Hybrid Community-Based Forwarding: A Complete Energy Efficient Algorithm for Pocket Switched Networks

Khadija Rasul, Dwight Makaroff, and Kevin G. Stanley
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, CANADA S7N 5C9
Email: khr667@usask.ca, makaroff@cs.usask.ca, stanley@cs.usask.ca

***Abstract*—Sensor devices and the emergent networks that they enable are capable of transmitting information between data sources and a permanent data sink. 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. Several previously developed algorithms employ models which considers the popularity of individual nodes within communities and forward messages to nodes with higher probability of delivery according to some heuristic. In previous work, we developed Community-Based-Forwarding (CBF) that considers the interactions between communities as a factor in message forwarding. Using this 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. One disadvantage of CBF was an increased delivery latency for some subset of messages that could not be delivered using other algorithms. In this paper, we extend the semantics of CBF with the Hybrid CBF algorithm (HCBF) by optimizing forwarding inside communities by considering the social diversity (measured by Unique Interactions). We find that all performance metrics are improved with this heuristic on a representative set of human mobility traces, but most significantly the message delivery latency is substantially improved over the other algorithms studied.**

***Index Terms*—delay-tolerant networks; pocket-switched networks; social-based routing; resource-constrained devices**

I. INTRODUCTION

Much of the world's population uses many mobile computing devices every day. Such devices vary from smart-phones to personal medical monitors to smart badges. These devices vary in capabilities, missions, and form factors, but all require limiting energy consumption to prolong battery lifetime.

Often these devices are required to communicate with each other (for example a medical device responding to a query from a particular physician), with a base station (for example collecting data on human behaviour), or receiving messages from a base station (for example a smart badge firmware update). While all of these missions could be accomplished by transmitting to fixed infrastructure, bridging across existing networks such as the internet, and transmitting to the recipient, this may not always be the most efficient solution, particularly if base stations are rare. Instead many researchers [1], [2], [3], have looked into sending messages from device to device, leveraging the contact patterns of humans to make a series of small energy-efficient hops rather than a larger, more energy intensive hop to a base station.

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 arising from human mobility patterns [4], [1], [2]. Significant differences in contact patterns have been observed in environments for which datasets are available [5], [6], [7]. These dynamics suggest that routing based on inter-community contacts may aid routing performance. However, care must be taken when designing these kinds of interactions, as maintaining traditional routining tables can soon become more expensive than routing the data itself [8]. Device sensing methodology and parameters as well as associated software for obtaining data (potentially multi-modal sensor data) may need to adapt in ways impossible to capture in an *a priori* manner. As well, other 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 [9].

In previous work, we developed and evaluated a Community-Based Forwarding algorithm (CBF) [10] that explicitly uses community structure as well as individual node-based connections of previous social-based approaches, and did not use a routing table, minimizing overhead. Leveraging community linkages exploited lower-centrality "bridging nodes" [11] to reduce buffer and power strain on the popular nodes, we were able to achieve savings compared to a well known social-based algorithm (BUBBLE) [2], for all measured metrics, except for average delivery latency. The delivery latency distribution showed that most messages had substantially lower delivery times under CBF, and indeed a lower average latency for the same delivery ratio. CBF delivered more messages, but some hard-to-deliver messages experienced greater latency due to the conservative nature of the forwarding heuristics utilized.

We also noted that there was a substantial gap between the performance of CBF and that of the Fastest Oracle, which routed packets along the minimum latency path given future knowledge of contact patterns. There are two options that come to mind when considering latency improvements: 1) improving the node clustering approach and identifying more complex community structures (perhaps hierarchical communities), and 2) optimizing the performance when routing inside a community. We have chosen the second strategy for the development of the Hybrid Community-Based Forwarding (HCBF) algorithm. In particular, we focus on improving the routing of packets within communities at every stage of the forwarding path, but most importantly, after a packet has arrived at a bridge node. This is distinct from general intra-community routing, as packets may not have to pass through bridging nodes in general, but do have to pass through bridging nodes when arriving from other communities. While we expect that the local routing heuristics employed in HCBF will improve routing quality in the more general intra-community routing case, that is not examined in this work.

We make the following contributions in this paper: 1) an optimized heuristic for intra-community forwarding, and 2) a quantitative comparison of the impact of limiting node resources on routing performance of the algorithms.

