Quality Distributed Community Formation for Data Delivery in Pocket Switched Networks

Matthew Orlinski and Nick Filer
School of Computer Science
University of Manchester
Manchester, M13 9PL, UK
{orlinskm, nick}@cs.manchester.ac.uk

ABSTRACT

In this paper we look at ways of detecting groups of strongly related devices called communities which are present in mobile Pocket Switched Networks (PSNs). We use existing methods to detect communities which leverage repeated human movement patterns and "familiar strangers" within a number of real PSNs extracted from the CRAWDAD repository. By using different community detection techniques we attempt to show that there is a correlation between community size and compactness and inter-community membership of devices with increased data delivery. Finally our findings are implemented in a prototype protocol called Quality which creates larger communities with increased inter-community membership distributively.

1. INTRODUCTION

Pocket Switched Networks (PSNs) are complex wireless networks in which connections are formed opportunistically between participants who possess wireless devices. In contrast to modern telecommunications infrastructure, PSNs do not necessarily have a permanent, hierarchical structure which can be used to deliver data. In cases which we are concerned with in this paper, PSNs are made up entirely of mobile phones, laptops, or just about any ubiquitous device which carry a non-infrastructure, ad-hoc capable, wireless networking interface.

In purely mobile PSNs it should be possible to deliver data from any member of the network to another. However, the movement patterns of individuals and close-range wireless communication currently makes delivering data without loss challenging in the real world. Devices in PSNs are frequently unconnected, meaning that data delivery is probabilistic and normally much slower and more uncertain than, for example, in best effort infrastructure networks except in remote areas. However, purely mobile PSNs can continue to work when infrastructure is disrupted and can therefore provide an important alternative delivery method in many disaster scenarios.

Data forwarding in PSNs usually involves a trade-off between the use of resources such as power, buffer utilisation, and bandwidth with data delivery probability and delay. Relaying extra network information or duplicating packets can often improve delivery but can also have an adverse effect on battery life and wireless channel congestion.

It has been demonstrated that segregating networks seen in Reality Mining Experiments [4] into logical partitions can significantly improve data forwarding efficiency [5] in PSNs. However, considering the many different partitions which can be created by automatic means (Section 4), it is important to be able to predict if the community partitioning being used is beneficial to data delivery. In many cases data delivery based on automatic community detection can yield bad results (Section 5.1).

In this paper we will analyse real world communities formed by data-sets from the CRAWDAD repository [1, 4, 10]. Our aim is to separate the good community partitioning from the bad, in ways which affect data forwarding efficiency in oblivious forwarding schemes. Finally, our findings are implemented in a prototype protocol called Quality (Section 6) which attempts to lower the number of distinct communities and increase inter-community membership in order to deliver more messages successfully.

2. COMMUNITY DETECTION

From Reality Mining experiments, "Familiar Strangers" [8] can be grouped together into closely related clusters. These clusters can be called "Communities" in PSNs, and can be detected by retrospective and centralised algorithms such as the K-medoids and Weak Clustering techniques [11]. Communities can also be detected in real-time using distributed techniques, in which members of the network work together to form social groups. This is called Distributed Community Detection and when applied to PSNs, enables the devices to discover their own communities without centralised control.

2.1 Distributed Community Detection

To the best of our knowledge, the performance of distributed community detection techniques relies heavily on user specified variables which require advanced knowledge of the scenarios where the algorithms are to be deployed. Seemingly endless variation in partitioning can be achieved by retrospectively tweaking variables to suit different experiments (See the variation of performance in Figure 1 in Section 5.1).

A common user defined variable is the "Familiar Threshold". Familiar Thresholds are used by devices to pick and

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIMPLEX '12, April 17 2012, Lyon, France
Copyright 2012 ACM 978-1-4503-1238-7/12/04 ...$10.00.

choose other devices which they encounter most often. In order to store communities, each device has a Local Community view which it stores in memory. Due to each device having its own Local Community and depending on the clustering algorithm used, globally a device may be in many different communities on many different devices. The following distributed community formation algorithms were used to generate the Local Communities with which we assessed oblivious forwarding performance in this report.

