Community-Based Forwarding for Low-Capacity Pocket Switched Networks

Khadija Rasul, Shaiful Alam Chowdhury, Dwight Makaroff and Kevin Stanley
Department of Computer Science, University of Saskatchewan
Saskatoon, SK, CANADA, S7N 5C9
{khr667, sbc882}@mail.usask.ca, {makaroff, kstanley} @cs.usask.ca

ABSTRACT

Sensor devices and the emergent networks that they enable are capable of transmitting information between data sources. Since these devices have low-power and intermittent connectivity, latency of delivery for certain classes of data may be tolerated in an effort to save energy. The BUBBLE routing algorithm, proposed by Hui *et al.*, provides consistent routing, employing a model which considers the popularity of individual nodes within communities and only passes messages to nodes with higher probability of delivery. We have developed an improvement to BUBBLE, called Community-Based-Forwarding (CBF) that considers the interactions between communities as an additional factor in message forwarding. By using community information, CBF is able to exploit intermediate connections between clusters to route messages with more balanced node participation and higher levels of reliability and efficiency.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing protocols; C.2.1 [**Network Architecture and Design**]: Network Topology

Keywords

delay-tolerant networks; pocket-switched networks; social-based routing; resource-constrained devices; clustering

1. INTRODUCTION

The number and variety of mobile computing/sensing devices is steadily increasing from smartphones, to personal medical monitors to smart badges. These devices vary in capabilities, but all desire to minimize energy consumption. We consider the least powerful class of devices with a single radio only capable of short-range transmissions on an opportunistic basis when other devices are detected.

Device sensing methodology and parameters as well as associated software may need to adapt in ways impossible to capture in an *a priori* manner. As well, delay-tolerant message generation paradigms can be anticipated. In particular, the recording of sensor values from many sensors to a single node for long-term trend analysis can be tolerant of delays on the order of days where message flooding is impractical due to energy concerns [25]. Individual nodes may communicate (one-to-one) when an implicit overlay of social connectivity exists that is unrelated to physical connectivity [16].

Pocket switched networks (PSNs) are a special case of delay tolerant networks (DTNs) where packets are routed from person to person in an ad-hoc manner, based on historical data regarding dynamic, non-uniform contact patterns [1, 13, 15]. Significant differences in contact patterns have been observed in environments for which datasets are available [3, 10, 11]. These dynamics suggest that routing based on inter-community contacts may aid routing performance.

Ideally, multi-hop routing tables would keep track of each PSN node's dynamic path length to all other nodes. This overhead rapidly outstrips the storage or power capacity of each node [20]; instead, researchers have leveraged the community structures that naturally arise from human interaction patterns [15], eliminating routing tables. Naive/greedy utilization of the network structure can favour nodes with high centrality measures. Where resources are limited, these "popular" nodes can experience packet loss due to buffer overflow and cause network failure through power depletion.

We develop and evaluate a Community-based Forwarding algorithm (CBF) that explicitly uses community structure as well as individual node-based connections of previous social-based approaches. Leveraging community linkages can exploit lower-centrality "bridging nodes" [6] to reduce buffer and power strain on the popular nodes. We compare CBF to a well known social-based algorithm (BUBBLE) [15], and both to a set of oracles to compare with optimal measures.

The contributions of this paper are threefold: 1) an approach identifying structures within the dynamic contact networks that can be leveraged for routing, 2) a new routing algorithm that provides superior performance in resource constrained environments, and 3) quantitative comparison of the impact of limiting key node resources on routing performance/efficiency of the algorithms considered.

2. RELATED WORK

The potential benefits and tradeoffs of DTN routing policies were first examined using a combination of simple routing strategies and oracle algorithms [16]. The simplest is Epidemic routing (ER) [30] where all nodes attempt to deliver all packets. While ER with unlimited resources pro-

MSWiM'14, September 21–26, 2014, Montreal, QC, Canada.

http://dx.doi.org/10.1145/2641798.2641801.

vides optimal reliability and delay, it is wasteful of bandwidth and buffers, forwarding packets to nodes with a low probability of being on the shortest path. Other researchers have tried to constrain the growth of multiple copies through adaptive limitations on packet time-to-live (TTL) [8] or by only passing additional packets through privileged nodes [2, 9]. While these approaches tend to improve both delivery ratio and latency, it is unclear whether these tradeoffs are worthwhile, given the delay tolerant properties of the data.

Many resource-conserving strategies consider the single copy case [28] which exhibits high latency and potentially poor delivery ratio, particularly when packet TTL indicates delay tolerance, but not delay immunity. Lindgren *et al.* proposed PRoPHET [20], which models nodes' future contacts directly from contact history. Context-Aware Adaptive Routing (CAR) [22] forwards packets to nodes with the highest probability of seeing the destination. Plankton [7] predicts the probability of future contacts and duration by classifying links based on the quantity/burstiness of previous contacts combined with a replica quota system controlling the number of copies allowed in the network. Node-level bookkeeping and communication overhead required can be prohibitively expensive for large networks or multi-hop routing.

Some researchers have attempted to model the overall behaviour of the dynamic graph by segmenting the graph into cliques, which we call communities in the rest of this paper. SimBet [6] employs betweenness and similarity metrics to route packets. Hui *et al.* [15] developed BUBBLE, which uses time-variant rankings based on recent history to route packets. A node is a member of at least one community and nodes are locally ranked, based on the number of contacts with other members of that community. Likewise, a global ranking is assigned to a node based on its global contacts.

Higher contact-rate, lower contact-duration nodes have been shown to play a major role in efficiently forwarding data [24]. In Lobby-Influence [18], the influence of the community structure derived from the dynamic network dominated the routing protocol employed.

The work presented in this paper extends work in PSNs by identifying the importance of intermediate bridging links for clusters of nodes, and by proposing a new heuristic-based routing algorithm which can capitalize on these links.