II. Background and Related Work

The potential benefits and tradeoffs of DTN routing policies were first examined by Jain *et al.* [3]. The simplest, Epidemic routing (ER), [12] where all nodes attempt to deliver all packets, is useful theoretically by providing minimum delay, but is horribly inefficient in practice. Constraining the growth of multiple copies is possible through adaptive limitations on packet time-to-live (TTL) [13] or limiting copies to privileged nodes [14], [15]. However, if data is truly delay tolerant the benefit of multi-copy transmission is unclear.

The single copy case [16] can exhibit high latency and potentially poor delivery ratio, but is often more efficient due to a lack of multiple transmissions. Lindgren *et al.* proposed PRoPHET [8], which models nodes' future contacts directly from contact history. Context-Aware Adaptive Routing (CAR) [17] forwards packets to nodes with the highest probability of meeting the destination. Plankton [18] predicts the probability of future contacts and duration by classifying links based on the quality of previous contacts.

Other researchers have attempted to gain efficiency by modelling the overall behaviour of the dynamic contact graph by segmenting the graph into cliques, which we have called communities. SimBet [11] employs betweenness centrality and similarity metrics. Hui *et al.* [2] developed BUBBLE, which uses time-variant rankings based on recent history. 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.

Peoplerank [19], motivated by Google's PageRank algorithm [20], uses node-ranking concepts based on social behaviour. Fabbri and Verdone utilize metrics for sociability of individual nodes in a vehicular networks [21]. Higher contact-rate, lower contact-duration nodes have been shown to play a major role in efficiently forwarding data [22]. In Lobby-Influence [23], the influence of the community structure of the dynamic network dominated the routing protocol employed.

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 [24]. This structure has been used in attempts to increase routing efficiency in PSNs [15], [2], [25] admitting that aggregate representation fails to capture instantaneous contact pattern dynamics. The periodicity of contact patterns has also been shown to influence the performance of DTN routing algorithms [26]. In particular, small world networks with highly connected clusters, containing short paths connecting every pair of nodes provide a promising means of improving routing [2]. CBF in general and HCBF in particular is based on the observation that strong paths within communities should provide fast intra-community routes and that there is a high probability for messages to be forwarded through nodes within the same 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 [7], [24] datasets; 2) inferred from higher-level mobility data such as class [27] or transit [14] schedules; or 3) from synthetic contacts generated directly from theoretically grounded mobility patterns [28]. The datasets employed in this work are representative of university environments, and are among the longest datasets available. Synthetic and mobility datasets may have different characteristics.

III. Algorithm

Previous analysis of the datasets shows that between 62% and 79% of the encounters occur between nodes in the same community [29]. This suggests that intra-community features might be exploitable to improve routing of packets to and from bridge node within communities. BUBBLE and CBF define node sociability or local popularity (LP) by the number of contacts a node has. *Diversity*, referring to the number of unique nodes that are encountered by a particular node, is potentially a more useful metric, as it encodes to propensity of a node to mix with others. If nodes with more diverse interaction patterns are considered, then the most diverse nodes will be selected as carriers, potentially reducing latency [22]. If LP is the only factor considered, there may be highly-active nodes that form strong subnets within the community, potentially leading to packets becoming stuck.

The use of community and popularity metrics to allows nodes to route messages locally, without routing tables. For intra-community delivery, HCBF uses Unique Interactions (UI) as opposed to BUBBLE's LP (Eq. (1)), which is also used in CBF. The values of UI and LP are not strongly correlated. For more details, see Rasul [29].

Unique Interactions

$$\forall(x \in C)\ UI_x = |S_x|\ where\ S_x = \bigcup_{y \in C_x} s.t.\ \exists g(x,y,k) = 1, k \leq K \quad (1)$$

A formal representation of HCBF is shown in Algorithm 1. It is identical to the CBF algorithm [10], except for the use of the intra-community heuristic (UI), based on diversity, but repeated here for completeness. In each time step, a node encounters a (possibly empty, but normally small) set of other nodes and potentially transfers messages to these nodes. Packet forwarding is accomplished through a set of heuristics. We never forward outside the community when the carrier is in the destination community. If the carrier encounters a node that neither belongs to its own community nor to that of the destination, community betweenness count (CBC) values with the destination community are used, selecting the encountered node with the maximum CBC. Next, when the encountered node and carrier are in the same non-destination community, the node with the highest nodal contribution factor (NCF) with the destination community is used. Otherwise, we choose the node encountered with the greatest UI value. Finally, LP is used.