2.1.1 Simple

Simple is a distributed community detection method developed by Hui et al. [6] which can be used to form communities within a dynamic PSN. Simple works by building communities of devices which share a large intersection of encountered devices. The size of intersection is dictated by the user provided variable, λ. By altering λ, Simple can produce a wide range of communities so that studying the relationship between different communities and data delivery can be undertaken [1].

2.1.2 Promote

In Simple, Familiar Devices, or devices which have the highest cumulative connection times should include each other in their Local Communities. Currently this is not guaranteed to happen in Simple, as joining a Local Community is based purely on the intersections of "Neighbour Tables" of two connected devices. If a Local Community is to reflect the connectivity within a PSN more realistically and include Familiar Devices, a secondary promotion method is needed in addition to the community formation techniques found in Simple. For this reason we devised Promote, which extends Simple by adding a secondary "promotion" method.

Algorithm 1 Promote
input:
User defined Familiar Threshold, γ
Local community, C_0
Remote device, i
Total time t, that local device has encountered device i
Remote device encountered most often (Familiar Device), p

```
if $t \geq \gamma$ then
  if $i \notin C_0 \wedge i = p$ then
    $C_0 = C_0 \cup i$
  end if
  The rest of the Simple logic...
end if
```

The extra logic in Promote works by looking for devices with the highest cumulative connection times. When Familiar Devices encounter each other, they will add each other to their Local Communities if they are not already members. This results in larger communities than the Simple algorithm which seem to better reflect the connectivity between devices.

3. EXPERIMENTAL ENVIRONMENT

Great care must be taken when performing Reality Mining experiments in order to ensure that wireless range and enquiry intervals are consistent. Otherwise, an unrealistic view of communications rather than purely social communities may be gained. Furthermore, reliance on simulated movement patterns and inferred connections between devices could result in unrealistic movement patterns and connection durations. To address this concern we have extended The One Simulator v1.4 [7] to use trace files from a number of real data-sets from the CRAWDAD repository [2].

The three data-sets used are shown in Table 1: An experiment from LocShare labelled UCL1 [1]; InfoCom 05 [10] and finally Cambridge city [9] from the Haggle project. In each case, external and long range devices present in the data-sets have been removed to concentrate on communities formed solely by ad-hoc mobile wireless users.

3.1 Data Delivery

Data delivery in PSNs is opportunistic, which means data can only be passed between devices if and when they are in range of one another. As devices are carried by users who are free to roam, being in contact with a particular other device is seldom the case. Table 1 shows the "Daily Degree Centrality", the probability of meeting a particular device in a day for each data-set. Note that this value is never high enough to guarantee a daily contact between 2 devices in any of the data-sets.

To compare community based data delivery against other methods, we have calculated the best and worst case scenarios for the data-sets in Table 1. The best case is provided by flooding the network with copies of a message with no attention paid to congestion in the wireless medium, and with each device possessing a large message buffer. In the worst case there is no attempt at routing and messages are only delivered directly to their final recipients. This approach is called "Wait" and is used in place of no delivery at all, which would obviously yield a delivery probability of 0 and give no insight into the benefits of communities.

	Infocom5	Cambridge	UCL1
Duration (Days)	3	12	6
Mobile Devices	41	36	20
Number of Connections	28216	21239	512
Daily Degree Centrality	0.78	0.24	0.53
Message TTL (Hour)	1		
Global Message Frequency	120 messages created per hour		
Transmit Speed	250kBps		
Message Size	1KB		
Buffer Size	5MB		
Best Case Delivery Ratio	0.92	0.98	0.61
Worst Case Delivery Ratio	0.56	0.39	0.12
Worst Case Cost	38.97	34.03	16.22

Table 1: Comparison of data-sets with simulated data delivery performance from Epidemic and Wait.