3. BACKGROUND

The stochastic processes that underlie human contact patterns have a non-uniform structure when aggregated over time. The contact probability network formed by summing time in contact between pairs of nodes in the graph tends to have small world properties [12]. This structure has been used in attempts to increase routing efficiency in PSNs [9, 15, 27] admitting that aggregate representation fails to capture instantaneous contact pattern dynamics. Periodicity of contact patterns has also been shown to influence the performance of DTN routing algorithms [23]. In particular, small world networks with highly connected clusters, containing short paths connecting every pair of nodes provide a promising means of improving routing [15]. Determining routing heuristics can then be reduced to solving two separate performance problems: 1) choosing the graph cluster structure, and 2) choosing nodes for inter-cluster communications.

Our approach is based on two observations of human contact network properties: 1) strong paths within communities due to small-world properties should provide fast intra-community routes, and 2) bridging nodes can be exploited to transfer packets closer to destination communities.

To use bridging nodes, some understanding of the cluster-level connectivity is required. For example, consider routing a packet from community A to community C. A does not contain any nodes that have strong connections to C, therefore its community-connection to C is low. However, if community B has nodes with strong connections to both A and C, then it could serve as an intermediate community.

Most PSN performance research uses simulation to compare algorithms fairly using the same contact patterns, which are typically generated in three ways: 1) directly from contact pattern traces [11, 12] datasets; 2) inferred from higher-level mobility data such as class [29] or bus [2] schedules; or 3) from synthetic contacts generated directly from theoretically grounded mobility patterns [19]. Our datasets are representative of university environments, and are among the longest datasets available. Synthetic and mobility datasets are likely to have different characteristics.

4. ALGORITHMS

4.1 Assumptions

We designed our algorithm subject to the following limiting assumptions: 1) PSN devices are **Resource Limited** in memory and computational power; energy usage is critical [16]; 2) packets possess a **Delay Tolerant** property encoded in TTL such that packet delivery is successful if delivered prior to TTL expiry [14]; 3) the contact durations have no effect on the ability to exchange buffer content metadata and messages themselves [5], simplifying the analysis; 4) the system represents a PSN that has non-degenerate **Human Mobility** patterns [20]; and 5) nodes are sufficiently **Socially Connected** to form quasi-stable cliques [27] with medium-term dynamics in which contact patterns change over time, but can be considered stable for an empirically determined particular amount of time (epoch).

Assuming delay tolerance and resource limitation permits us to emphasize delivery ratio and resource usage under resource constrained profiles. Energy savings and increased delivery ratio at the cost of latency is a beneficial because packet delivery prior to TTL is the primary metric.

4.2 Forwarding Algorithm

We employ communities to eliminate the routing table; community affiliations determine whether a message is to be forwarded or retained. CBF uses BUBBLE's *local popularity* (Eq. (1)) for intra-community delivery. For 'inter-community' routing, BUBBLE's *global popularity* (Eq. (2)) is replaced by *community betweenness count* (CBC) and *nodal contribution factor* (NCF) (Eqs. (3) and (4)). CBC is the number of contacts between two communities; NCF(n,C) is a node's contacts with every other community. In all equations, $g(x, y, k) = 1$ if an encounter between nodes x and y occurs in the k^{th} measurement interval, and 0 otherwise, C is a node's community, k is a particular aggregation interval within the epoch and K is the epoch duration. Epoch e's data is used in forwarding decisions in epoch $e + 1$.

Local Popularity:

$$\forall(x) \quad LP_x = \sum_{y \in C_x} \sum_{k=0}^{K} g(x, y, k) \tag{1}$$

Global Popularity:

$$\forall(x) \quad GP_x = \sum_{y \notin C_x} \sum_{k=0}^{K} g(x,y,k) \tag{2}$$

Community Betweenness Count:

$$\forall(C_x, C')_{s.t.(C_x \neq C')} \quad CBC_{C_x,C'} = \sum_{x \in C_x} \sum_{y \in C'} \sum_{k=0}^{K} g(x,y,k) \tag{3}$$

Nodal Contribution Factor:

$$\forall(x)\forall(C')_{s.t.(x \notin C')} \quad NCF_{x,C'} = \sum_{y \in C'} \sum_{k=0}^{K} g(x,y,k) \tag{4}$$

A formal algorithm representation of CBF is shown in Algorithm 1. In each timestep, a node encounters a (possibly empty, but normally small) set of other nodes and potentially transfers messages to these nodes. A node may also receive messages as the encountered nodes simultaneously execute CBF.

Algorithm 1 CBF (Node me, Node dest, Node metNode[], int numEncountered, msgType Msg)

```
Node maxCBC= me; Node maxNCF=me;
Node maxLP = NULL;
for (i = 1 to numEncountered) do
    if (metNode[i] == dest) then // Destination
        dest.addMessageToBuf(Msg); return;
    end if
    if (C(metNode[i]) == C(dest)) then
        maxLP = maximumLP(metNode[i], maxLP);
    else if (C(metNode[i]) ≠ C(me)) then
        maxCBC = maximumCBC(metNode[i], maxCBC);
    else maxNCF = maximumNCF(metNode[i], maxNCF);
    end if
end for
if ((maxLP ≠ NULL) & (C(me) ≠ C(dest))) then // entering dest. comm.
    maxLP.addMessageToBuf(Msg)
else if ((maxLP ≠ NULL) & (LP_maxLP > LP_me)) then
    maxLP.addMessageToBuf(Msg) // in dest. comm.
else if (maxCBC ≠ me) then
    maxCBC.addMessageToBuf(Msg) // new comm.
else if (maxNCF ≠ me) then
    maxNCF.addMessageToBuf(Msg) // same comm.
end if
```

Packet forwarding is accomplished through a set of heuristics. We never forward outside the community when the carrier is in the destination community, and and choose the node with the greatest LP value as the next carrier. When the carrier encounters a node that neither belongs to its own community nor to that of the destination, CBC values with the destination community are used, selecting the encountered node with the maximum CBC. Finally, when the encountered node and carrier are in the same non-destination community, NCF with the destination community is used.

No routing loops are possible during an epoch, as only nodes with higher delivery metrics are chosen. It is possible that a carrier node will change communities between epochs. A message may return to a previous community, as a type of backtracking. With TTL less than the epoch, at most one backtracking operation per message is possible for messages in-transit at epoch end.

