ARTICLE IN PRESS

Ad Hoc Networks xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

The rise and fall of spatio-temporal clusters in mobile ad hoc networks

M. Orlinski *, N. Filer

School of Computer Science, University of Manchester, Manchester M13 9PL, UK

ARTICLE INFO

Article history:
Received 7 September 2012
Received in revised form 17 December 2012
Accepted 2 March 2013
Available online xxxx

Keywords:
Mobile ad hoc networks
Pocket switched networks
Temporal contact graphs
Spatio-temporal clustering
Opportunistic routing

ABSTRACT

Cluster detection has been widely applied to the problem of efficient data delivery in highly dynamic mobile ad hoc networks. By grouping participants who meet most often into clusters, hierarchical structures in the network are formed which can be used to efficiently transfer data between the participants. However, data delivery algorithms which rely on clusters can be inefficient in some situations. In the case of dynamic networks formed by encounters between humans, sometimes called Pocket Switched Networks (PSNs), cluster based data delivery methods may see a drop in efficiency if obsolete cluster membership persists despite changes to behavioural patterns. Our work aims to improve the relevance of clusters to particular time frames, and thus improve the performance of cluster based data delivery algorithms in PSNs. Furthermore, we will show that by detecting spatio-temporal clusters in PSNs, we can now improve on the data delivery success rates and efficiency of data delivery algorithms which do not use clustering; something which has been difficult to demonstrate in the past.

1. Introduction

Cluster detection has been an essential part of a data analysts' toolkit ever since Sokal and Sneath first refined it in the field of numerical taxonomy in the early 1960s [1]. Since then, many distributed cluster detection techniques have been developed [2–4] and applied to the problem of opportunistic data delivery in highly dynamic Mobile Ad hoc Networks (MANETs) where the probability of a device being able to deliver a packet is unknown or difficult to calculate.

One example of a highly dynamic MANET where connections between participants are often short lived and difficult to predict is sometimes called a Pocket Switched Network (PSN) [5]. PSNs are created by personal mobile devices carried by humans forming opportunistic connections with each other over short range wireless interfaces such as Bluetooth and Wi-Fi. As a result, end-to-end paths between participants in PSNs are relatively unstable when compared to other types of MANETs due to link quality [6] and the different movement patterns of participants [7].

By providing data-sets containing encounters between personal mobile wireless devices, some recent Reality Mining experiments [8] provide researchers with a valuable resource with which to explore the possibilities of PSNs. By analysing Reality Mining data centrally, or by taking into account all previous encounters using distributed methods, people and/or their devices can easily be grouped together to form aggregated clusters. In this paper, aggregated clustering refers to clustering based on all of the available data, without looking at the situational relevance or time passed since the data was collected. Thus, if cluster size is not controlled, aggregated clustering can give rise to monotonically increasing cluster sizes over time [4], and obsolete cluster memberships can persist if movement patterns change [9]. Even if clusters in dynamic networks are given an upper bound for size as in budget-based clustering [10], it is not easy to infer temporal infor-

* Corresponding author.
E-mail addresses: orlinskm@cs.manchester.ac.uk (M. Orlinski), nick@cs.manchester.ac.uk (N. Filer).

1570-8705/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2013.03.003

mation from the resulting clustered data. Therefore, data delivery methods which rely on aggregated or monotonic clustering techniques to pass data between participants in PSNs can suffer efficiency losses as packets are duplicated along obsolete or slower paths.

With the help of Fig. 1 and a simple example, we will now attempt to define this problem in a little more detail. Imagine that d_i in a PSN wishes to send a message to another called d_k, but does not know the exact location or have the ability to find the quickest path to d_k. Because end-to-end paths are unstable or unlikely in PSNs [11], MANET routing protocols such as AODV [12] and OLSR [13] cannot be used, as flooding route discovery packets may not be able to identify an end-to-path at a particular time, or paths change more often than they are discovered. However, information such as a cluster label [14] which identifies the cluster to which a device belongs could be easily obtained on an opportunistic basis from directly connected devices. Then the process with which d_i might get a message to d_k could be summarised as follows:

1. Device d_i wants to send a message to d_k. d_i comes into contact with device d_j from cluster B who reports that its cluster also contains d_k. So d_i passes a copy of the message to d_j. d_i does not need to duplicate this message further unless it meets another device from cluster B.
2. Device d_j does not have a direct link to d_k according to the graph in Fig. 1. However, it knows it belongs to the same cluster and copies the message to whoever it encounters within the cluster boundaries. This process is repeated until the message is delivered to d_k.

Upon consideration of this simplistic example, it may be apparent that the number, size, and membership of the clusters will impact upon message duplication. In a more detailed description of the problem there is also the added complication that clusters must be generated by devices themselves using distributed cluster detection algorithms,

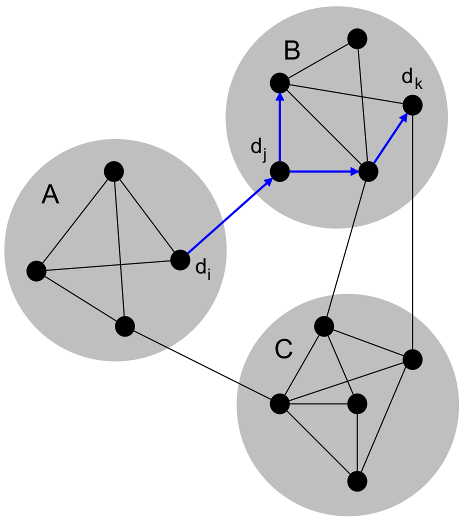
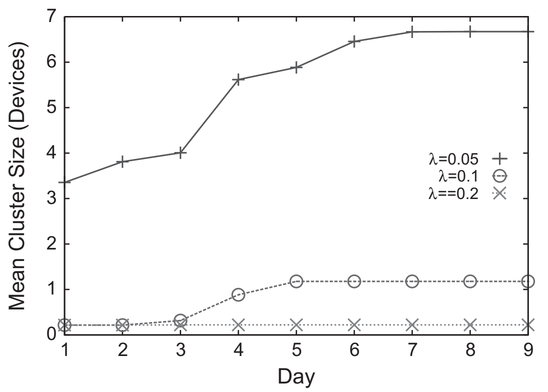


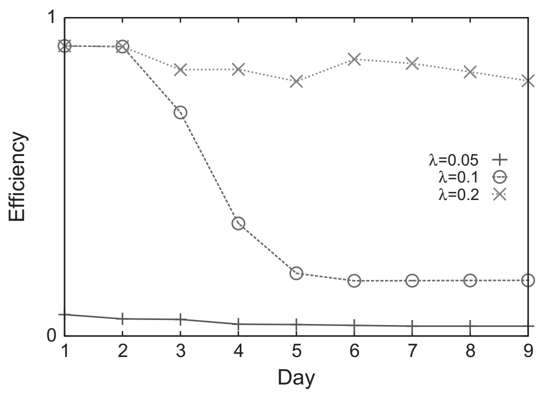
Fig. 1. Data delivery example using clusters to limit data duplication in MANETs where paths are unknown.

and confirmation that a message has been delivered may not be able to be sent across a wide area. Indeed, when the Quality distributed cluster detection and data delivery algorithm [4] is used, the delivery cost per message increases linearly with cluster size. Like the cluster detection in Quality, the Simple [15] distributed cluster detection method produces monotonically increasing cluster sizes. In Simple, λ is one of the parameters used to govern cluster membership, and controls how fast clusters grow. In the example in Fig. 2a, mean cluster size can be seen to increase monotonically or not at all depending on the λ value chosen. Using the same clusters produced in Fig. 2a, the cluster based data delivery algorithm Bubble [15] exhibits very low efficiency when λ is below 0.2 as shown in Fig. 2b.

When using either Simple or Quality, cluster size can be seen to increase monotonically in many other Reality Mining data-sets because of the densification [17,18] over time in graphs generated from the encounters between devices. Due to this densification and the changing movement patterns of humans, obsolete membership within clusters persists for the duration of experiments. Therefore, clustering algorithms used for data delivery should take into account temporal considerations such as; Do some participants only meet during certain times of the day? How likely are participants to meet again?



(a) Cluster size over time using Simple.



(b) Data delivery efficiency using Bubble.

Fig. 2. Cluster size and data delivery efficiency (packets delivered/relayed) over time using Simple, and Bubble using the same clusters to deliver data with the Cambridge Reality Mining data-set [16].

In 2008, Hui [9] stated that spatio-temporal clusters may only be valid for a particular time, such as during a conference or meeting with friends. He also stated that the current distributed cluster detection algorithms cause spatio-temporal clusters to be lost entirely due to aggregation of individual encounters between participants. Our approach in this paper, called Distributed Rise And Fall spatio-Temporal clustering (DRAFT) presented in Section 3, aims to detect the lost spatio-temporal clusters by allowing clusters to decay over time. As part of our analysis we will look at the spatio-temporal clustering behaviour of participants in Reality Mining experiments in Section 4, and then show how spatio-temporal clusters can improve on the long term efficiency of cluster based data delivery algorithms in Section 5.