The Time To Live (TTL) of messages is tailored to the sparsity of the data-sets rather than a specific application, and in all cases here a TTL of 1 hour is used. For data delivery, users are accustomed to relatively rapid delivery in seconds, minutes or hours; seldom days or longer. The aim

[1] Our implementation of Simple to be used in conjunction with The One Simulator [7] is available for download at http://apt.cs.man.ac.uk/projects/wireless/SimpleRouter.java

[2] A copy of Connection Mode can be obtained from http://apt.cs.man.ac.uk/projects/wireless/connections.rar

is to keep TTL as low as possible so as to suggest useful applications such as messaging for future use.

3.2 Measuring Delivery Cost

Delivery cost is measured using the ratio of the total number of messages transferred between devices to those successfully delivered within the message Time To Live (TTL). This is a measure of the inefficient use of duplicated packets. Whilst not accounting for all costs this method is sufficient to measure delivery efficiency as lower cost implies fewer copied packets produced and transmitted which leads to less energy consumption.

4. PROPERTIES OF DISTRIBUTIVELY DETECTED COMMUNITIES

So that we may discuss how different community partitions affect date delivery, we must first understand how communities are formed in detail. In this section characteristics of communities detected by distributed formation algorithms are described, starting with the devices themselves;

Distributed Community Characteristic 1. *Each device running a distributed community detection algorithm will belong to at least 1 community.*

As devices are not guaranteed to ever come into contact in PSNs, they may not have the opportunity to form communities larger than themselves. Requiring that devices belong to at least 1 community means that the rule for the number of communities created is:

Distributed Community Characteristic 2. *n devices running distributed community detection algorithms will create n, sometimes identical, non-empty communities. The communities which are created are called the Produced Communities, $\mathbb{P}$ with each member in $\mathbb{P}$ being one of the $2^n - 1$ Possible Communities.*

Each Produced Community in $\mathbb{P}$ will contain at least the device ID of the device it was created by and possibly any other ID from the n devices. Therefore the communities produced are all none-empty. The cardinality $2^n - 1$ of the Possible Communities comes from the cardinality of the power set of devices in the PSN, minus the empty set – as no Produced Communities can be empty.

Distributed Community Characteristic 3. *The Produced Communities from n devices are a multi-set of community sets with a cardinality of n.*

The Produced Communities in $\mathbb{P}$ can either be identical, or different. Hence why $\mathbb{P}$ is a multi-set of communities. From $\mathbb{P}$ we can extract all the distinct sets, minus the duplicates. The distinct sets are called Natural Communities.

Distributed Community Characteristic 4. *Within $\mathbb{P}$ there is a family of Natural Communities called $\mathbb{N}$:*

$$\mathbb{N} = \{x : x\, is\, a\, community\, in\, \mathbb{P}\} \tag{1}$$

More properties of Natural Communities to note are; Every device is guaranteed to be in at least one Natural Community, and devices can belong to multiple Natural Communities at the same time. The final observation to be made about the characteristics of Local Communities is that some can be contained within others.

Distributed Community Characteristic 5. *Natural Communities can be encompassed by others. The set of the largest Natural Communities minus any sub-sets is called the family of Community Super-Sets, $\mathbb{S}$.*

Again, some further properties of Community Super-Sets to be aware of are: There can be multiple Community Super-Sets reported by a distributed community detection algorithm; A Community Super-Set may encompass one or more Natural Communities; Every device will be in at least one Community Super-Set.

4.1 Community Validation

The community labelling scheme mentioned in the previous section can be used with different cluster validation techniques in order to perform analysis on communities within a PSN.

$$PC(c) = \frac{1}{N} \sum_{i=1}^{c} \sum_{j=1}^{N} (\mu_{ij})^2 \tag{2}$$

The Jaccard Index is used to assess the similarity of clusters, and has been used previously in PSN research to benchmark distributed community detection against centralised algorithms [6]. Dunn's Validity Index is a ratio of within cluster and between cluster separations which can also be applied to communities. A high Dunn's Validity Index can identify clustering which is compact and well separated. To measure mutual inclusion between communities, or "fuzzy" communities – the Partition Coefficient (2) can be used.