We compare HCBF with BUBBLE as an implementation of a context-aware forwarding approach. 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. As delay is the primary metric we are attempting to minimize, we use an oracle that records the first arrival at the destination using ER with unlimited resources.

Algorithm 1 HCBF (Node me, Node dest, Node met[], int numEncountered, msgType Msg)

Node *maxCBC= me*; Node *maxNCF=me*;
Node *maxLP* = NULL;
Node *maxUI* = NULL;
for $(i = 1\ to\ numEncountered)$ **do**
  if (*met*[*i*] == *dest*) **then** // Destination
    addMessageToBuf(dest,Msg); **return;**
  end if
  if $((UI_{met[i]} > UI_{me})\ \&\ (C(met[i]) == C(me)))$ **then**
    maxUI = maximumUI(met[i], maxUI);
  end if
  if (*C(met[i]) == C(dest)*) **then**
    maxLP = maximumLP(met[i], *maxLP)*;
  else if (*C*(*met*[i]) $\neq$ *C*(*me*)) **then**
    maxCBC = maximumCBC(met[i], *dest*, *maxCBC)*;
  else *maxNCF = maximumNCF(met[i], dest, maxNCF)*;
  end if
end for
if ((*maxUI* $\neq$ NULL) & (*C(maxUI) == C(dest)*)) **then** // entering dest. comm.
  addMessageToBuf(MaxUI,Msg); **return;**
end if
if ((*maxLP* $\neq$ NULL) & (*C(maxLP) == C(dest)*)) **then** // entering dest. comm.
  addMessageToBuf(MaxLP,Msg); **return;**
end if
if (*maxCBC* $\neq$ *me*) **then**
  addMessageToBuf(maxCBC,Msg); **return;** // new comm.
end if
if (*maxNCF* $\neq$ *me*) **then**
  addMessageToBuf(maxNCF,Msg); **return;** // same comm.
end if
if ((*maxUI* $\neq$ NULL) & $(UI_{maxUI} > UI_{me})$) **then**
  maxLP.addMessageToBuf(Msg); **return;** // in any comm.
end if
if ((*maxLP* $\neq$ NULL) & $(LP_{maxLP} > LP_{me})$) **then**
  maxLP.addMessageToBuf(Msg); **return;** // else in dest. comm.
end if

IV. Experimental Details

Our first dataset, Flunet [7], contains contact 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) [5] 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) [6] - covers 5 weeks of Bluetooth contact records of 39 participants who were primarily CS graduate students and staff (22721 distinct contacts).

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 I. 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 [10]. Similarly, maximum TTL values were set to 15 days for Flunet/Sassy and 7 days for SHED1. Unless otherwise specified, 50K, 40K and 48K messages were generated for Flunet, Sassy, and SHED1, respectively. 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.

We are primarily concerned with improving routing performance in resource limited nodes such as smart badges [10]. To evaluate the impact of selected resource limitations, we alter the capabilities of nodes in a series of experiments, starting with no limitations, then analyzing the effect of limiting TTL and buffer size. For many of the reported metrics under each experimental condition, no meaningful differences between algorithms or heuristics could be found. In the interest of brevity, those results are not included in this paper. Metric and conditions which exhibit differential performance over algorithm are reported. To understand the relative impact of HCBF over CBF, we also compare results where only the intra-community portion of an inter-community routing task is considered.

TABLE I: 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	10-40000	10-50000	10-48000
	Unlimited Resource		
	Flunet	Sassy	SHED1
TTL(hours)	1-4320	1-4320	1-2160
Buffer Capacity	40-40000	50-50000	48-48000
Messages	10-40000	10-50000	10-48000

V. Performance Evaluation

A. Impact of Low-Diversity Social Nodes.

High-contact nodes are those that have a large number of contacts per unit time, and *social* nodes are those that spend significant amounts of time in a ‘temporal community’ [22]; i.e. repeated contacts with the same smaller set of nodes. Those with the higher contact rate are considered more diverse. In the first set of experiments, we compared BUBBLE and CBF with two versions of HCBF: one that considers diverse social nodes (HCBF), and one that considers all social nodes (HCBF-NS, for non-stationary). To test this method, 50000 messages were generated with fixed TTL and 10% buffer capacity for each of the three datasets. The average results are shown in Figure 1; one standard deviation is shown by the whiskers.