1.1. Related work

Considering temporal behaviours along with clustering can give us new insights into cluster characteristics and relevance. Recent work from Pietilainen and Diot [19] has identified a number of clusters which occur within short time frames, and found a correlation between clusters that occur within several time frames, and social characteristics such as friendship and home city.

Pietilainen and Diot also went onto show that devices that spend the most time within these *social clusters* do not impact on data delivery performance metrics as much as other devices. Moreover, Gaito et al. [20] have shown that less than 10% of online friends met during experiments with students. So methods which hope to bootstrap the clustering process using social clusters generated from social networking websites [21] may not provide an improvement to data delivery efficiency in PSNs.

Contacts between non-social devices, also known as vagabonds [22], significantly outnumber social contacts and therefore have a greater collective effect on data delivery in PSNs [19]. Therefore, it may be reasonable to suggest that non-social or spatio-temporal, clusters which include social and non-social devices, may lead to more efficient data transfer.

Two of the most advanced data delivery schemes for PSNs that have the capability to consider transient social as well as non-social links are PRoPHETv2 [6] and Bubble [2]. PRoPHETv2 consistently performs well in a variety of simulated MANETs as well as in the Networking for Communications Challenged Communities (N4Cs) deployment [6]. This makes PRoPHETv2 the protocol to beat in terms of data delivery and efficiency. PRoPHETv2 uses the history of previous encounters to estimate delivery predictability for messages and gives more weight to recent ties. PRoPHETv2 works because individual devices have different chances of delivering a particular message, but it does not use the clustering paradigm where messages are duplicated amongst cluster members.

It was shown in the N4C experiment that some of the parameters PRoPHETv2 needs to function correctly (typical inter-connection times and a suitable constant for delivery predictability ageing [6]) are difficult to calculate for different areas of the network. The N4C network also highlighted the Parking Lot Problem in which many short encounters are separated by short time periods, whilst human movement patterns dictate longer durations. These short encounters are attributed to poor Wi-Fi connections, and are the justification for using cumulative encounter times rather than single encounter times when deciding whether to include devices in clusters later on in Section 3.

Like PRoPHETv2, Bubble [2] contains a directional routing protocol in which paths to destinations are found by *bubbling* data through the network. Bubble uses global centrality (which is difficult to estimate distributively [23]) as a guide for the bubbling process, but unlike PRoPHETv2, Bubble uses clusters provided by distributed cluster detection algorithms such as Simple to prune the epidemic [24] distribution tree once messages reach a cluster containing the destination.

To ensure high delivery success rates, messages can be duplicated within clusters that grow quickly. The Quality [4] distributed cluster detection and delivery mechanism depends on the identification of pairs of devices with high cumulative encounter times to each other. These well connected pairs add each other to their respective *local clusters*, which are an individual's view of the cluster to which they belong held in local memory. If devices with different local clusters meet and exchange their local clusters, a wider view of the network is seen by both devices. Local clusters help to prune the epidemic distribution tree of the network because messages will only be copied to devices which contain the message destination in their local cluster. Quality produces very large monotonically increasing local clusters, and duplicates messages across a large number of devices in order to ensure delivery. As a result, Quality has a tendency to deliver more messages than Bubble, but suffers from poor efficiency, especially when run for long periods [18].

Work by Borgia et al. [25] proposed a temporal adaptation to the Simple distributed clustering algorithm [15] which can be used to inhibit local cluster growth. Their proposal called AD-Simple, relies on pruning clusters of obsolete members using a timer which counts down from the moment devices are entered into local clusters. However, AD-Simple maintains *home* clusters for long periods, thus AD-Simple may not be suitable as a purely spatio-temporal approach to cluster detection.

Another distributed clustering mechanism often used is epidemic label propagation [26]. Like AD-Simple, the cluster sizes produced by epidemic label propagation are not monotonic. However, they can sometimes suffer from the monster cluster problem where a single cluster evolves to dominate the entire network. SHARC [27] prevents monster clusters from forming but suffers from the *wandering cluster* problem which is caused when large groups of devices propagate their cluster labels elsewhere.

The work in this paper will focus on the detection of local spatio-temporal clusters which are made up of both social and non-social devices, and which are only relevant to a particular space and time. The approach is similar to that of our recent Distributed Expectation-Based Spatio-Temporal (DEBT) clustering algorithm [18] where time is split into a number of discrete time frames in order to judge connectivity using cumulative encounter times. However, increasing data delivery rates whilst maintaining efficiency

Please cite this article in press as: M. Orlinski, N. Filer, The rise and fall of spatio-temporal clusters in mobile ad hoc networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.03.003

in DEBT proved difficult. This paper details our newest approach where clusters are non-monotonic and devices cooperate to remove others from the spatio-temporal clusters. This approach will be shown to be able to compete with PRoPHETv2 on both data delivery success rates and efficiency.

2. Temporal data in Reality Mining experiments

Spatio-temporal clusters were extracted from the Reality Mining data-sets Infocom5, Infocom6, Cambridge [16], and Reality [28]. Each of these is available from the CRAWDAD repository [1] and each has been converted to run in The One Simulator [29]. As the encounters within these Reality Mining experiments are recorded using Bluetooth, they are not symmetric [30]. However, because this paper is exploring what might be possible using PSNs [31], the data-sets are used to represent "data transfer opportunities that each of the participants would have, if they were equipped with devices which are always-on and always carried" [16]. Furthermore, when using Bluetooth on modern smartphones, there is often a period of time where user interaction is needed to "pair" devices so that data can be exchanged. This is obviously not ideal for PSNs which would require autonomous networking; so that consideration is also dropped from our analysis. We believe these assumptions are not unrealistic. In the near future, autonomous ad hoc network technologies, perhaps even Bluetooth Scatternets [32,33], will become more widely available. These will also offer faster device discovery [30] so that data can be exchanged symmetrically between nearby devices for most of the duration of the encounter.

Many observations about the temporal information in the current Reality Mining data-sets have been made previously. Henderson et al. [7] showed that encounter patterns change hour-to-hour and day-to-day, yet human movements are often repeated on a day to day (diurnal) or week to week basis. The daily habits of individuals also cause changes to measurements such as degree centrality, closeness, geodesic betweenness, shortest-path, and fastest-path when analysing traces from rush hour or non-rush hour traffic [34].

The diurnal movement patterns of the human working day affect the probability of a meeting between devices and total encounter times at different times of the day [35]. Furthermore, the distribution of inter-contact times in the data-sets has been shown to differ greatly in 3 h long time frames [36], and the global centrality of devices in 6 h time frames was shown to correlate well with a device's global centrality when calculated across an entire experiment [2]. Table 1 expands on this by showing the changing probability of a meeting between any two devices within 6 h time frames. We only studied the internal devices from each data-set and did not include external devices found by participants in our analysis. This was for two reasons; Firstly, it was unclear which external devices were static and which were mobile, and secondly, some of the external devices found were only seen once. The different probability for each quarter day confirms the observations made by Leung et al. [34] that network measurements change at different times of the day, and it hints that cluster membership may also be different if calculated separately in each time frame.

2.1. Dynamic encounter graphs

PSNs are made up from many personal mobile devices forming opportunistic, ad hoc connections between themselves. In each of the Reality Mining data-sets used to describe PSNs in this paper, new encounters arrive in a bursty fashion, with periods of high activity lasting up to 12 hours often followed by much quieter periods, as illustrated by Fig. 3.

Despite this temporal behaviour, encounters within PSNs and MANETs are commonly expressed spatially using aggregated contact graphs wherein devices are represented as vertices and pair-wise encounters between devices are shown as edges. In aggregated contact graphs, edges often contain information from many separate encounters and details such as the bursty behaviour in Fig. 3 are missing.

In a dynamic encounter graph [37], encounter data is split into short sequential time frames. Each time frame contains an aggregated form of a contact graph albeit retaining some relevance to certain periods depending on the time frame length. This is sometimes called stratified sampling [34]. An illustration of one possible dynamic encounter graph G is given in Fig. 4. Within G there are a number of devices which do not change between time frames $t_1 - t_3$, and pair-wise observations within time frames are independent of observations from other time frames. For example, an edge representing an asymmetric encounter between devices d_i and d_j within time frame t_1 in Fig. 4 is represented as $e^{t_1}_{d_i d_j}$.