4.2 Data Delivery Using Communities

The algorithms in Section 2.1 were designed for community discovery, not for data delivery. Few data delivery schemes for PSNs are transferable to hierarchical structures which communities offer, and there are few shared characteristics between data delivery schemes which we can use to assess communities for data delivery. However, one characteristic which many data forwarding schemes do share is flooding, and duplication of data. With this in mind we have produced an oblivious forwarding scheme aimed at providing a control case (null hypothesis) which uses message duplication and flooding to look for important community characteristics. The simple flooding technique shown in Algorithm 2, aims to flood an encountered community containing the destination device with copies of the data as much as possible.

Algorithm 2 Community Based Data Delivery

The Local Community of a particular device i is denoted C_i

input:

A list of all devices currently in range, E

A data packet called m

The final destination of m is d

```
for all E as i do
  if i = d then
    $DeliverMessage(m)$
  else
    if $d \in C_i$ then
      $CopyMessageToEncounteredNode(m)$
    end if
  end if
end for
```

5. QUALITY COMMUNITIES

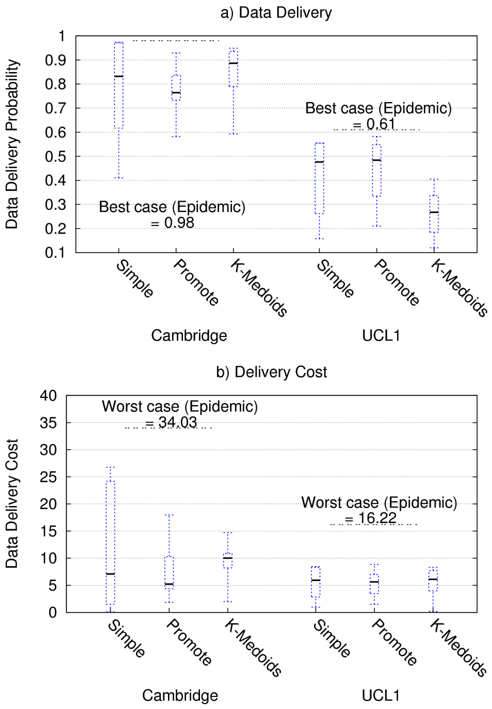

Figure 1: Delivery probability and cost variance of different communities in Cambridge and UCL1.

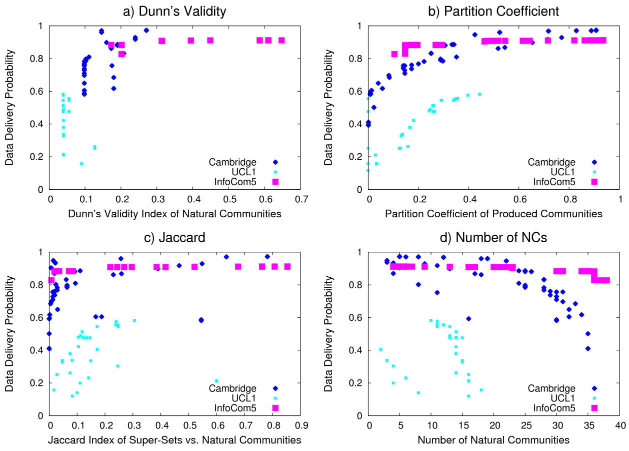

Figure 2: Community measurements.

	PCC for data-sets		
	UCL1	Cambridge	InfoCom5
Dunn's Validity	-0.43	0.67	0.61
Partition Coefficient	0.95	0.88	0.80
Jaccard	0.03	0.37	0.68
Number of NCs	-0.05	-0.75	-0.77

Table 2: Pearson Correlation (PCC) of community measurements against data delivery.