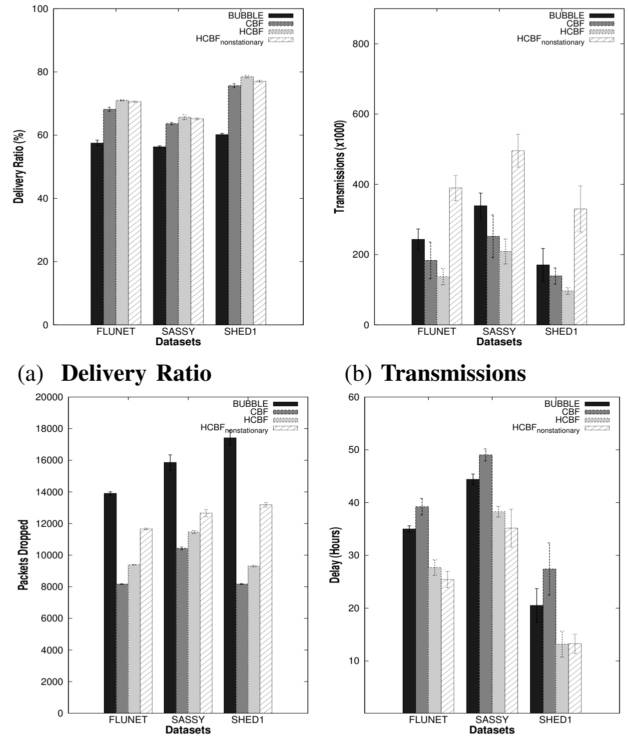
HCBF improves the delivery ratio slightly over HCBF-NS, and both are less than 2% better than CBF. In terms of packet drops, HCBF-NS has a larger number of dropped packets. The biggest problem with HCBF-NS is the large number of transmissions required (Figure 1 (b)), exceeding that of BUBBLE, and at least twice that of HCBF for all datasets.

Delivery latency is shown in Figure 1 (d). HCBF-NS has slightly lower delays than HCBF for Fluent and Sassy, but is slightly higher for SHED1. In all 3 datasets, the delay is better for both than for BUBBLE and CBF. For the remainder of the experiments, only HCBF will be used, since the delay improvement is not worth the increase in the energy required for the larger number of transmissions. Packets should be forwarded to highly mobile, high-diversity nodes, or socially mobile (SM) nodes.

B. Inter-community Routing

Messages are generated such that the source and destination belong to different communities. Figure 2 shows the delivery ratios for the SHED1 dataset. The other datasets showed similar results. The delivery ratio improves slightly over CBF.

When messages have longer TTL values, older undelivered messages are dropped out of a node’s buffer to make room for



(a) **Delivery Ratio** (b) **Transmissions**

(c) **Packet Drops** (d) **Delay**

Fig. 1: **HCBF-NS vs. HCBF Performance Measures**

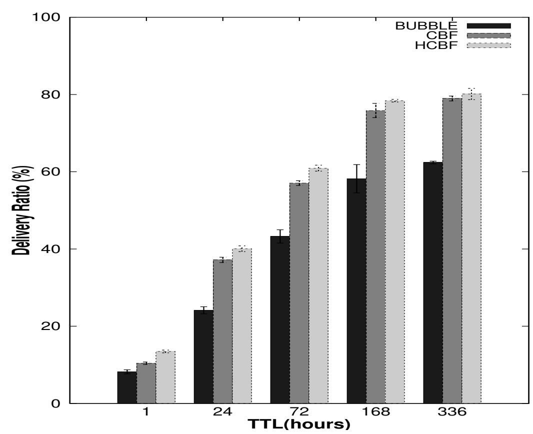


Fig. 2: **Delivery Ratio: Inter-community Messages (SHED1)**

newer ones. In Table II, drops increase dramatically between 24 and 72 hours, but only slightly between 7 and 14 days. The number of drops of CBF and HCBF are similar, but the variability between runs in HCBF is lower.

