To take into account the dynamic nature of contact networks, the authors also threshold edges based on the connected time between nodes, edges where nodes have been connected for less than 4.5 days are removed before the K-CLIQUE algorithm is applied.

Incorporating Community Hierarchy in Routing

Here we present the algorithm that exploits community hierarchical structure, which we call BubbleH. As with the BUBBLERap algorithm, we calculate local rankings based on centrality, however, we do not use the global rank. Instead, we use the hierarchy generated by Hierarchical Greedy Clique Expansion (H-GCE) to drive the mobility of messages within the network. When a node encounters another, it considers whether to pass the message on based on how *close* the other node is in the network structure to the destination node. The algorithm is shown in Figure 1.

We define a *bridging community* as the smallest community that contains the node in question, and the recipient, or destination node. To find the smallest community between two nodes, we find all of the communities that the nodes share. Then, we find the shared community that is lowest in the hierarchical structure. If there are multiple communities at this lowest depth, the community with the lowest membership count is chosen. In the case where we still have multiple candidate communities, we choose the last candidate community found by the H-GCE algorithm. Alternatively, at this point we could also apply further rules for choosing the community, for example, summing the weight of edges and using the highest score.

In the simplified example in Figure 2, to get a message from C to P , a node must determine whether an encountered node is closer to P than itself, if it encounters M it will calculate the smallest *bridging community* as $C1$, and compare

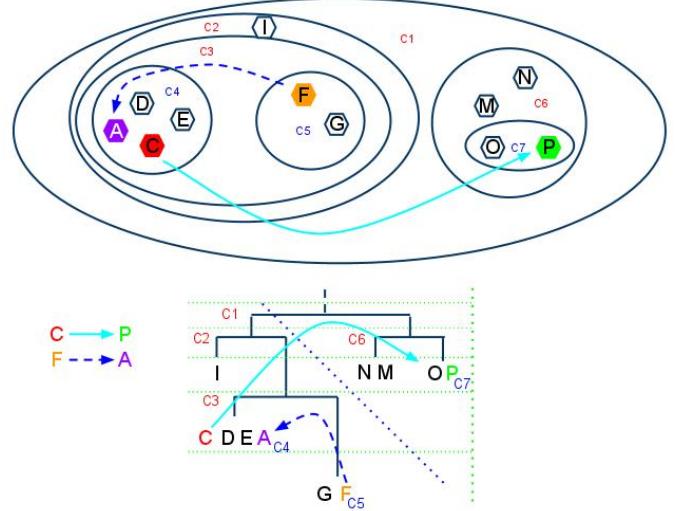


Figure 2: Simple example of hierarchical community structure showing grouping of nodes A to G , with communities identified $C1$ to $C7$.

it to the smallest community shared by M and P , which is $C6$. $C6$ is smaller than $C1$, so it will pass the message to M .

The intended effect of BubbleH, is that we continually narrow down the scope of the message, so that it reaches the smallest community or *target* possible. We believe that in doing so, a message has a better chance of reaching the destination node than by considering communities of any size and structure for message passing.

Hierarchical Community Finding

It has recently been observed that mobile phone users typically belong to several network communities, so a suitable community finding algorithm should be capable of detecting overlapping communities. For this reason, we develop H-GCE, a hierarchical community detection method based on GCE (Greedy Clique Expansion), which has been shown to perform well in networks with high levels of community overlap (Lee et al. 2010).

Before describing the algorithm, we define some terms. Let a community consist of a set of nodes. We will call a pair of communities C and C' *near duplicates* if the distance between them $\delta(C, C')$ is less than the value of ϵ , where

$$\delta(C, C') = 1 - \frac{|C \cap C'|}{\min(|C|, |C'|)}. \quad (1)$$

Given a set of communities, we can *merge* two communities by removing them from the set and adding their union.

H-GCE consists of two phases. The first involves identifying a set of seeds and expanding these such that a dendrogram is formed. To find seeds, we first enumerate the set of all cliques in the network, and then remove all cliques which have a larger near-duplicate, as in GCE. Thus, each seed is a clique that is not a near duplicate of a clique of equal or greater size. We then select all seeds of the smallest size and expand them in parallel by one node by greedily optimizing a local fitness function—see (Lee et al. 2010) for details. After each expansion step, we merge any pair of communities

which have become near duplicates. This two-step process (expanding all smallest communities by one node, merging any near duplicate pairs) is repeated until each seed has expanded to contain the entire graph. At some point in this phase all seeds will have been merged into one community. Thus, the expansion process produces a singly-rooted tree, or dendrogram, with seeds at the leaves and one big community containing all nodes at the root.

In this dendrogram, the path from a leaf (the seed) to the root is the history of nodes that are added either by greedy expansion or by merging. Thus, each point in this path can be thought of as representing a community containing the node added at that step in the expansion and all nodes added previously.