Using The One Simulator and the data-sets from CRAWDAD, we will now investigate how the "quality" of community configuration affects delivery in PSNs. Figure 1 shows how data delivery rates can vary significantly with differing choices of variables in the Simple, Promote, and K-medoid algorithms. All of these algorithms have one thing in common, they develop communities based on the frequency of encounters between devices. For Simple and Promote we measured for a wide range of λ values between 0 and 1 with a resolution of one hundredth of a second, and integer values between 3 and 16 for the K-medoid technique.

5.1 Community Compactness And Separation

To discover if there is a significant relationship between data delivery and the separation of communities, one can adopt the qualitative Dunn's Validity Index (Section 4.1).

Results for Dunn's Validity Index of the Natural Communities produced on our 3 different data-sets are presented in Figure 2a. Pearson Correlation Coefficient (PCC) is used to measure the linear dependence between Dunn's Validity Index and data delivery. PCC is positive in the Cambridge and InfoCom5 data-sets shown in Table 2, although there exists a negative PCC for the UCL1 data-set. This discrepancy suggests that there is more to data delivery than simply discovering isolated communities.

5.2 Inter-community membership

Inter-community membership is only possible when communities share devices in common. This is sometimes called "Fuzzy clustering". Produced Communities created by distributed community detection algorithms running on multiple devices can have this "fuzzy" property because devices form community views independently. As we saw in Section 4, devices will always belong to at least one Produced Community and up to n more. As a consequence all the classifications of communities produced by distributed means are "fuzzy" and not "crisp".

The results from the previous Section 5.1 were inconclusive, showing that community segregation alone can not predict network performance. This is partly due to the fuzzy nature of Produced Communities, which we have measured using the Partition Coefficient for community overlap (Section 4.1) in Figure 2b. Table 2 shows communities with a greater degree of overlap give the greatest probability of data delivery with a PCC consistently higher than 0.8.

5.3 Agreement Between Devices

Another factor for data delivery could be the level of agreement between devices. As Local Communities are created by devices independently, there is scope for vastly different Local Community views, even between devices which have each other in their Local Communities. To see what the extent of this variance is and how it affects data delivery we have plotted the Jaccard Index (Section 4.1) of Community Super-Sets against the Natural Communities produced in the same experiments. Here, the Jaccard Index assesses the similarity between all the Natural Communities and the largest communities reported by devices – and therefore the level of agreement within the network.

The data in Table 2 and Figure 2c shows that an increased level of disagreements between devices actually improves data delivery based on the positive PCC observed. However, a significant PCC is only observed in the InfoCom5 data-set.

5.4 Network Fragmentation

To further investigate how disagreements between devices can affect data delivery, we looked at the number of Natural Communities in the PSNs. Table 2 and Figure 2d show a negative relationship between the number of Natural Communities and data. In other words the fewer Natural Communities there are, the greater the data delivery should be.

The implication of this finding, along with device conformity, is that devices should negotiate shared local community views where appropriate. This lowers the number of distinct Natural Communities across the network, but care should be taken to ensure inter-community overlap is not destroyed completely.

5.5 Heterogeneity Of Community Popularity

In previous work by Hui et al. [5] it was observed that pre-existing schemes for passing data within PSNs are inefficient because they assume that every node is statistically equivalent and homogeneous. A PSN is an example of a “small world network” with some devices having more connections to other devices than others [3] causing large variances in “betweenness centrality”.

In a similar way to inter-device betweenness centrality we want to know whether communities exhibit a similar behaviour. Is the heterogeneity of popularity of communities based on anything other than the number of connections, for example on unbounded size [2]?