Change in TTL also has an impact on the transmissions. Figure 3 shows the transmissions for SHED1, and we can see that they are reduced dramatically. The socially-diverse mobile nodes use these characteristics to find the most suitable carrier in a shorter period of time than BUBBLE or CBF for the same TTL. The average delay increases as TTL

TABLE II: Packet Drops- HCBF and CBF

TTL (hrs)	Alg.	SHED1	
		Mean	SD
24	BUBBLE	3693	151
	CBF	267	106
	HCBF	238	12
72	BUBBLE	11163	648
	CBF	6754	322
	HCBF	6630	38
168	BUBBLE	17418	5312
	CBF	9171	3496
	HCBF	9310	56
336	BUBBLE	17658	350
	CBF	9210	566
	HCBF	10050	62

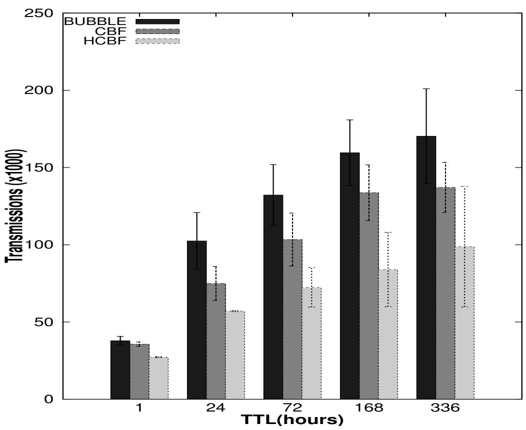


Fig. 3: **Transmissions: Inter-community Messages (SHED1)**

increases, but the difference between CBF and HCBF remains constant (Figure 4). In our previous work [10], we had noted that increases in delivery ratio for CBF came at the expense of additional latency. By improving intra-community routing through HCBF, we have ameliorated this shortcoming, consistently outperforming BUBBLE in all measured metrics.

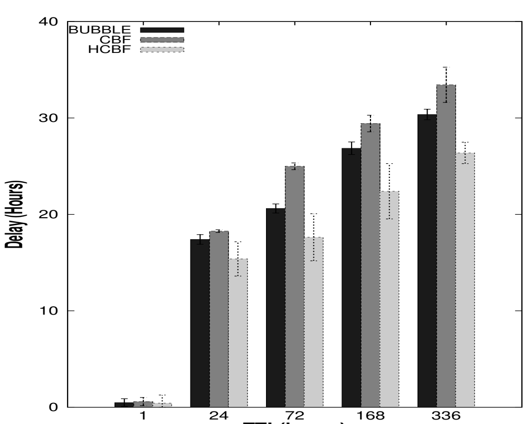
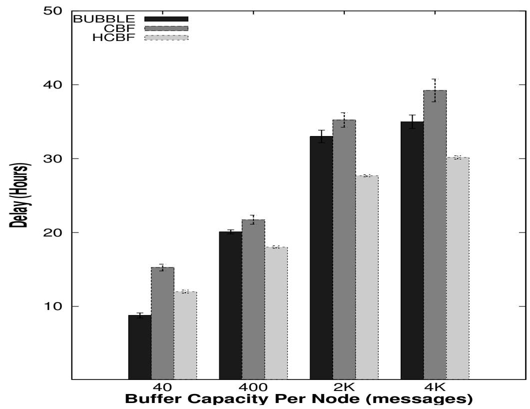
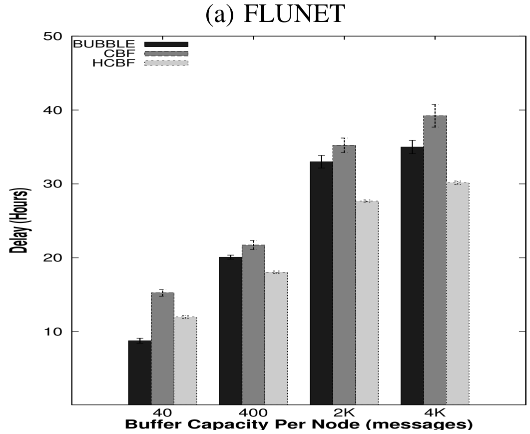


Fig. 4: **Latency: Inter-community Messages (SHED1)**

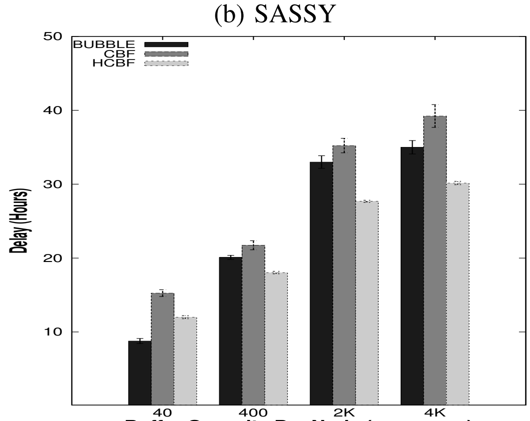
The next set of experiments consider nodes with limited buffer space. The delay decrease expected is confirmed by this set of experiments and shown in Figure 5. Delay increases as buffer capacity increases, as older packets can stay longer without being dropped and thus, be delivered eventually. In each case, the average delay for HCBF is less than that of BUBBLE for buffer capacities greater than 40 messages. Further detail on a sample run is described in the next subsection, along with a comparison with the Fastest Oracle.