As in aggregated contact graphs, the interactions between devices represented in dynamic encounter graphs can be arbitrary measurements, such as signal strength and the number of messages transferred. Suppose that the dynamic encounter graph in Fig. 4 shows transient asymmetrical connections between devices where data transfer is possible. Then it is important to note another property of these MANETs which can be lost during aggregation: the shortest path between devices in a dynamic encounter graphs is often not the quickest [36]. Furthermore, until the relationship between edges in Reality Mining data is fully understood, every edge should be believed to be independent of each other, which means a connection between two devices does not guarantee a second connection between two other devices in the next time frame.

Even within shorter time frames, temporal information about interactions between devices could still be lost in dynamic encounter graphs. Occasions within each time frame where the frequency of encounters varies may not be retrospectively identified. It has been found in previous work on temporal contact graphs [38] that identifying meaningful resolution levels is critical to matching the rate of change in network structure. To guard against loss of temporal information, a number of other strategies for labelling edges can be adopted:

[1] CRAWDAD Repository http://crawdad.cs.dartmouth.edu/.

Table 1
Comparison of mobility traces and the average encounter probability between devices for the 1st, 2nd, 3rd, and 4th quarter of the day.

	Infocom5	Infocom6	Cambridge	Reality
Environment	Conference		City	Campus
Duration (day)	3	3	12	246
Number of devices	41	78	36	97
Inter-probe time (s)	120	120	600	300
Daily encounter probability	0.7807	0.7324	0.24	0.0022
Prob. 1st 1/4 day	0.3892	0.3549	0.0122	0.0003
Prob. 2nd 1/4 day	0.4049	0.0447	0.1754	0.0011
Prob. 3rd 1/4 day	0.0173	0.3116	0.0852	0.0019
Prob. 4th 1/4 day	0.4086	0.4683	0.0113	0.0012

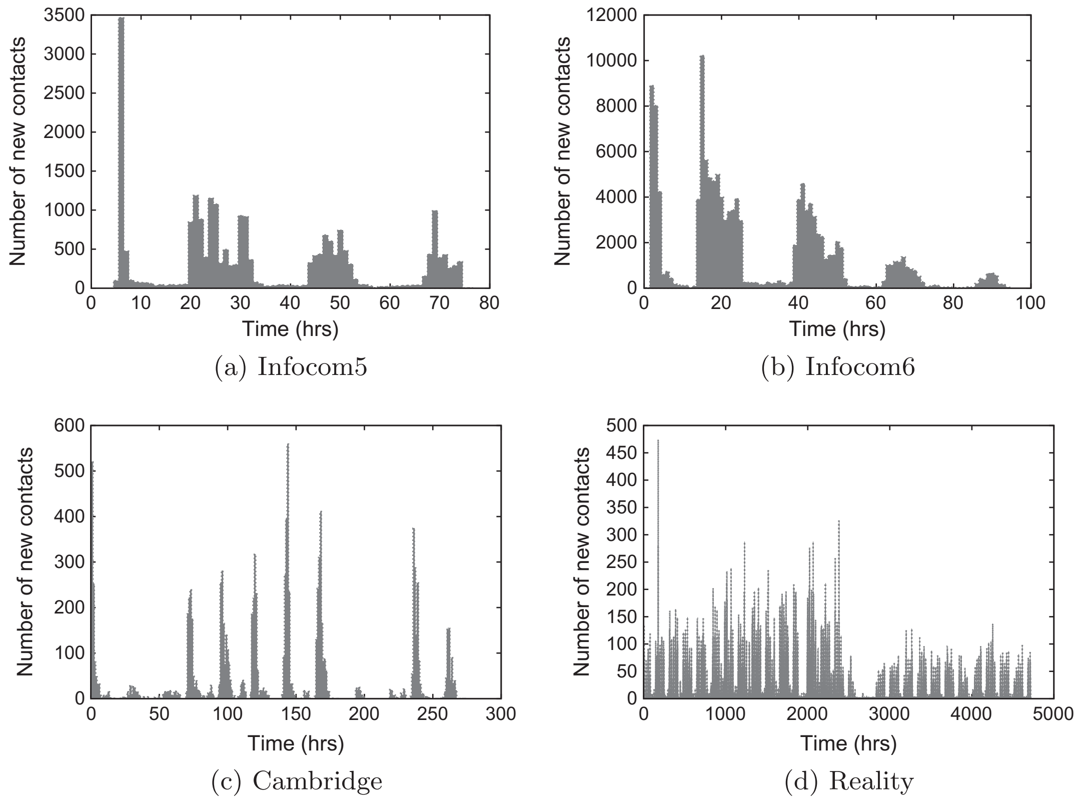


Fig. 3. New contacts in hourly time frames for the different Reality Mining data-sets.

1. Edges between vertex pair d_i and d_j which occur between times t_{start} and t_{end} can be represented in the form $e_{ij}^{t_{start}t_{end}}$. Furthermore $e_{ij}^{t_{start}t_{end}}$ could be weighted to represent connection strength between d_i and d_j during the interval t_{start} to t_{end}.
2. Edges between d_i and d_j could be weighted to show total encounter or connection time after t_{start}. E.g. $e_{ij}^{t_{start}}$ could be used to express the duration of an encounter which started at t_{start} between d_i and d_j.

However, neither of these approaches are adopted for this paper. The creation of dynamic encounter graphs allows us to easily identify collective behaviours of devices in temporal regions within Reality Mining data-sets, similar to the identification of the bursty behaviour seen previously in Fig. 3. Furthermore, the slight aggregation of encounter data allows us to mitigate for the Parking Lot Problem in our analysis.

3. Distributed spatio-temporal clustering

Distributed Rise and Fall spatio-Temporal (DRAFT) clustering is our proposed method to provide spatio-temporal, non-social clustering within dynamic encounter graphs. It combines spatial clustering with a decay function. This means that clusters reflect current and recent behaviour patterns by excluding devices which have not been seen for a long time.

The protocol needs three parameters to govern the rate at which clusters grow and decay, suggested values for which will be discussed in the following sections and de-

Please cite this article in press as: M. Orlinski, N. Filer, The rise and fall of spatio-temporal clusters in mobile ad hoc networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.03.003

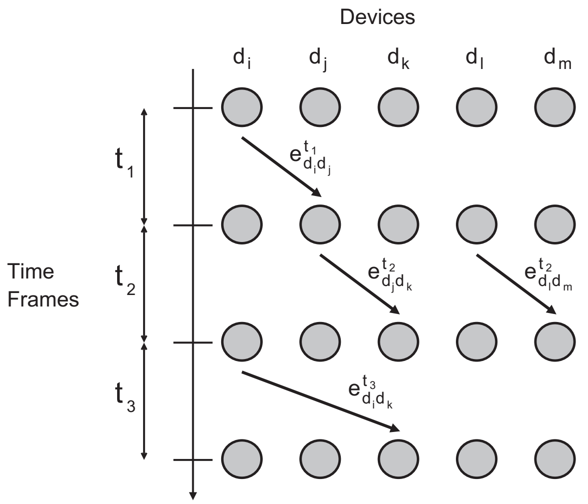


Fig. 4. A dynamic encounter graph between time frames t_1 and t_3. With asymmetric encounters between devices during time frames shown.

pend on the mobility, expected encounter duration, and what length of time spatio-temporal clusters describe:

1. The familiar threshold of length τ seconds is the threshold at which cumulative encounter durations between devices trigger the cluster inclusion process.
2. A time frame of length t seconds governs the interval at which the cumulative encounter durations for each device are decayed.
3. The decay ratio δ which should be in the range $0 \leqslant \delta \leqslant 1$ governs how much the cumulative encounter durations are reduced at the end of each time frame.

The reason why cumulative encounter durations are stored for neighbouring devices running DRAFT, rather than a single encounter or inter-encounter times [6], is to allow for the Parking Lot Problem. If encounters are frequently interrupted by lost neighbour discovery requests, then basing cluster membership on single encounter durations or the time between encounters will be unreliable. It is also important to note that after the first time frame, pair-wise encounter durations are no longer truly cumulative. They have been decayed by a certain amount, but future encounter durations between the pair will continue to be added to the new amount. Also, encounter duration decay is multiplicative rather than additive because some devices may have very different mean encounter durations to others. Thus encounter time decay being multiplicative allows for different levels of connectivity, and so decay can be specified easily for the entire network.

3.1. Building clusters

Spatio-temporal clusters in DRAFT are formed opportunistically by non-social [19], pair-wise encounters, which are then used as a network hierarchy with which to relay messages within PSNs. The process with which devices are added to local spatio-temporal clusters involves three data structures for efficient processing. A device d_i maintains the following information:

1. A set of tuples containing encountered devices and associated encounter durations, called the neighbour set N_i.
2. A local spatio-temporal cluster C_i.
3. A table D_i, containing devices marked for deletion from the local spatio-temporal cluster, and devices already deleted.

The process with which clusters are then built up can be summarised as:

Rule 1 Initially C_i is set to $\{d_i\}$, N_i and D_i are set to $\emptyset$.