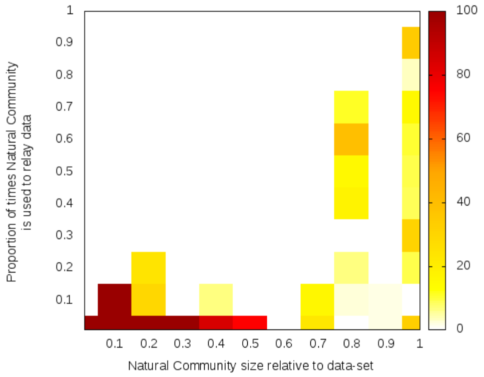

Figure 3: Which Natural Communities are used to relay packets.

The number of times that a Natural Community is used to relay data can be counted. Using the experiments across the data-sets a picture emerges about which Natural Communities are used more frequently which is presented in Figure 3. The heat in this graph shows cross experiment frequency of Natural Community size vs. the proportion of times they were used to relay data, with the majority of Natural Communities used being less than half the size of the total PSN.

The commonness of smaller Natural Communities across PSN experiments can be used as an explanation for the performance variation encountered in Figure 1. Figure 4 shows delivery ratio and cost rising with natural community size. A delivery ratio close to that of Epidemic’s (0.98 shown in Table 1) is achieved when Natural Community size is half that of the PSN. This suggests that Epidemic routing can achieve its best result by utilising half of the total data-sets, which could have repercussions for future protocol design.

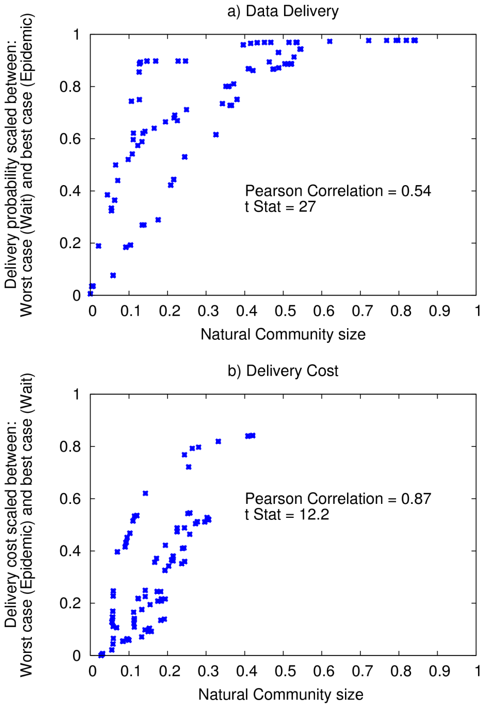

Figure 4: Delivery ratio and cost based on Natural Community size.

In general we can not conclude that community popularity is heterogeneous, because it depends on the size of community. However the larger the community, the more likely it is to contain the target destination – and the higher the heterogeneity of popularity of the community tends to be compared to other smaller communities.

6. NEW PROTOTYPE – QUALITY

To demonstrate how inter-community isolation, community size and overlap can be controlled distributively, we have implemented a simple prototype protocol called Quality to compare against Promote and Simple. Quality aims to reduce the number of Natural Communities created by using a more active community merging procedure than these other protocols. It also introduces overlap in Produced Communities which have no devices in common by always adding Familiar Devices regardless of Local Community comparisons. An overview of the algorithm which takes place on a device during an encounter can be found in Algorithm 3.

6.1 Results

Quality offers more consistent performance than Simple or Promote because it is less dependent than either protocol on user defined variables. Figure 5a shows Quality consistently performing better than the other community detection algorithms albeit at a higher cost, shown in Figure 5b . Quality does this by introducing overlap and forcing devices to merge their local community views more often.

```
Algorithm 3 Quality Community Formation
input:
User defined familiar threshold, $\gamma$
User defined community inclusion threshold, $\lambda$
Local community, $C_0$
Remote device, i
Local community of i, $C_i$
Total time, t local device has encountered i
Remote device encountered most often (Familiar Device), p

if $t \geq \gamma$ then
  if $(i \notin C_0 \wedge i = p)$ then
    $C_0 := C_0 \cup i$
  end if
  if $(|C_0 \cap C_i| > (\lambda * |C_0 \cup C_i|))$ then
    $C_0 := C_0 \cup C_i$
  end if
  if $(|C_0 \cap C_i| = 0)$ then
    $C_0 := C_0 \cup i$
  end if
end if
```