4.3 Comparator Algorithms

We compare CBF with BUBBLE as an implementation of a context-aware forwarding approach and Epidemic Routing (ER) [30], a commonly considered performance benchmark algorithm ([15, 20, 21, 22]). Additionally, we are interested in determining how both algorithms perform in comparison to oracles that are optimal with respect to two of our main metrics: delivery latency and number of message transmissions.

We use two oracles: the *Minimum Cost Oracle* and the *Fastest Oracle*. For the minimum cost oracle we take the shortest of all ER paths that successfully delivered a packet to the destination. The minimum cost oracle should approach one hop for a fully connected graph as time goes to infinity. Values greater than 1 indicate both the small world nature of the graph and the time horizon on simulation forced by the duration of the datasets considered. The fastest oracle records the first arrival at the destination using ER with unlimited resources.

4.4 Clustering Algorithm

The K-Cliques algorithm has been used for detecting communities in different kinds of networks and was used in BUBBLE [15]. It requires the minimum size of communities to be specified prior to forming the communities. Outlier nodes are placed in communities by K-cliques, but should be left isolated to avoid poor intracommunity routing decisions.

Inspired by this concern and Pietilänen and Diot's methods [24], we used the Louvain algorithm [4] to find communities characterized by frequent, sustained contacts. Louvain clustering is fast, simple to implement and does not require a predefined minimum community size. Extensive evaluation on a variety of datasets shows superior performance in terms of modularity/centrality than comparable techniques, though the communities obtained are not always identical. We believe the small number of nodes and similar contact properties make this an appropriate clustering algorithm and that Louvain clustering will cause CBF to ignore isolated individuals. Future work will compare routing performance with different clustering algorithms.

The Louvain algorithm works in multiple iterations, each consisting of two phases. During the first phase, each node is considered as a separate community. In each iteration, every node is selected and potentially merged with each of its neighbouring communities to see if the merger improves network *modularity*. If no potential merges improve the modularity, the algorithm stops. During the second phase, new community formed in phase 1 is converted to a single node, represented by some centroid value. The phases are repeated until a locally near optimal point or a proscribed number of iterations is reached. The algorithm is guaranteed to converge, but may not be optimal, as the greedy approach is a heuristic solution to this NP-hard problem.

Since the graph is dynamic, and evolves over time, communities should be refreshed periodically.[1] Community formation based on a very short period or number of contacts may not reflect the actual graph structure (e.g., when only day 1's contacts are used, 33 communities are generated for SHED1 as few contacts occurred on that day).

[1]refresh interval depends on stability of contact patterns

5. EXPERIMENTAL DETAILS

Our first dataset, Flunet [11], contains contacts information for 36 computer science graduate students at the University of Saskatchewan, as well as staff and undergraduate students associated with those labs, collected over a period of 3 months. Approximately 70,000 contact records were collected from wireless sensor motes (MicaZ). Our second data set is St. Andrews (Sassy) [3] measuring the contacts of 22 undergraduate students, three postgraduate students and two staff members of the University of St. Andrews for 79 days (similar to Flunet) with 113,000 contacts. Our last data set - the Saskatchewan Human Ethology Dataset 1 (SHED1) [10] - covers 5 weeks of Bluetooth contact records of 39 participants who were primarily CS graduate students and staff (22721 distinct contacts).

A custom simulator was developed for our experiments as the integration of clustering and routing was simpler and could be more focussed than existing simulators such as ONE [17] and OMNET++.[2] Source and destination were chosen randomly from different communities to focus on the impact on inter-community message passing. Twenty experimental runs were performed for each parameter combination.

The values assigned to each input parameter are described in Table 1. In all experiments, a single parameter was varied. For all datasets, in the limited resource experiments, 10% of the number of messages generated was used as the fixed buffer capacity; trial and error showed that increased buffer did not provide a proportionate increment in delivery ratio for either algorithm. Similarly, maximum TTL values were set to 15 days for Flunet/Sassy and 7 days for SHED1. Unless otherwise specified, 100K, 80K and 48K messages were generated for Flunet, Sassy, and SHED1, respectively.

Table 1: Experimental Factor Input Ranges

Input	Limited Resource		
	Flunet	Sassy	SHED1
TTL(hours)	1-336	1-336	1-168
Buffer Capacity	10% msgs	10% msgs	10% msgs
Messages	10-80000	10-100000	10-480000
	Unlimited Resource		
	Flunet	Sassy	SHED1
TTL(hours)	1-4320	1-4320	1-2160
Buffer Capacity	10-200000	10-200000	2-100000
Messages	10-80000	10-100000	5-48000

We first investigated algorithm performance without resource constraints to establish best case performance baselines where one parameter was varied and the others set to inexhaustible values. The limited resource cases constrained parameters within a limited range. A cool-down period is not used; no algorithm delivers messages generated late in the simulation. The first epoch's data is used as a training session; packets are forwarded starting in the second epoch. More details can be found in Rasul [26].

6. PERFORMANCE EVALUATION

6.1 Community Determination

Our data sets ranged from extremely clustered to more isolated. Further examination of the datasets shows that between 62% and 79% of the encounters are local encounters (intra-community) and 21% to 38% are inter-community (i.e. global encounters). Intra-community message passing is expected to have short/fast paths for all heuristics, as nodes are placed communities by contact frequency. This intuition to optimize inter-community delivery is additionally inspired by the work done by Sastry *et al.* [27] where routing performance has a crucial dependence on the less frequent contacts. In all our datasets, global encounters can be treated as the somewhat *'rare'* or *'novel'* contacts.

Figure 1 shows the total number of membership changes for different community formation periods for all 3 datasets. The number of changes decreases with increasing time period. Membership changes are frequent for very short periods.