Rule 2 When d_i encounters another device d_j, d_i enters d_j into N_i if it is not already there, and begins to add the duration of the encounter to the corresponding record in N_i.

Rule 3 If the encounter time stored in N_i for d_j, called N_{ij}, exceeds the familiar threshold τ; or d_i encounters a device d_j which is already a member of C_i; or it is the end of the current time frame on d_i and $N_{ij} > -\tau$; then d_i requests information from d_j i.e. d_i requests C_j and D_j from d_j. If the request is successful the algorithm then:

1. Adds d_j to the local spatio-temporal cluster, C_i of d_i.
2. If d_j has been marked for deletion by being present in D_i (see Section 3.2), then d_j is "forgiven" and removed from D_i.

This process is performed independently by all the other devices in the network, including d_j.

3.2. Cluster decay and device cooperation

To facilitate cluster decay, the passage of time is split into a number of discrete time frames of length t. At the end of each time frame, associated encounter times in neighbour sets are decreased by multiplying them by the decay ratio δ ($\delta = 1$ no decay, $\delta = 0$ absolute decay). Cluster membership is reassessed by each device:

Rule 4 Once the requested information from Rule 3 in Section 3.1 has been received and processed, the algorithm also tries to delete old records:

1. d_i checks records in D_i against those in D_j. As encounters in PSNs are opportunistic, a spatio-temporal commonality test has been passed and any devices which are in both D_i and D_j are deleted immediately from D_i and C_i without waiting until the end of the next time frame.
2. If a record in D_i is in C_j but not D_j then the device is not deleted.
3. If a record is in D_i but not in C_j or D_j then the record is left in D_i in case another device is encountered with a matching record in the future.

Rule 5 At the end of each time frame:

1. Any records still in both C_i and D_i are considered old and removed from C_i.
2. The records are kept in D_i for commonality tests with other devices, or until the device in the record is added to C_i once more.

Please cite this article in press as: M. Orlinski, N. Filer, The rise and fall of spatio-temporal clusters in mobile ad hoc networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.03.003

3. All connected times for devices in N_i are multiplied by the decay ratio δ in order to keep records fresh. Any records which fall below the familiar threshold τ are marked for deletion by being added to D_i ready for the end of the next time frame.

To save memory, devices are also removed from neighbour sets once their associated encounter durations decay to below a small number e.g. 0.1 s. Informally, the values of δ, t, and τ will be related to the mobility of the participants and how reactive cluster membership should be, which is determined by the application and/or user. As inter-human encounters are transient and diurnal [7], the length of τ for PSNs should be greater than the mean encounter duration for each device, but less than 24 h. The combination of these three variables makes the DRAFT algorithm tunable for a variety of applications. If clusters are needed which grow rapidly and decay quickly, τ should be close to the mean encounter duration, and δ should be closer to 0 than 1. Another way to phrase this is that if mobility is high, and clusters should reflect recent encounters, δ should be low but not zero, and τ close to the mean encounter duration. Conversely, if clusters are needed that reflect longer periods, δ can be made higher.

3.3. Data delivery

One of the aims of this work is to test spatio-temporal clustering for data delivery efficiency against aggregated monotonic clustering. For this reason, the semi-oblivious data forwarding mechanism used in Quality [4] is adopted, thus testing only the different cluster definitions. This data forwarding method floods clusters with duplicate packets. It is unforgiving in that many duplicate packets will be created if clusters do not accurately reflect current or future connectivity.

Algorithm 1. Data delivery in DRAFT.

```
for each connected device as e do
  if NotInLocalCluster(e) then
    RequestLocalClusterFrom(e)
  end if
  if HaveLocalClusterOf(e) then
    for each message as m do
      if e = DestinationOf(m) then
        DeliverMessage(e,m)
        DeleteMessage(m)
      els if LocalCluster of e contains
      DestinationOf(m) then
        CopyAndTransferMessage(e,m)
      end if
    end for
  end if
end for
```

The ability of devices to request local clusters from nearby devices allows them to check for possible 2-hop paths. Note, no explicit extra roles for devices are assigned during this process. 2-Hop routing is simply a consequence of the movement patterns of participants, and being able to either:

1. Ask a remote device if it has a message destination in it's local cluster.
2. Or inspect a copy of the remote device's local cluster to see if it contains a message destination.

Both approaches would work with DRAFT, but as devices may have many messages ready to transmit and the DRAFT algorithm can request a remote device's local cluster, the later approach is used to cut down on the number of requests. The actual checks performed before message duplication are detailed in Algorithm 1. They do not include checks for devices further than 2 hops away as this would require devices to exchange and store a large amount of additional data.

Fig. 5 illustrates the resulting 2-hop delivery possibilities using example local spatio-temporal clusters of d_i and d_j. Upon meeting d_j, the device d_i can inspect d_j's cluster information C_j to see if the destination of a message lies within C_j. If it does, then the message can be copied to d_j. There is no guarantee that a message will be delivered immediately, or even by device d_j due to device mobility. However, the message is now with both d_i and d_j which may increase the chance of the message reaching the destination without obviously flooding the network.

3.4. Simulation environment

The experiments contained within this paper were conducted using The One [29] network simulator and encounters provided by the Reality Mining data-sets from Section 2. Using this method, simulations attempt to create the conditions found in future PSNs. As encounters found in the Reality Mining data-sets are records of real events, the resulting simulations are free from errors introduced when using synthetic movement models such as The Working Day Movement Model [35]. However, one

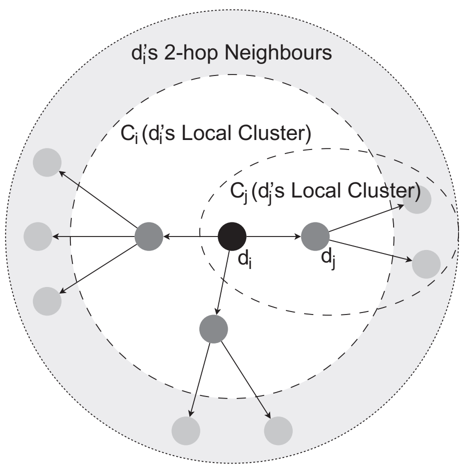


Fig. 5. A device d_i, can see potential 2-hop neighbours upon meeting another device d_j.

drawback to this approach is that the networks are fragmented as increasing participation in Reality Mining experiments can be expensive.

Even though our simulations assume symmetric encounters and communication channels with a 2 Megabits per second data rate up and down, the DRAFT algorithm does not break down if requests for information fail. In fact, failures to reply to requests is handled in the DRAFT protocol and this helps to prune the epidemic distribution tree of unreliable links.

In the experiments in Section 5, all messages are 160 bytes in size to model typical mobile phone text message size. New messages are generated to be sent between random pairs of devices at 30 s intervals. For controlled experiments, the same five random message generation patterns were used for DRAFT [2] and other protocols against which we compare data delivery. The message Time To Live (TTL) was set at 1 h for all the data-sets other than Reality, where message TTL was set at 1 day due to the sparsity of the encounters. Once the TTL for a message has expired then the message is deleted.

It is also important to note that for the Reality data-set, we have truncated the data and only used data available between the time-stamp ranges 1,094,545,041 and 1,111,526,856. This is because there is no significant activity before and after these times respectively. Furthermore, the parameters suggested by the PRoPHETv2 [6], Bubble [2] and Quality [4] research papers were used for their respective protocols.

4. Spatio-temporal cluster analysis

This section will offer analysis of the spatio-temporal clusters and possible 2-hop neighbours formed when using the DRAFT algorithm with the chosen Reality Mining data-sets. We will also attempt to describe some of the differences between building spatio-temporal and aggregated monotonic clusters.

4.1. Updates to clusters

In both aggregate monotonic and spatio-temporal clustering, clusters undergo a number of changes as they are created. Table 2 shows the number of changes to spatio-temporal clusters in DRAFT with $\delta = 0.8$, $\tau = 120$ s and $t = 3600$ s as an example, compared to the aggregated monotonic cluster detection method called Simple [15] which was introduced in Section 1. Remember that cluster decay is used by DRAFT to remove devices from clusters. With DRAFT continuously assessing cluster membership, it creates many more cluster formation related operations in each data-set.

4.2. Resulting cluster size

As cluster membership in DRAFT ($\delta = 0.8$, $\tau = 120$ s, and $t = 3600$ s) is continuously reassessed, cluster size varies over time as cluster members are added and removed. Table 3 shows that normal local spatio-temporal clusters also end up being smaller than aggregated monotonic clusters from the Simple [15] algorithm used by Bubble.

Table 2
Average number of instructions per device issued by DRAFT to increase (Inc.) and decay (Dec.) cluster size.