7. CONCLUSIONS & FUTURE WORK

A study of data delivery via communities formed in PSNs has been presented. It shows that data delivered by community formation algorithms is not reliant on any single factor. However there are a number of tests which can be used to identify whether community partitions are suitable for data delivery. The findings reiterate that Epidemic routing for PSNs is very inefficient but by partitioning the devices in the following ways efficiency can be improved;

1. Natural Communities should be "fuzzy" (Section 5.2) but not to the detriment of compactness and good community isolation (Section 5.1).
2. Disagreements between devices are essential to provide "fuzzy" communities. However too much disagreement and too many Natural Communities can harm data delivery (Sections 5.3 & 5.4).
3. Natural Community popularity can be heterogeneous, with a strong linear dependence between Natural Community size and the number of packets it can relay (Section 5.5).

This list does not claim to be exhaustive for all the factors associated with community based routing. This work has gone some way to highlighting the need for further development of community detection in PSN data-sets. The findings will prove useful when developing distributed community detection algorithms with data routing in mind.

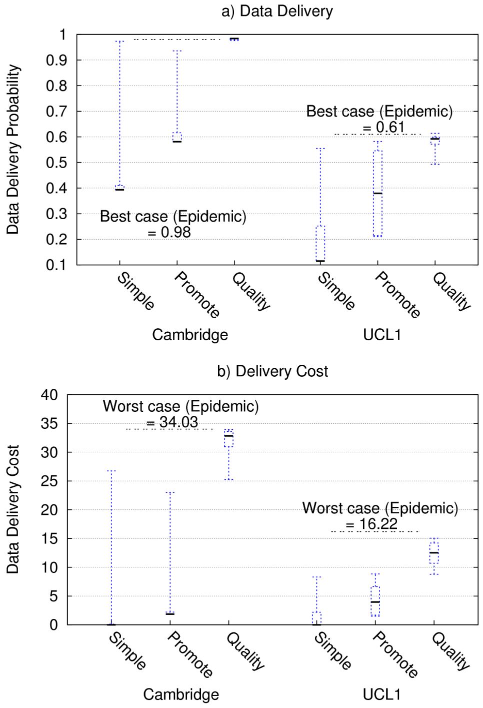

Figure 5: Variance of data delivery and cost.

8. REFERENCES

[1] F. Abdesslem, T. Henderson, and I. Parris. *CRAWDAD data set st_andrews/locshare (v. 2011-10-12).* October 2011.

[2] F.B. Abdesslem, A. Ziviani, M.D. de Amorim, and P. Todorova. Looking around first: Localized potential-based clustering in spontaneous networks. *Communications Letters, IEEE*, 11(8):653–655, 2007.

[3] E.M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40, 2007.

[4] N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. *PNAS*, 106(36):15274–15278, 2009.

[5] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE rap: Social-based forwarding in delay tolerant networks. 6(1), January 2007.

[6] 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, 2007.

[7] A. Keranen, J. Ott, and T. Karkkainen. The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, page 55, 2009.

[8] S Milgram. *The Familiar Stranger: An Aspect of Urban Anonymity.* Mcgraw-Hill Book Company, 1st edition, 1977.

[9] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. *CRAWDAD trace cambridge/haggle/imote/content (v. 2006-09-15).* September 2006.

[10] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. *CRAWDAD trace cambridge/haggle/imote/infocom (v. 2006-01-31).* January 2006.

[11] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Third Edition.* Academic Press, Inc., Orlando, FL, USA, 2006.