(a) FLUNET



(b) SASSY



(c) SHED1

Fig. 5: **Message Latency - Inter-community messages**

Inter-community hops and intra-community hops have been traced and post-processed. The results are shown in Table III. For all delivered messages, CBF has one of these in common with the other 2 algorithms, so it is not included. HCBF requires on average at least 2 fewer hops than BUBBLE. The range of intra-community hops is shorter; allowing messages on difficult paths to have fewer hops. When compared to other datasets, messages in SHED1 using HCBF have the tightest intra-community hop distribution, because SHED1 has more nodes with higher UI values.

TABLE III: Hop Count

Dataset	Alg.	Intra-comm. Hop		Inter-comm. Hop	
		Range	Quartile	Range	Quartile
Flunet	Oracle	0-6	1,1,1	1-4	1,2,2
	Bubble	2-11	2,3,6	2-7	2,3,4
	HCBF	1-9	1,3,4	1-5	1,2,3
Sassy	Oracle	0-3	1,1,1	1-4	1,2,3
	Bubble	1-11	2,4,7	2-9	3,4,6
	HCBF	1-8	2,3,5	1-6	3,4,5
SHED1	Oracle	0-4	1,1,1	1-3	1,1,2
	Bubble	0-11	2,3,6	2-9	3,4,5
	HCBF	0-7	2,3,4	1-6	2,2,3

C. Oracles

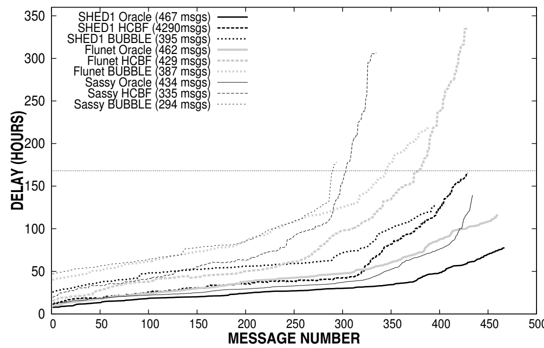
Further analysis of HCBF and BUBBLE's performance is done by comparing them with the oracles. We define the Fastest Oracle as the latency of the first packet copy to arrive at the destination using ER [10]. The Minimum Cost oracle is the minimum number of hops to reach the destination using ER before TTL expiry.

Figure 6 shows the distribution of delay for BUBBLE and HCBF along with the Fastest Oracle. A sample run is shown, but most of the runs are similar in shape. In all 3 scenarios, when messages are sorted by delivery latency, HCBF outperforms BUBBLE on every message that both can deliver. In the experiments with CBF, there were cases where a message of the same delivery rank (i.e. n^{th} fastest) had a lower delay with BUBBLE, but the *average* latency was less with CBF for the number of messages that both could deliver. It is very difficult to determine if specific messages had different relative delivery latencies, but one could imagine that the heuristics in both CBF and HCBF improve delivery latency for some messages and degrade it for others. For the fastest 50% of the messages, it appears there is a constant difference in delay between BUBBLE and HCBF as the corresponding lines appear close to parallel.

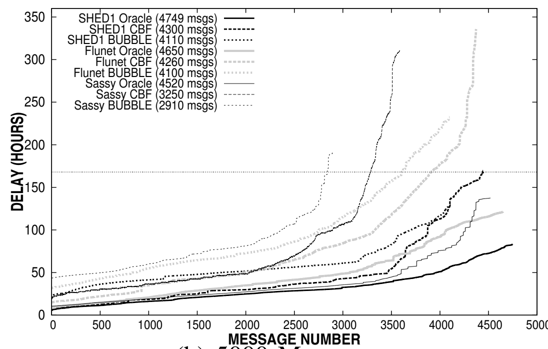
With 500 messages, most of the the message experience a consistent delay (50-60 hours in BUBBLE, and 30-40 hours in HCBF) for SHED1. As the number of messages increases, the delay distribution decreases in all datasets.