Data-set	Simple		DRAFT	
	Inc.	Dec.	Inc.	Dec.
Infocom5	17.6	n/a	46.1	17.5
Infocom6	29.8	n/a	136.4	116.3
Cambridge	10.6	n/a	29.5	12.5
Reality	16.3	n/a	34.4	25.3

Table 3
Mean local cluster size.

Data-set	Simple	DRAFT
Infocom5	81.46	75.20
Infocom6	79.59	49.99
Cambridge	86.81	54.37
Reality	55.84	13.62

When using cluster based forwarding, smaller clusters may lead to fewer packets being delivered to their final destinations [4]. However, the next subsection will show that when considering the possible number of 2-hop neighbours, smaller cluster size may not prove to be a barrier to data dissemination.

4.3. Cluster size and composition over time

Fig. 6 shows daily snapshots of the mean number of devices contained in local spatio-temporal clusters and the mean number of possible 2-hop neighbours for each data-set (DRAFT with $\delta = 0.8$, $\tau = 120$ s, and $t = 3600$ s). In the Infocom and Cambridge data-sets, the number of 2-hop neighbours is on average 87% larger than the number of devices in local spatio-temporal clusters. In the Reality data-set the proportion of 2-hop neighbours to local spatio-temporal cluster size is much larger, with on average three times more 2-hop neighbours than devices in spatio-temporal clusters.

It is also interesting to note that the number of 2-hop neighbours increases over time, despite local spatio-temporal cluster size possibly decreasing over the same period. Furthermore, the set of 2-hop neighbours usually contains most of the devices in each experiment (>60% of devices in Reality, and >90% in other experiments), but this may be due to the experiment being enclosed in a relatively confined geographic space.

4.4. Cluster size and 2-hop neighbours

Heat-maps for normalised local spatio-temporal cluster sizes and 2-hop neighbours for hourly snapshots, using DRAFT settings $\delta = 0.8$, $\tau = 120$ s, and $t = 3600$ s, are shown in Fig. 7. They show a marked difference between the Reality trace and other data-sets in terms of both

[2] An implementation of the DRAFT protocol for The One Simulator can be obtained from http://bit.ly/Rvvo86.

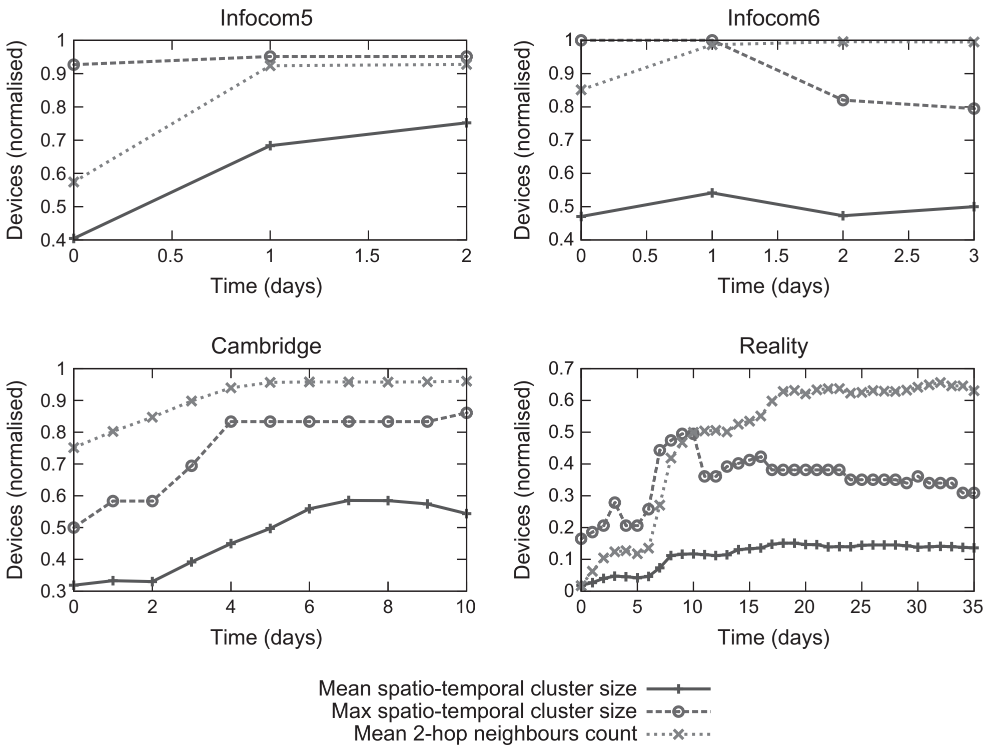


Fig. 6. Mean and max local spatio-temporal cluster sizes and 2-hop neighbours over time.

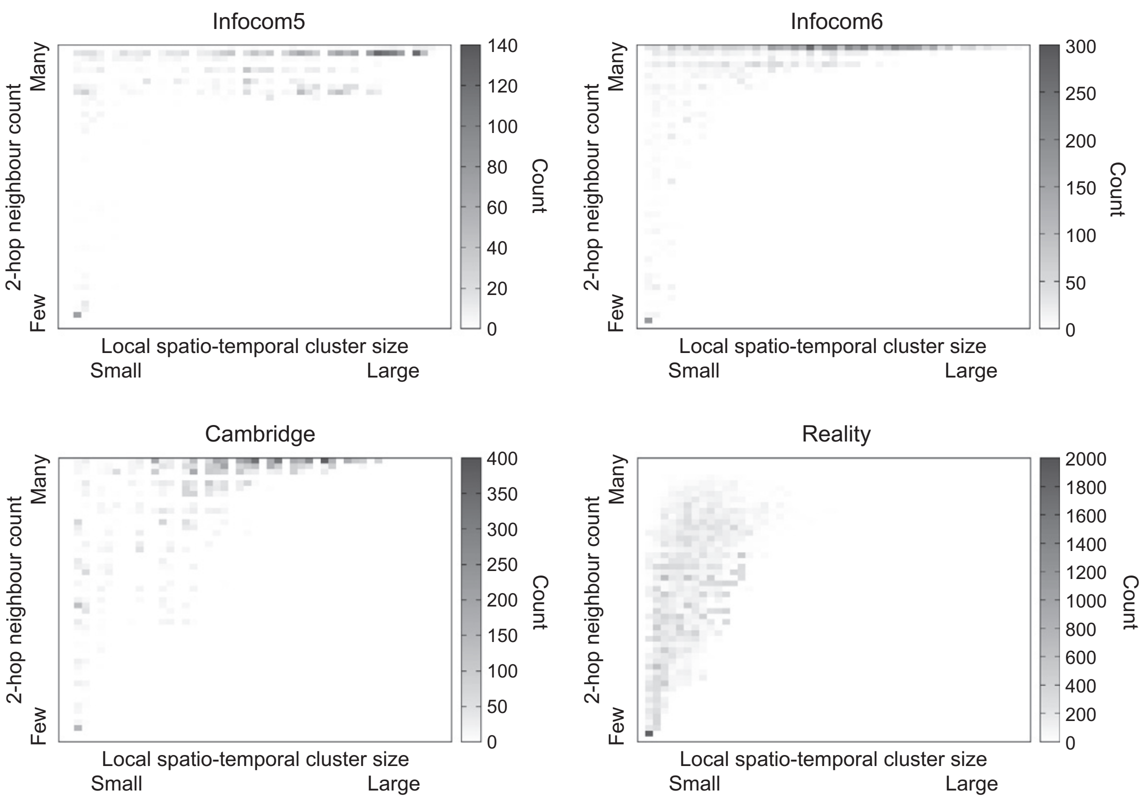
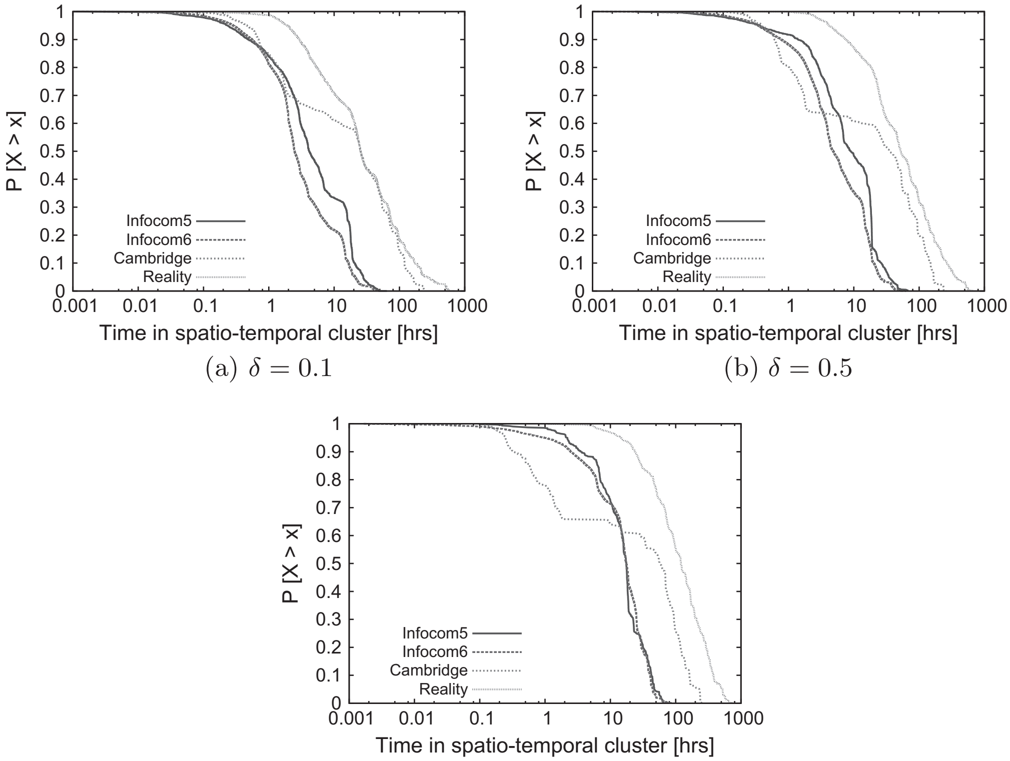