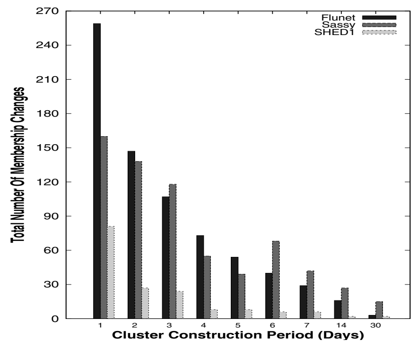


Figure 1: Membership Changes

We also measured the average number of communities for different community formation periods. The number of communities stabilizes for SHED1 and Flunet after 4 and 7 days, respectively. In Sassy, the number of communities varies slightly, regardless of the measurement period. We choose 7 days as the epoch duration, since it fits two of the three datasets well. A notable consequence of an epoch duration of 7 days and a maximum TTL of 15 days (as in the limited resource experiments) is that messages exist in at most 3 epochs before expiring, for a maximum of 2 community changes for message undelivered during re-clustering.

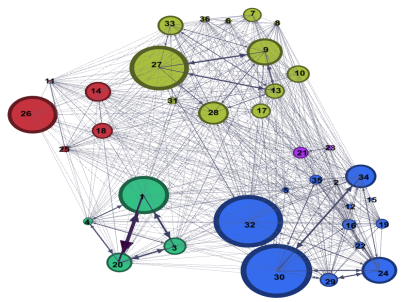
6.2 Dataset Characteristics

We investigated the overall internal network structure for each dataset by sending 5000 inter-community messages. Figure 2 provides an overall idea of the delivery behaviour of each of the algorithms, visualized in communities formed over all contacts. For all detailed forwarding experiments, weekly community membership is used and will not exactly match the aggregated membership visualized here.[3]

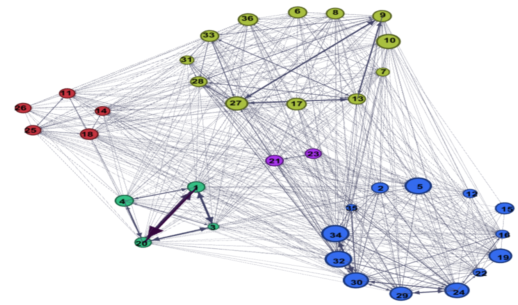
In particular, we see the underlying network structure, the contact patterns and the message forwarding behaviour. Since the same community determination and same contact patterns are used for both BUBBLE and CBF, network connectivity remains unchanged. Encounters are represented as edges in the graph where edge thickness is proportional to the number of encounters between nodes; node size is proportional to the number of packets forwarded. BUBBLE has nodes that transmit many messages, while load is spread

[2] www.omnetpp.org

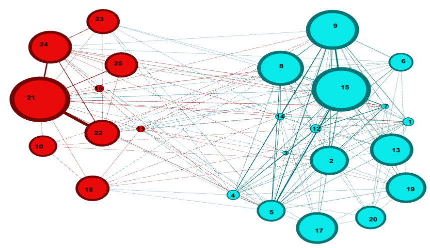
[3] e.g. weekly data for Sassy produces 5 communities in each week, whereas contacts aggregated over the entire dataset produce only 2 communities.



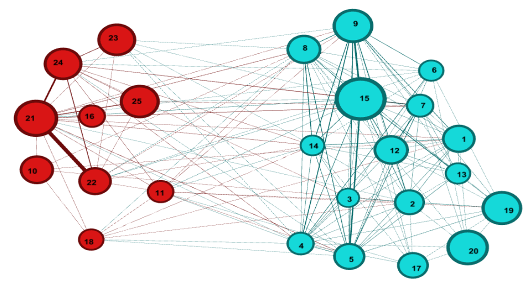
(a) (Flunet): BUBBLE



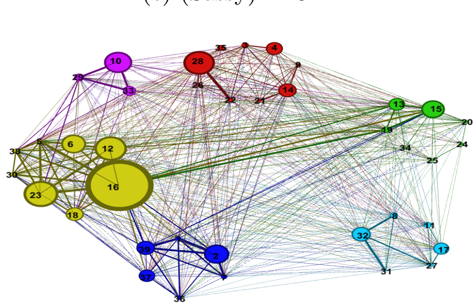
(b) (Flunet): CBF



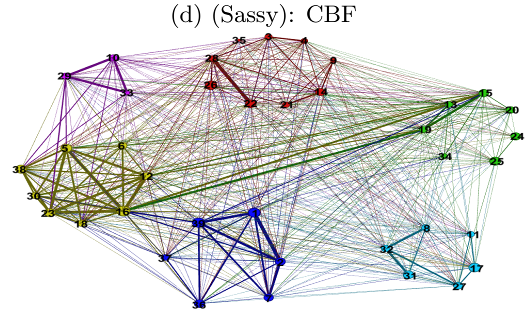
(c) (Sassy): BUBBLE



(d) (Sassy): CBF



(e) (SHED1): BUBBLE



(f) (SHED1): CBF

Figure 2: Internal Network Structure

more evenly in CBF. Further analysis showed that the median number of messages forwarded per node was similar, but some nodes transmitted disproportionately many messages using BUBBLE, especially in Flunet/ SHED1, implying high energy usage and potential buffer overflow at the over-utilized nodes. In particular, 25% of the nodes transmit over 45,000/22000 messages, using BUBBLE. CBF has no node send over 30000/15000 messages.

6.3 Forwarding Results

Delivery Ratio. In the first set of experiments we varied buffer size, but TTL was unlimited and the number of messages was generated as in Table 1. Figure 3 shows the mean (bar) and the standard deviation (whiskers) for the delivery ratio. The differences between the algorithms were not significant for unlimited buffer sizes (80%) for Flunet and SHED1. With large buffer (over 100 KB), ER performed the best (90%). For Sassy, delivery ratios were worse overall, because of the instability of the network, but CBF had about 5% higher delivery ratio than BUBBLE for buffer sizes above 2.5 KB. ER with unlimited buffer and TTL delivers the maximum number packets.