There is a problem with the dendrogram produced in phase one: even if a graph contains no community structure or only flat community structure, the procedure will return a dendrogram indicating hierarchical community structure—however, this dendrogram will be a mere artifact of the algorithm. The second phase of the algorithm determines which points in the dendrogram produced in the first phase represent significant community structure. The fundamental idea behind our approach is that if community structure is significant, then it will be robust against small perturbations in the graph, whereas insignificant communities will not display such stability. In other words, significant communities will be recoverable even if some noise is added to the graph; this idea has been previously developed in (Karrer, Levina, and Newman 2008; Rosvall and Bergstrom 2008). This and other approaches to measuring the significance of communities are outlined in Fortunato’s review of community finding methods (Fortunato 2010).

Phase one leaves us with a set of seeds and a dendrogram. The purpose of phase two is to solve the problem just identified by labeling the points in this dendrogram as significant or insignificant. Phase two consists of two steps: the first step produces several dendograms as in phase one but on graphs that have been perturbed by randomly rewiring ten percent of the edges in such a way that the original degree distribution is preserved. In our experiments we create 100 dendograms on graphs perturbed using the method outlined in (Karrer, Levina, and Newman 2008).

In order to make sure that all graphs contain the same seeds, we did not rewire edges within seeds. Thus, all of these dendograms share the same leaves and root as the one produced on the unperturbed graph. This means that as one traverses the path from a particular seed s to the root, the n th step of the traversal will have a corresponding point in every dendrogram, i.e., the n th parent of seed s .

Because each of these points corresponds to a community (as explained above), the distance of two corresponding points in two dendograms can be calculated using the distance measure in eq. 2. We call two corresponding points similar if the communities represented by these points have a distance less than 0.1.¹ The second step of phase two com-

¹As any two communities approach the size of the entire graph, they will become similar because they are forced to contain the same nodes. For this reason, we use a different similarity measure

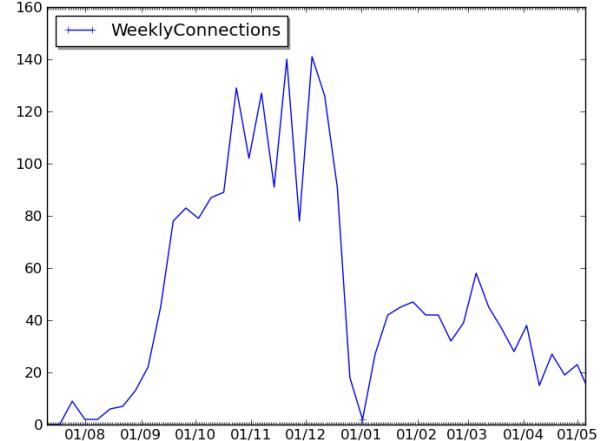


Figure 3: Average weekly number of connections for MIT Reality dataset.

pares each point n in the unperturbed dendrogram with the corresponding points n' in the perturbed dendograms. If n is similar to n' in at least 90% of the perturbed dendograms, then it is labeled as significant. This definition of significance is similar to the one used in (Rosvall and Bergstrom 2008). Finally, H-GCE returns the communities corresponding to all points in the original dendrogram which were labeled as significant in phase two.

Preliminary Evaluation

To evaluate the performance of BubbleH, we have used ContactSim, a discrete time event simulator designed specifically for evaluating DTN routing. ContactSim is capable of using contact traces either recorded during real-world experiments or generated synthetically.

In addition to BubbleH, we have also implemented several other routing algorithms. In order to compare the performance of using hierarchical community structure versus the original K-CLIQUE algorithm, we have implemented BUBBLERap. PROPHET, as one of the most well-known and mature DTN routing protocols, has also been implemented to provide a standard for comparison. Finally, we also show results for unlimited flooding of messages across the network: this gives us bounds on maximum *delivery ratio* and minimum *delivery latency* achievable.

In this experiment, we use Bluetooth proximity traces from the MIT Reality project (Eagle and Pentland 2005) to drive contact events within the simulator. The authors have captured communication, proximity, location, and activity information from 100 subjects at MIT over the course of the 2004-2005 academic year. This data represents over 350,000 hours of continuous data on human behavior.

The MIT Reality dataset has the most number of connections between Oct 2004 and Jan 2005 (Figure 3), so

for two communities if they both contain more than half the nodes in the graph, based on the proportion of nodes that they both exclude, as in

$$\delta(C, C') = 1 - \frac{|(G - C) \cap (G - C')|}{\min(|G - C|, |G - C'|)}. \quad (2)$$