Fig. 7. Number of potential 2-hop neighbours depending on local spatio-temporal cluster size.



(a) $\delta = 0.1$ (b) $\delta = 0.5$ (c) $\delta = 0.8$

Fig. 8. Probability distribution of cluster membership times.

spatio-temporal cluster sizes and 2-hop neighbours. In the Reality trace, there are often many 2-hop neighbours but cluster size remains low with a mean hourly cluster size of just 10 devices compared with 16 in Cambridge, 25 in Infocom5, and 38 devices in Infocom6. This limited cluster size in Reality can have a negative effect on spatio-temporal cluster based data delivery as shown later in Section 5.1.

4.5. Time spent in spatio-temporal clusters

The time spent in spatio-temporal clusters is dependent on how the spatio-temporal clusters are defined. DRAFT is no different and spatio-temporal cluster membership times depend on factors such as decay rate, familiar threshold and time frame size. Fig. 8 shows the probability that spatio-temporal cluster membership time will exceed a given value x in each data-set for δ values of 0.1, 0.5, and 0.8 when using a time frame length of 1 h and a familiar threshold of 120 s.

Interestingly, the data shows that with a fast decay rate $\delta = 0.1$, over 81% of all cluster memberships still last longer than 1 h. In Reality though, 98% of spatio-temporal clusters memberships last longer than an hour, suggesting that the number of vagabonds (devices which move between clusters) in Reality is low.

There is also a marked difference in cluster duration between the conference and campus experimental environments. In the Infocom5 and Infocom6 data-sets, between 40% and 57% of spatio-temporal clusters last longer than 3 h. However in the campus datasets the time spent in spatio-temporal clusters is generally less. In Cambridge less than 30% of devices spent longer than 3 hours in DRAFT clusters, and in Reality only 10%. This indicates that the campus wide experiments have a more diverse selection of participants who interact for shorter periods than participants at a conference.

5. Data delivery results

In this section, the DRAFT protocol will be compared against two opportunistic message delivery protocols, Bubble [2] and Quality [4]. We also provide figures for the PRoPHETv2 [6] forwarding protocol which does not use clustering but provides state of the art delivery success ratios and efficiencies. For Bubble, both the K-clique and Simple [15] clustering techniques were used to provide the aggregated monotonic clusters needed for the experiments.

For δ values less than 0.5 (but τ still 120 s and $t = 3600$ s), spatio-temporal clusters in DRAFT decay too rapidly to give reliable data-dissemination. It should also be stressed that "cherry picking" parameters for protocols in this way is acceptable and consistent with other works. The Bubble and PRoPHETv2 algorithms were afforded the same consideration, and only the value ranges from their respected papers and One Simulator implementations were used to give the best possible results for each protocol.

Please cite this article in press as: M. Orlinski, N. Filer, The rise and fall of spatio-temporal clusters in mobile ad hoc networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.03.003

Table 4
Mean data delivery results across all experiments. Time frame length for DRAFT was always $t = 3600$ s.

Method	Data delivery ratio	Overheads
Bubble	0.1141	20.6972
PRoPHETv2	0.1445	25.6832
DRAFT ($\delta > 0.5$, $\tau = 120$ s)	0.1472	25.3480
Quality	0.1717	56.3885
DRAFT ($\delta = 0.99$, $\tau = 5$ s)	0.1620	37.1880

Overall DRAFT with δ values in the range $0.5 < \delta \leqslant 0.99$, $\tau = 120$ s, and $t = 3600$ s delivers as many packets as PRoPHETv2, but with slightly lower overheads in the form of duplicate packets (see Table 4). The data delivery results for the algorithms presented here may seem low, but this is not a result of a limitation in the technology. The low data delivery success rates are a consequence of the random message generation, short TTL of messages (see Section 3.4), and the fragmented nature of the Reality Mining data-sets due to low participation and long intervals between neighbour discovery probes [30].

The variation of the data delivery results for each protocol can be found in Fig. 9. Generally, DRAFT can be counted upon to deliver more packets than Bubble or PRoPHETv2. However, Fig. 9 also shows that DRAFT's data delivery rate is comparatively lower than that obtained using PRoPHETv2 in the Reality case. A reason for this low delivery rate is likely to be the small size of spatio-temporal clusters as seen previously in Fig. 6 and Section 4.4 which will be explored more fully in the next section.

5.1. Data delivery in the Reality data-set

One mechanism to increase average spatio-temporal cluster size in DRAFT is to lower the rate of cluster decay. For example, at $\delta = 0.99$ encounter durations are only decreased by 1% at the end of each window. Also, by lowering the familiar threshold τ, more devices are included in local spatio-temporal clusters in the first instance. If t still equals 3600 s but τ is limited to 5 s and δ set to 0.99, the resulting clusters are three times larger in Reality than when $\delta = 0.8$, $\tau = 120$ s, and $t = 3600$ s (see Fig. 10a), but data delivery in the Reality data-set using these settings is still 6% lower than that given by PRoPHETv2. The trade off is still efficiency, with PRoPHETv2 needing to relay twice as many packets as DRAFT to achieve that 6% increase in data delivery, as illustrated by the duplicate packet overheads over time in Fig. 10b.

Most of the efficiency gains of DRAFT over PRoPHETv2 in Fig. 10b come at the start of the experiment. Further inspection of Fig. 10a shows that spatio-temporal clusters during the early stages of the experiment are still very small compared with later on. Moreover, Fig. 10c shows that DRAFT is delivering slightly fewer packets than PRoPHETv2 during this period, for reasons which appear to

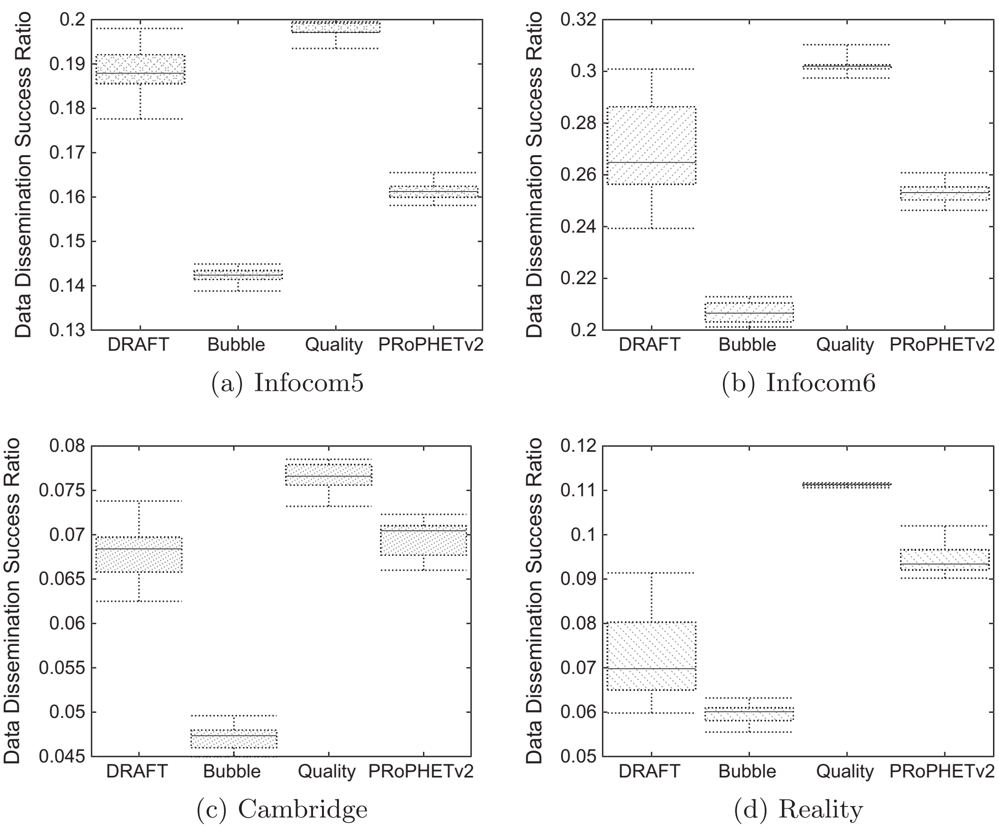
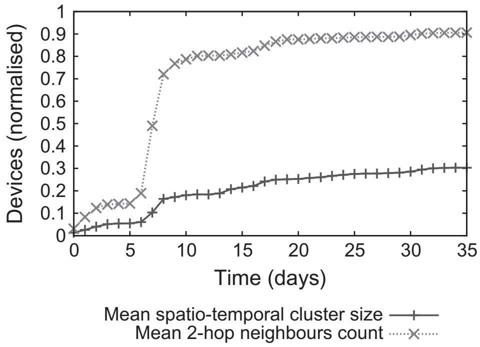
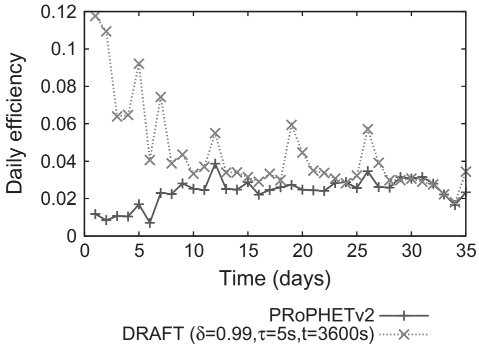