With 5000 messages, HCBF for Flunet delivers many more messages with large latencies that otherwise could not be delivered at all with BUBBLE. This indicates that buffers can be emptied with some messages getting delivered very quickly by HCBF to keep room for the hard-to-deliver messages that do eventually find the destination.

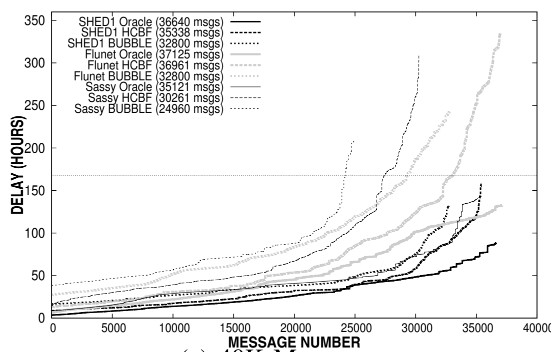
It is interesting to note that HCBF approaches the Oracle for the first 30000 messages in the 40000 message case. After



(a) 500 Messages



(b) 5000 Messages



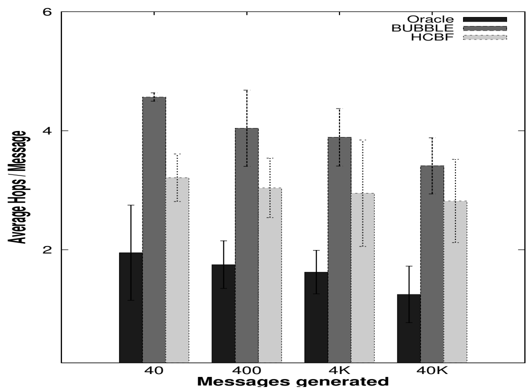
(c) 40K Messages

Fig. 6: **Latency Distributions**

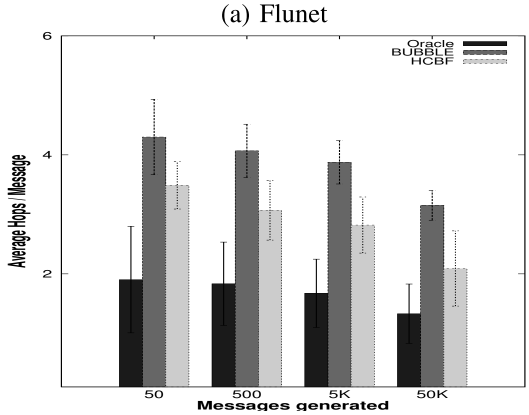
that, the delay of HCBF increases sharply. This suggests that HCBF often chooses the best possible node for forwarding, but when it makes a bad choice, either earlier or later in the path, the latency increases significantly.

In general, the more stable the communities in the dataset, the closer the delay distribution is to that of the Oracle for at least 60% of the messages delivered, but occasionally sub-optimal choices are made that extend delivery. HCBF is conservative, but enough appropriate nodes are selected for forwarding to reduce the average and individual latencies below those of BUBBLE.

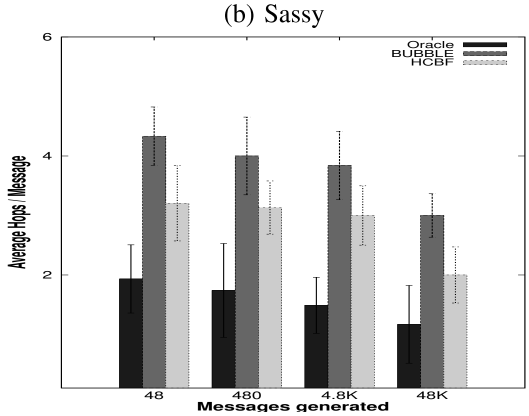
The minimum cost Oracle is used to compare the number of transmissions made by each algorithm for successfully delivered messages. Figure 7 shows that the number of hops is reduced in HCBF, and as more messages are generated, the number of hops required in HCBF approaches that of the Oracle. On average, HCBF only takes one additional hop above the oracle, which is impressive for simple heuristics on stochastic systems.