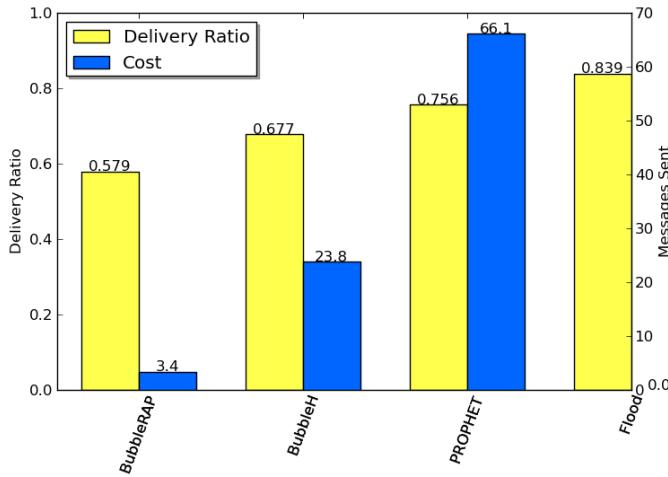


Figure 4: Simulation results for BUBBLERap, BubbleH, PROPHET and Flood during the *MIT-NOV* period.

we chose the period between 10 Nov 2004 and 10 Dec 2004 for community detection and testing, referred to as *MIT-NOV*. A weighted edge list is created from *MIT-NOV*, which is used by both BUBBLERap’s thresholded K-CLIQUE and H-GCE to create community structures. For K-CLIQUE the threshold parameter is 2 days, aggregated over the entire period, with $K = 3$. This creates 6 communities of average size 3 for KCLIQUE, H-GCE creates 20 communities of average size 15. At the start of the simulation, each node is initialized with a message for each other node.

The results in Figure 4 show that in this initial evaluation, BubbleH achieves a higher delivery ratio than BUBBLERap, but with a cost that is higher. PROPHET does better still in this time period, but with a much higher cost associated with it.

Not shown is the latency for BUBBLERap, BubbleH, PROPHET and Flood, the differences between them appear to be insignificant, apart from Flood, which as expected, gives the best possible latency, a few points better than the other candidates.

Conclusion and Future Work

We have presented methods for identifying hierarchical community structure and routing messages based on this structure. We have shown that the extra information on hierarchy can improve the efficiency of message delivery. In future work we plan to perform a more comprehensive evaluation to see if these improvements hold up. A key question for the significance of this work is the extent to which hierarchical community structure actually exists in real networks. This is an under-explored question and we plan to test this across a range of real social networks.

Acknowledgments

This work is supported by a PhD scholarship from the Irish Research Council for Science Engineering and Technology and by Science Foundation Ireland Grant No. 08/SRC/I140 (Clique: Graph and Network Analysis Cluster).

References

Crowcroft, J.; Yoneki, E.; Hui, P.; and Henderson, T. 2008. Promoting tolerance for delay tolerant network research. *SIGCOMM Comput. Commun. Rev.* 38(5):63–68.

Eagle, N., and Pentland, A. S. 2005. CRAWDAD data set mit/reality (v. 2005-07-01). Downloaded from <http://crawdad.cs.dartmouth.edu/mit/reality>.

Fortunato, S. 2010. Community detection in graphs. *Physics Reports* 486(3-5):75–174.

Hui, P.; Chaintreau, A.; Scott, J.; Gass, R.; Crowcroft, J.; and Diot, C. 2005. Pocket switched networks and human mobility in conference environments. *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking - WDTN '05* 244–251.

Hui, P.; Crowcroft, J.; and Yoneki, E. 2010. Bubble rap: social-based forwarding in delay tolerant networks. *IEEE Transactions on Mobile Computing*.

Karrer, B.; Levina, E.; and Newman, M. E. J. 2008. Robustness of community structure in networks. *Physical Review E* 77(4):046119.

Lee, C.; Reid, F.; McDaid, A.; and Hurley, N. 2010. Detecting highly overlapping community structure by greedy clique expansion. In *SNA-KDD 2010*, 33–42. Washington, DC: ACM.

Lindgren, A.; Doria, A.; and Schelén, O. 2003. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review* 7(3):19.

Musolesi, M., and Mascolo, C. 2009. CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks. *IEEE Transactions on Mobile Computing* 8(2):246–260.

Nazir, F.; Ma, J.; and Seneviratne, A. 2009. Time critical content delivery using predictable patterns in mobile social networks. In *Social Mobile Web (SMW'09) at 2009 IEEE International Conference on Social Computing*, 1066–1073. IEEE.

Palla, G.; Derényi, I.; Farkas, I.; and Vicsek, T. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435(7043):814–818.

Rosvall, M., and Bergstrom, C. 2008. Mapping change in large networks. *arXiv* 1–9.

Spyropoulos, T.; Psounis, K.; and Raghavendra, C. S. 2005. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, 252–259. New York, NY, USA: ACM.

Vahdat, A., and Becker, D. 2000. Epidemic Routing for Partially Connected Ad Hoc Networks.

Vallina-Rodriguez, N.; Hui, P.; and Crowcroft, J. 2009. Has anyone seen my goose? social network services in developing regions. In *Social Mobile Web (SMW'09) at 2009 IEEE International Conference on Social Computing*. IEEE. 1048–1053.