We expect that CBF will deliver more messages than BUBBLE by having fewer packet deletions due to buffer overflow. Therefore, we examined the impact of varying buffer size keeping TTL and number of messages constant. Figure 4 shows delivery ratio as a function of buffer space. At moderate buffer space (500 to 1000 msgs), CBF increases delivery ratio by between 30% and 40% for Flunet/SHED1. For Sassy, delivery ratio is improved by 16%. At larger buffer sizes, there was still an almost 20% improvement for Flunet/SHED1. In the limited environments, ER quickly saturates all node buffers and its delivery ratio plummets.[4]

Next, TTL was varied, constraining the buffer and number of messages as specified above. Increasing TTL in ER decreases delivery ratio, likely since fewer packets expire, causing potentially deliverable packets to be dropped. For BUBBLE and CBF, delivery ratio increases similarly to increasing buffer size as TTL is varied between 24, 72, 168, and 336 hours. Due to space considerations, the graph is not shown. The instability of the community associations in Sassy causes slightly poorer performance for both BUBBLE and CBF under constrained resources, as the fundamental assumptions about network structure are violated.

[4]The remaining graphs omit ER, because messages transmitted and dropped packets are substantially higher.

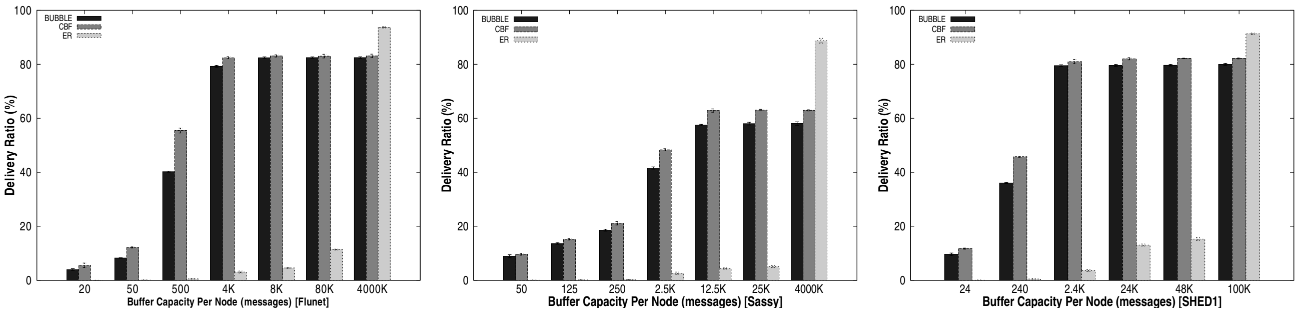


Figure 3: Delivery Ratio: Unlimited Resource Environment

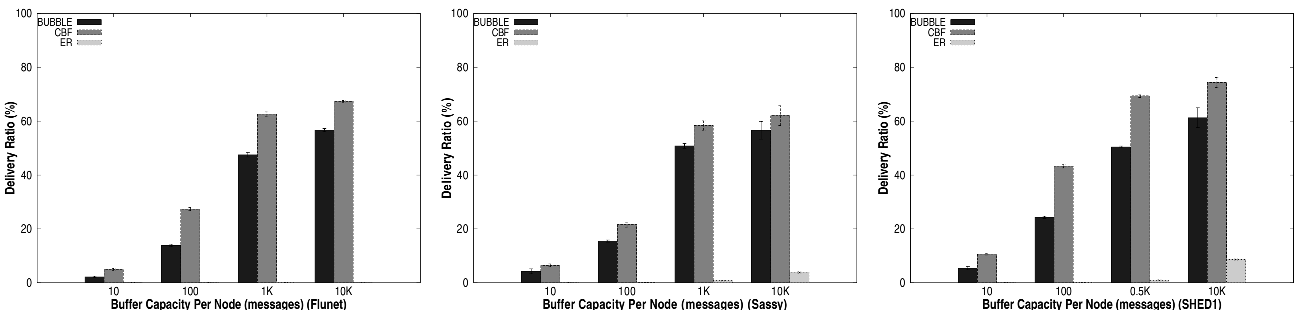


Figure 4: Delivery Ratio: Limited Buffer Capacity

Latency. Each subgraph of Figure 5 shows the results for a sample run at a particular level of load. All runs had a similar latency profile. As the number of messages increases, the oracle delivers more messages closer to TTL expiry, increasing the latency profile. For BUBBLE and CBF, however, the concentration of simultaneous live messages increases; more messages are delivered to their respective destinations faster, decreasing average delay.

For the majority of messages successfully delivered, CBF has a lower delay. In particular, for the sample run of 500 messages with SHED1, 310 out of the fastest 390 messages had a lower latency for CBF (by between 10 and 15 hours). Other datasets show a similar trend: CBF slightly outperforms BUBBLE for a majority of the low-latency messages, and then suffers high latency for the hard-to-deliver messages. In these cases, a high contribution to total delay is provided by that relatively small proportion of such messages, plus those that BUBBLE could not deliver, but that CBF did deliver, albeit slowly.

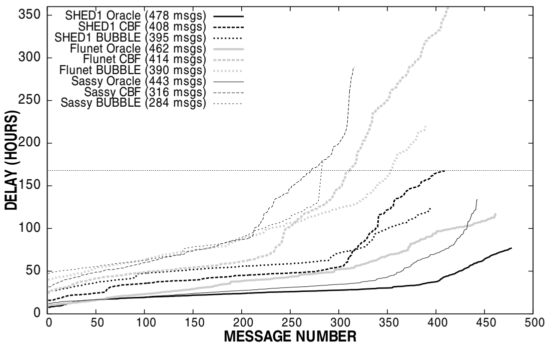
Over all experimental runs, the CBF median delay is between 73% and 97% of BUBBLE's (500 messages), between 66% and 90% (5000 messages) and between 61% and 89% (40,000 messages). For lower loads, SHED1 has the smallest relative median delay and Sassy the largest. With 40,000 messages, SHED1 actually had the largest relative median delay, but the median for BUBBLE was substantially lower than with 5000 messages (reduced from 53 to 37 hours).