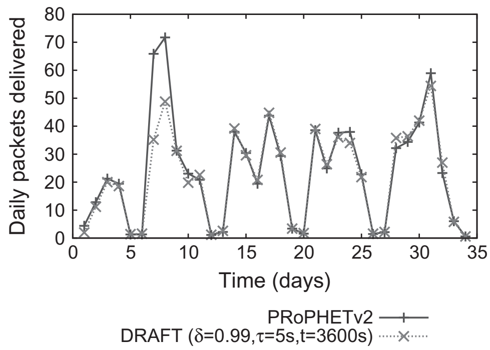
Fig. 9. Minimum, first quartile, median, third quartile and maximum data delivery performance for each data-set.



(a) Cluster composition.



(b) Efficiency.



(c) Data delivery.

Fig. 10. There are long periods of time between encounters in the Reality data-sets. As such decay rate and familiar thresholds have been altered for these results to be $\delta = 0.99$ and $\tau = 5$ s respectively (t is still 3600 s).

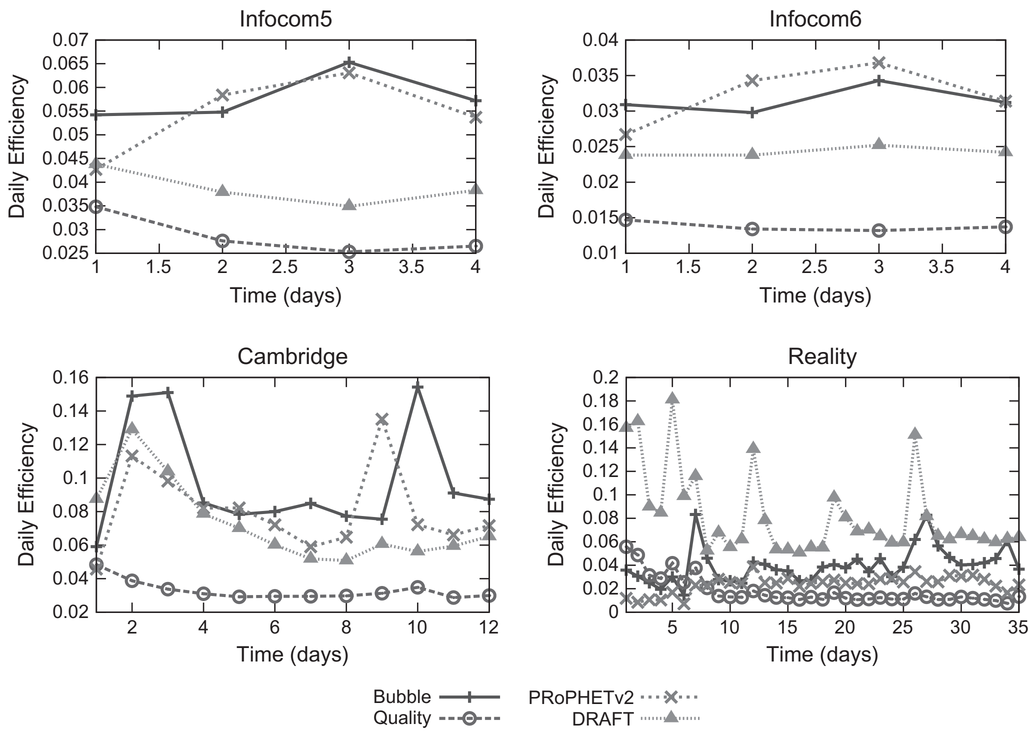


Fig. 11. Duplicate message overheads over time. DRAFT settings, $\delta > 0.5$, $\tau = 120$ s, and $t = 3600$ s.

Please cite this article in press as: M. Orlinski, N. Filer, The rise and fall of spatio-temporal clusters in mobile ad hoc networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.03.003

be related to cluster size. As the early stages of the Reality experiment have low delivery success rates, and local spatio-temporal cluster sizes are lower than 25% of the total data set size, the findings are consistent with those in [4] which stated that large cluster sizes are needed to disseminate the maximum amount of data using this 2-hop delivery method.

5.2. Efficiency over time

One of the predictions for spatio-temporal clustering is that it could improve the data delivery efficiency from that produced by aggregate clustering by lowering message duplication overheads. This hypothesis in now explored in more detail.

The data delivery mechanism in DRAFT is the same as in Quality. Even so, Fig. 11 shows DRAFT is a marked improvement on the data-delivery efficiency of Quality across each of the data-sets explored. Therefore it is fair to say that spatio-temporal clustering is more efficient as it performs better than Quality in terms of creating fewer duplicate data packets. However, Fig. 9 showed that DRAFT does not deliver as many packets to their final destinations as Quality in the Infocom6 and Reality data-sets. To attempt to correct this, τ was limited to 5 s and δ set to 0.99, and all experiments were repeated. Table 4 shows the results of doing this were that the data delivery rates of DRAFT are only 6% lower than Quality with 34% fewer packets relayed.

6. Conclusions

We set out to prove that spatio-temporal clusters can achieve high delivery rates and efficiency in a specific type of MANET called a PSN. We have shown that this is the case when using the new DRAFT protocol, but also that the choice of data-delivery mechanism to take advantage of 2-hop neighbours, and identifying suitable lower bounds for spatio-temporal cluster sizes will be crucially important in later editions of the protocol. By forwarding data using non-social spatio-temporal clusters, we have shown that it is possible to deliver 95% of all the messages it is possible to deliver with Quality but more efficiently (see Table 4), but how to deliver the final 5% of messages in a timely and efficient fashion is still an open problem.

Whilst we could argue that the results in Sections 5.1 and 4.3 suggest that spatio-temporal clusters should contain between 25% and 50% of the devices in the data-set (10–38 devices), this may not be scalable for larger experiments, and would depend on whether the goal is maximum or efficient data delivery. Similarly, we have not made recommendations of best values to choose for δ, t, and τ for other Reality Mining data-sets because we simply do not have enough data to support such a claim. Therefore, larger data-sets are needed before optimum spatio-temporal cluster sizes, and suitable mechanics to control clustering algorithm's parameters can be determined.

As an interesting side note, because data delivery rates and efficiencies are on a par overall with PRoPHETv2, we tentatively conclude that spatio-temporal clusters formed from well connected pairs in DRAFT are as good an indication of message delivery success as the predictions based on previous encounters used in PRoPHETv2. At least when considering human behavioural patterns. Therefore, future work into this area may want to investigate if pair-wise encounters in PSNs have the Markov property as future encounters may depend upon the present state in some predictable way. If this were true it may lead to cheaper, faster PSN routers as there would be no need to store data on encounters for longer than 24 h. The diurnal patterns discovered by Henderson et al. [7] also hint that this may be the case if predictions were taken based on day long time frames.