When we consider the average delay of the *same number of messages* delivered by BUBBLE and CBF, there is very little difference between the algorithms (some runs produce at most 7% difference for the 500 message case). For SHED1, CBF delay is 7.5% to 12% lower than BUBBLE. For smaller numbers of messages, BUBBLE outperformed CBF for Sassy and Flunet by as much as 6%, but with the 40000 message case, Flunet's CBF delay was only 83% of BUBBLE's. When the additional messages that only CBF could deliver are included, the mean delay increases. For 5000 messages and 40000 messages, we see a similar trend, but the delay measurements are even closer between CBF and BUBBLE. Sometimes CBF even surpasses BUBBLE. In particular, for the SHED1 dataset, the average delay is 1.8 hours less under CBF than BUBBLE with 5000 messages. The worst performance for CBF was a 21% increase in delay for Flunet (500 messages), though there was only a 6% increase for the fastest 390 messages delivered.

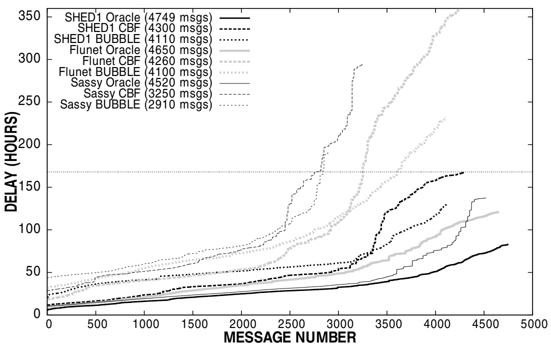
For CBF with Flunet, messages were continued to be delivered until TTL expiry, but not with BUBBLE, whose longest successful delivery was 230 hours. Similar results are shown for SHED1. Even more dramatic are the results for Sassy, where there were no messages delivered close to the TTL expiry time in either forwarding scheme, the longest successful latency being 294 hours for CBF. Extending the TTL does not necessarily allow more messages to be delivered, even with unlimited buffer space. In the samples, only Flunet using CBF has a substantial percentage of messages delivered after 7 days.

Message Transmissions. We next compare the efficiency of BUBBLE and CBF with the *Minimum Cost Oracle*, measuring the total number of transmissions required to deliver the same set of messages unconstrained. Table 2 shows that CBF uses 2 more hops on average than the oracle in SHED1, whereas BUBBLE uses 3 more hops. The differences between CBF and BUBBLE are less for the other 2 datasets.

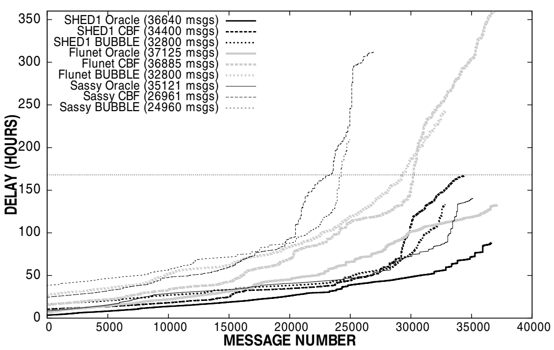
After comparing with the oracle, we compared the forwarding cost between the algorithms in a realistic, resource-constrained environment. Figure 6 shows transmissions as a function of buffer space, indicating absolute differences of transmissions between CBF and BUBBLE for larger buffer sizes. In this scenario, CBF also had a higher delivery ratio (see Figure 4), so the transmission is more efficient.



(a) 500 Messages



(b) 5000 Messages



(c) 40000 Messages

Figure 5: Message Latency (Sample Runs)

By increasing TTL, the lifespans of messages were increased, allowing them undergo more exchanges before getting dropped, shown in Figure 7. Increasing the TTL has a diminishing effect on the *difference* between number of messages sent by each algorithm. For both algorithms, SHED1 has no noticeable increase in transmissions with a TTL longer than 7 days, and doubling the TTL (from 168 to 336 hours) for the other 2 datasets has the effect of increasing the average number of transmissions by only 25-28% in CBF and only 12-18% in BUBBLE. Doubling it again had less than a 10% effect in any of the algorithms. The fact that SHED1 has a stable number of transmissions indicates that all messages were delivered before the original TTL, were victims of buffer overflow, or found no suitable carriers.

For all datasets, CBF outperforms BUBBLE by transferring more messages with fewer transmissions. Results using the SHED1 dataset shows the maximum difference, whereas the least difference is seen in Sassy.

Table 2: Hop Count

Msgs	Algorithm	Flunet	Sassy	SHED1
50	Oracle	2	1.6	2
	BUBBLE	4.8	4.6	5
	CBF	4.6	4	4
500	Oracle	1.2	1.4	2
	BUBBLE	4.4	4.3	5.1
	CBF	4	4.2	3.5
5000	Oracle	1.1	1.33	2
	BUBBLE	4.33	4.2	5.06
	CBF	4	4.13	3.67
40000	Oracle	1.06	1	2
	BUBBLE	3.83	5	5
	CBF	3	4	4

Packets Dropped. As suggested by Figure 2, CBF spreads traffic more evenly, reducing buffer congestion. Table 3 shows the average and standard deviation of packet drops when TTL is varied and buffer space is limited. Social nodes in BUBBLE experience more packet drops. The standard deviation in packet drops is comparable, but is usually lower with CBF. Both algorithms have nodes drop older packets to make space for newer messages. As TTL is increased, more long-lived packets must be dropped due to congestion. With short TTL, CBF has close to an order of magnitude fewer drops for all datasets. This difference drops to less than twice for the longest TTL. At the default TTL (168 hours), the differences range from 33% (Flunet) to 60% (Sassy).