An assumption we made at the start of this work is that the data delivery success rates of cluster based routing can never reach those achieved by oblivious forwarding of packets along all possible paths (which is in effect what Quality does by creating huge aggregated monotonic clusters). Spatio-temporal clusters in DRAFT are typically much smaller than Quality's, and as a result the set of possible paths which messages can follow are limited, but no more so than with PRoPHETv2 or Bubble. In each protocol analysed there is a trade off between data delivery success rates and efficiency. Therefore, without better methods of predicting the whereabouts of devices, or knowing the likelihood of there being some interaction with other devices in the near future, it may not be possible to push the efficiency of these protocols further. Some small gains may be made by storing local clusters of encountered devices to go beyond 2-hops as in DEBT [18], or use a hybrid approach like Bubble [2], but neither of these methods have been shown here to deliver as much data as efficiently as DRAFT.

References

[1] R. Sokal, The Principles and practice of numerical taxonomy, Taxon 12 (5) (1963) 190–199. ISSN 00400262.
[2] P. Hui, J. Crowcroft, E. Yoneki, BUBBLE rap: social-based forwarding in delay tolerant networks, Mobile Computing 6 (1) (2007) 1576–1589.
[3] E. Daly, M. Haahr, Social network analysis for routing in disconnected delay-tolerant manets, ACM International Symposium on Mobile Ad hoc Networking and Computing, ACM, Quebec, Canada, 2007. pp. 32–40.
[4] M. Orlinski, N. Filer, Quality distributed community formation for data delivery in pocket switched networks, in: Annual Workshop on Simplifying Complex Networks for Practitioners, ACM, Lyon, France, 2012a, pp. 31–36.
[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding, Tech. Rep. University of Cambridge, Computer Lab. UCAM-CL-TR-617, 2005.
[6] S. Grasic, E. Davies, A. Lindgren, A. Doria, The evolution of a DTN routing protocol-PRoPHETv2, in: ACM Workshop on Challenged Networks, 2011, pp. 27–30.
[7] T. Henderson, D. Kotz, I. Abyzov, The changing usage of a mature campus-wide wireless network, Computer Networks 52 (14) (2008) 2690–2712.
[8] N. Eagle, A. Sandy Pentland, Reality mining: Sensing complex social systems, Personal and Ubiquitous Computing 10 (4) (2006) 255–268.
[9] P. Hui, People are the network: experimental design and evaluation of social-based forwarding algorithms, Tech. Rep. University of Cambridge, Computer Lab. UCAM-CL-TR-713, 2008.
[10] R. Krishnan, D. Starobinski, Efficient clustering algorithms for self-organizing wireless sensor networks, Ad Hoc Networks 4 (1) (2006) 36–59.

[11] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, Maxprop: Routing for vehicle-based disruption-tolerant networks, in: IEEE International Conference on Computer Communications, 2006, pp. 1–11.
[12] C. Perkins, E. Royer, Ad hoc on-demand distance vector routing, in: IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.
[13] P. Jacquet, P. M++hlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, in: IEEE International Multi Topic Conference, Technology for the 21st Century, 2001, pp. 62–68.
[14] P. Hui, J. Crowcroft, How small labels create big improvements, in: IEEE International Conference on Pervasive Computing and Communications Workshop, 2007, pp. 65–70.
[15] P. Hui, E. Yoneki, S.Y. Chan, J. Crowcroft, Distributed community detection in delay tolerant networks, in: Proceedings of 2nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture, Kyoto, Japan, 2007b.
[16] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, A. Chaintreau, CRAWDAD trace – Haggle. <http://crawdad.cs.dartmouth.edu/cambridge/haggle>, 2009.
[17] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, The Eleventh International Conference on Knowledge Discovery in Data Mining, ACM, 2005. pp. 177–187.
[18] M. Orlinski, N. Filer, Distributed expectation-based spatio-temporal cluster detection for pocket switched networks, in: IFIP International Wireless Days Conference, Dublin, Ireland, 2012b.
[19] A. Pietilainen, C. Diot, Dissemination in opportunistic social networks: the role of temporal communities, in: ACM International Symposium on Mobile Ad hoc Networking and Computing, South Carolina, USA, 2012, pp. 165–174.
[20] S. Gaito, C. Quadri, G. Rossi, M. Zignani, THINPLE – the new online Sociality is built on top of NFC-based contacts, in: IFIP International Wireless Days Conference, Dublin, Ireland, 2012.
[21] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, S. Bhatti, Exploiting self-reported social networks for routing in ubiquitous computing environments, in: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2008, pp. 484–489.
[22] G. Zyba, G. Voelker, S. Ioannidis, C. Diot, Dissemination in opportunistic mobile ad hoc networks: the power of the crowd, in: IEEE International Conference on Computer Communications, 2011, pp. 1179–1187.
[23] K. Wehmuth, A. Ziviani, Distributed assessment of the closeness centrality ranking in complex networks, IEEE International Conference on Wireless and Mobile Computing, ACM, Lyon, France, 2012. pp. 4348.
[24] X. Zhang, G. Neglia, J. Kurose, D. Towsley, Performance modeling of epidemic routing, Computer Networks 51 (10) (2006) 827–839.
[25] E. Borgia, M. Conti, A. Passarella, Autonomic detection of dynamic social communities in opportunistic networks, in: IFIP MedHocNet, Favignana, Italy, 2011, pp. 142–149.
[26] I. X. Leung, P. Hui, P. Lio, J. Crowcroft, Towards real-time community detection in large networks, Physical Review E 79 (6) (2009).
[27] G. Herbiet, P. Bouvry, SHARC: Community-based partitioning for mobile ad hoc networks using neighborhood similarity, in: IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, 2010, pp. 1–9.
[28] N. Eagle, A. Pentland, CRAWDAD trace – Reality. <http://crawdad.cs.dartmouth.edu/mit/reality>, 2005.
[29] A. Keranen, J. Ott, T. Karkkainen, The ONE simulator for DTN protocol evaluation, in: International Conference on Simulation Tools and Techniques, ICST, Rome, Italy, 2009.
[30] M. Orlinski, N. Filer, Movement speed based inter-probe times for neighbour discovery in mobile ad hoc networks, Ad Hoc Networks, vol. 111, Springer, Paris, France, 2012.
[31] A. Lindgren, P. Hui, The quest for a killer app for opportunistic and delay tolerant networks, in: ACM Workshop on Challenged Networks, 2009, pp. 59–66.
[32] C. Petrioli, S. Basagni, I. Chlamtac, BlueMesh: degree-constrained multi-hop scatternet formation for Bluetooth networks, Mobile Networks and Applications 9 (1) (2004) 33–47.
[33] J.W. Lin, W.S. Wang, An efficient reconstruction approach for improving Bluetree scatternet formation in personal area networks, Network and Computer Applications 33 (2) (2010) 141–155.
[34] I.X. Leung, S.Y. Chan, P. Hui, P. Lio, Intra-city urban network and traffic flow analysis from GPS mobility traces, arXiv:1105.5839.
[35] F. Ekman, A. Keranen, J. Karvo, J. Ott, Working day movement model, in: ACM SIGMOBILE Workshop on Mobility Models, Hong Kong, China, 2008, pp. 33–40.
[36] A. Panisson, A. Barrat, C. Cattuto, W. Broeck, G. Ruffo, R. Schifanella, On the dynamics of human proximity for data diffusion in ad hoc networks, Ad Hoc Networks 10 (8) (2011) 1532–1543. special issue on social-based routing in mobile and delay-tolerant networks.
[37] M.J. Williams, R.M. Whitaker, S.M. Allen, Decentralised detection of periodic encounter communities in opportunistic networks, Ad Hoc Networks 10 (8) (2012) 1544–1556. special issue on social-based routing in mobile and delay-tolerant networks.
[38] R. Sulo, T. Berger-Wolf, R. Grossman, Meaningful selection of temporal resolution for dynamic networks, in: Workshop on Mining and Learning with Graphs, Washington, D.C., 2010, pp. 127–136.

Matthew Orlinski is a Ph.D. student in School of Computer Science at the University of Manchester. He holds a B.Eng. (2007) in Internet Engineering from Aberystwyth University and an M.Sc. (2008) in Computer Security from the University of Birmingham. His research interests include pattern recognition in human movement and encounters, and data dissemination techniques for mobile ad hoc networks.

Nick Filer graduated with a B.Sc. in Computer Science from Queen Mary College London University in 1980. He obtained a Ph.D. from the University of Manchester in 1988 for work applying Artificial Intelligence to CAD systems for electronics design. He has been a Faculty member in Computer Science at Manchester University since 1984 and currently teaches computer networks at under/post graduate levels. He has research and 60+ publications in ECAD, AI, Software Engineering and more recently wireless networking. His current interests are sensor and ad hoc mesh networks, intelligent network stacks, storage and information dissemination in wireless social networks.

Please cite this article in press as: M. Orlinski, N. Filer, The rise and fall of spatio-temporal clusters in mobile ad hoc networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.03.003