Table 3: Packet Drops

TTL (hrs)	Alg.	Flunet		Sassy		SHED1	
		Mean	SD	Mean	SD	Mean	SD
24	BUBBLE	341	9	208	31	3693	20
	CBF	64	18	20	5	267	21
72	BUBBLE	2135	38	1831	68	11163	92
	CBF	491	24	785	14	6754	87
168	BUBBLE	12486	52	14080	133	17418	507
	CBF	4150	71	8232	102	9171	332
336	BUBBLE	20666	230	26436	784	17658	3650
	CBF	12212	191	17384	483	9210	3811

In SHED1, where the top four social nodes belong to the same community, there is higher packet drop. For all three datasets, Pearson's correlation coefficient was calculated to determine the impact of node GP on the number of packets dropped. Packets dropped by nodes using BUBBLE have a high positive correlation with global popularity, whereas with CBF packet drop has a low positive correlation with global popularity. In Sassy, nodes using BUBBLE have a high correlation of +0.85, but using CBF have a low correlation of +0.41. These values are +0.90/+0.70 and +0.26/0.30 in SHED1/Flunet, respectively. In a realistic limited resource environment, the dependency of BUBBLE on the most social nodes becomes its major drawback rather than its strength.

Analysis of Forwarding Dynamics. To illustrate the underlying mechanics which lead to the trends observed, we consider a final experimental scenario. A fixed number of messages, TTL and and 10% buffer capacity was employed as in Section 5. With lower packet transmission for CBF than BUBBLE (Figures 6 and 7), a higher proportion of passes (12.34% - Flunet, 5.87% - Sassy, 17.91% - SHED1) were made based on inter-community factors in CBF than in BUBBLE, indicating that CBF prefers an appropriate node along a likely path rather than a globally popular node.

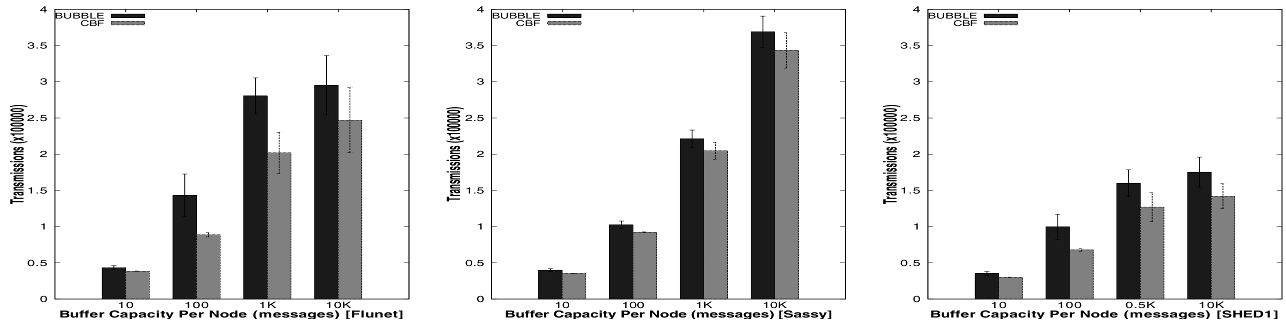


Figure 6: Transmissions: Limited Buffer Capacity

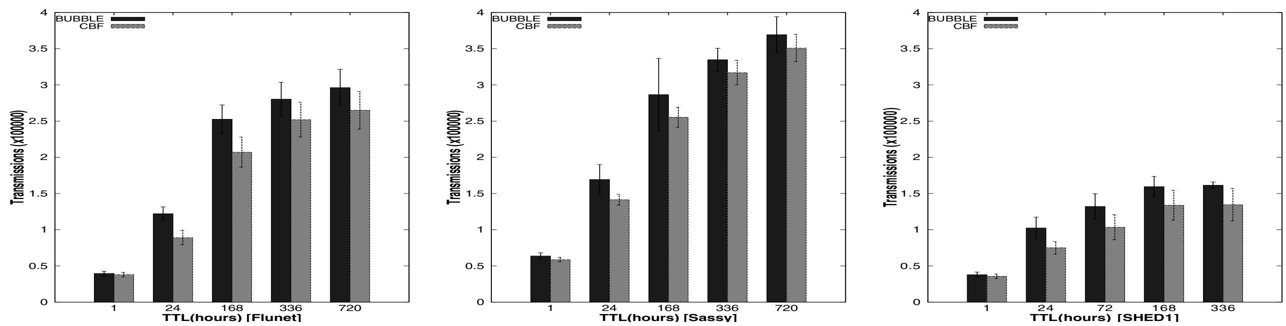


Figure 7: Transmissions: Limited TTL

7. DISCUSSION

Contributions. Our primary contribution is the CBF algorithm which uses more nuanced information about social network structures arising from human mobility patterns in forwarding decisions. We also analyzed the role of network versus personal interconnectivity in PSNs. Well-connected nodes may span communities, but nodes close to the destination bridging relevant communities are important in reducing packet drops in resource constrained systems and asymmetric resource utilization on globally popular nodes compared to BUBBLE [15] or PRoPHET [20].

By employing datasets with varying degrees of community stability, we established that even under varying community membership, CBF can provide superior performance to BUBBLE. Obviously, there are limits to the extent to which our assumptions apply to degenerate systems: for example contact networks at immigration points where few people will ever see each other again, will require different routing systems. Finally, by comparing the limited and unlimited cases, we quantified the degree to which resource constraints impact the performance of PSN algorithms. Severe performance degradation for greedier schemes was noted, as expected. CBF increases the delivery ratio by up to 40% as the number of messages and TTL were varied in the limited resource situation. When buffer capacity was varied, delivery ratio and transmissions were consistently better for CBF than BUBBLE; packet loss was lower with CBF as well.

Limitations and Future Work. While constituting a significant contribution to the study of PSN routing, there are several short-comings to be addressed in future work. Our datasets are strongly biased toward University lifestyles and relationships. Replicating these results with datasets from other populations or on synthetic data [12] could shed further light on design tradeoffs. We also neglected channel noise, a common assumption in DTN work, as packet drops are to some extent implicitly encoded in the datasets. Modelling communication channel noise and investigating the role of packet size would substantially extend our work. Message source and destination were generated at random from different communities. More realistic message generation scenarios would help quantify the applicability of the results. In particular, the many-to-one scenario should show different packet drop profiles. Finally, we only investigated a single community generation technique. Additional community formation techniques and even self-identification may enhance CBF generalizability.

8. CONCLUSIONS

Social networks often exhibit small world network traits. For efficient routing in social contexts, forwarding algorithms can use routing heuristics that encompass small world network properties. Moreover, in a resource constrained PSN system, the routing algorithm must use available resources appropriately. Our algorithm balances the use of resources with the likelihood of delivery, by employing decision making metrics that can evaluate arising social structure of a network, in turn enabling CBF to preferentially choose nodes with a greater probability of being in the target community. When compared to BUBBLE, CBF's transmission performance reaches closer to that of the shortest route oracle and in all cases CBF transmits fewer messages, making it more energy-efficient. Quantitative analysis shows that more stable networks reap the benefits of CBF's enhancements to a greater degree. The delivery ratio of CBF is bounded below by BUBBLE and we experience acceptable additional delay for a DTN, due to the additional messages delivered.

9. REFERENCES

[1] L. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of Small-world Networks. *Proceedings of the National Academy of Science USA*, 97:11149–11152, 2000.

[2] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine. Relays, base stations, and meshes: enhancing mobile networks with infrastructure. In *ACM Mobicom*, pages 81–91, San Francisco, CA, Sept. 2008.

[3] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti. Exploiting Self-Reported Social Networks for Routing in ubiquitous Computing Environments. In *WIMOB*, pages 484–489, Avignon, France, Oct. 2008.

[4] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[5] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot. The Diameter of Opportunistic Mobile Networks. In *CoNEXT*, pages 12:1–12:12, Copenhagen, Denmark, Dec. 2007.

[6] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs. In *ACM Mobihoc*, pages 32–40, Montreal, Canada, Sept. 2007.

[7] X. F. Guo and M. C. Chan. Plankton: An efficient DTN routing algorithm. In *IEEE SECON*, pages 550–558, New Orleans, LA, June 2013.

[8] Z. J. Haas and T. Small. A new networking model for biological applications of ad hoc sensor networks. *IEEE/ACM Transactions on Networking*, 14(1):27–40, Feb. 2006.

[9] M. Hashemian and K. G. Stanley. Effective utilization of place as a resource in pocket switched networks. In *Local Computer Networks*, pages 247 –250, Bonn, Germany, Oct. 2011.

[10] M. S. Hashemian, K. G. Stanley, D. L. Knowles, J. Calver, and N. Osgood. Human network data collection in the wild: the epidemiological utility of micro-contact and location data. In *ACM SIGHIT International Health Informatics Symposium*, pages 255–264, Miami, FL, Jan. 2012.

[11] M. S. Hashemian, K. G. Stanley, and N. Osgood. Flunet: Automated tracking of contacts during flu season. In *WiOpt*, pages 348–353, Avignon, France, May 2010.

[12] T. Hossmann, T. T. Spyropoulos, and F. Legendre. Putting contacts into context: mobility modeling beyond inter-contact times. In *ACM Mobihoc*, pages 18:1–18:11, Paris, France, May 2011.

[13] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *ACM SIGCOMM Workshop on Delay-tolerant Networking*, pages 244–251, Philadelphia, PA, Aug. 2005.

[14] P. Hui and J. Crowcroft. How Small Labels Create Big Improvements. In *PERCOM Workshops*, pages 65–70, White Plains, NY, Mar. 2007.

[15] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *ACM Mobihoc*, pages 241–250, Hong Kong, China, May 2008.

[16] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *ACM SIGCOMM*, pages 145–158, Portland, OR, Sept. 2004.

[17] A. Keränen, J. Ott, and T. Kärkkäinen. The one simulator for dtn protocol evaluation. In *Simutools '09*, pages 55:1–55:10, Rome, Italy, 2009.

[18] S. Khan, R. Mondragon, and L. Tokarchuk. Lobby Influence: Opportunistic forwarding algorithm based on human social relationship patterns. In *IEEE PERCOM Workshops*, pages 211–216, Lugano, Switzerland, Mar. 2012.

[19] K. Lee, S. Hong, S. Kim, I. Rhee, and S. Chong. SLAW: self-similar least-action human walk. *IEEE/ACM Transactions on Networking*, 20(2):515–529, Apr. 2012.

[20] A. Lindgren, A. Doria, and O. Schelen. Probabilistic Routing in Intermittently Connected Networks. *SIGMOBILE Mobile Computing and Communication Review*, 7(3):19–20, 2004.

[21] A. Mtibaa, M. May., C. Diot, and M. Ammar. Peoplerank: Social opportunistic forwarding. In *IEEE INFOCOM*, pages 1–5, San Diego, CA, Mar. 2010.

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

[23] A. Nguyen, P. Senac, and M. Diaz. How Disorder Impacts Routing in Human-centric Disruption Tolerant Networks. In *Workshop on Future Human-centric Multimedia Networking*, pages 47–52, Hong Kong, China, Sept. 2013.

[24] A. Pietilänen and C. Diot. Dissemination in opportunistic social networks: the role of temporal communities. In *ACM MobiHoc*, pages 165–174, Hilton Head, SC, June 2012.

[25] V. Pulimi, T. Paul, K. Stanley, and D. Eager. Near-optimal Routing for Contour Detection in Wireless Sensor Networks. In *Local Computer Networks*, pages 462–469, Clearwater, FL, Oct. 2012.

[26] K. Rasul. Community-based forwarding for low-capacity pocket switched networks. Master's thesis, University of Saskatchewan, 2013.

[27] N. Sastry, K. Sollins, and J. Crowcroft. Delivery properties of human social networks. In *IEEE INFOCOM Miniconference*, pages 2586–2590, Rio de Janiero, Brazil, Apr. 2009.

[28] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in intermittently connected mobile networks: The single-copy case. *IEEE/ACM Transactions on Networking*, 16(1):63 –76, Feb. 2008.

[29] V. Srinivasan, M. Motani, and W. T. Ooi. Analysis and implications of student contact patterns derived from campus schedules. In *ACM Mobicom*, pages 86–97, Los Angeles, CA, Sept. 2006.

[30] A. Vahdat and D. Becker. Epidemic Routing for Partially-connected Ad Hoc Networks. Technical Report CS-2000-06, Duke University, Apr. 2000.