(a) Flunet



(b) Sassy



(c) SHED1

Fig. 7: **Average hops: Minimum Cost Oracle**

VI. Discussion

While substantial routing performance gains can be made by intelligently partitioning the dynamic graph into communities, and passing messages from community to community, most of the individual message passes happen within a community. Leveraging recent work in characterizing human contact patterns, we defined a new metric based on contact diversity: Unique Interactions. We were able to leverage UI through simple heuristics to address performance shortcomings, and provide equal to or better than benchmark algorithm performance across all measured metrics.

A. Contributions

Metric: We identified a new metric, Unique Interactions, and leveraged it to provide increased performance by improving intra-community routing.

Measurement: We validated our modified algorithm using the new metric against real contact data and oracles, providing empirical evidence for the existence improvement and the possible scope of future improvements.

Modality: By splitting the analysis into intra and inter community routing we were able to establish the degree to which each contributed to overall performance.

B. 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. Validating this work with other data or simulated contact patterns could indicate the degree to which our findings generalize. We have adopted a simple, stylized message-passing model, consistent with other work. Our findings could be strengthened with load characteristics drawn from a specific message-passing problem. Finally, we have only tested in simulation, and have neglected issues such as packet drop or collisions. More complete transmission models might indicate additional areas of strength and weakness for the approach.

VII. 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. 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 HCBF to preferentially choose nodes with a greater probability of being in the target community, and intra-community nodes more likely to get the message to those bridge nodes. When compared to BUBBLE and CBF, HCBF's transmission performance reaches closer to that of the shortest route oracle and in all cases HCBF transmits fewer messages, making it more energy-efficient. Quantitative analysis shows that more stable networks reap the benefits of HCBF's enhancements to a greater degree. HCBF is an attractive lightweight algorithm for resource limited PSN routing.

References

[1] 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*, Philadelphia, PA, Aug. 2005, pp. 244–251.

[2] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based Forwarding in Delay Tolerant Networks," in *ACM Mobihoc*, Hong Kong, China, May 2008, pp. 241–250.

[3] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *ACM SIGCOMM*, Portland, OR, Sep. 2004, pp. 145–158.

[4] L. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, "Classes of Small-world Networks," *Proceedings of the National Academy of Science USA*, vol. 97, pp. 11 149–11 152, 2000.

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

[6] 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*, Miami, FL, Jan. 2012, pp. 255–264.

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

[8] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," *SIGMOBILE Mobile Computing and Communication Review*, vol. 7, no. 3, pp. 19–20, 2004.

[9] V. Pulimi, T. Paul, K. Stanley, and D. Eager, "Near-optimal Routing for Contour Detection in Wireless Sensor Networks," in *IEEE LCN*, Clearwater, FL, Oct. 2012, pp. 462–469.

[10] K. Rasul, S. A. Chowdhury, D. Makaroff, and K. Stanley, "Community-Based Forwarding for Low-Capacity Pocket Switched Networks," in *ACM MSWiM 2014*, Montreal, Canada, Sep. 2014, pp. 249–257.

[11] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *ACM Mobihoc*, Montreal, Canada, Sep. 2007, pp. 32–40.

[12] A. Vahdat and D. Becker, "Epidemic Routing for Partially-connected Ad Hoc Networks," Duke University, Tech. Rep. CS-2000-06, Apr. 2000.

[13] Z. J. Haas and T. Small, "A new networking model for biological applications of ad hoc sensor networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 27–40, Feb. 2006.

[14] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, base stations, and meshes: enhancing mobile networks with infrastructure," in *ACM Mobicom*, San Francisco, CA, Sep. 2008, pp. 81–91.

[15] M. Hashemian and K. G. Stanley, "Effective utilization of place as a resource in pocket switched networks," in *IEEE LCN*, Bonn, Germany, Oct. 2011, pp. 247 –250.

[16] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 63 –76, Feb. 2008.

[17] M. Musolesi and C. Mascolo, "CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks," *IEEE Trans. on Mobile Computing*, vol. 8, no. 2, pp. 246–260, 2009.

[18] X. F. Guo and M. C. Chan, "Plankton: An efficient DTN routing algorithm," in *SECON*, New Orleans, LA, Jun. 2013, pp. 550–558.

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

[20] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," in *WWW*, Brisbane, Australia, Apr. 1998, pp. 161–172.

[21] F. Fabbri and R. Verdone, "A Sociability-Based Routing Scheme for Delay-Tolerant Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, pp. 1:1–1:13, 2011.

[22] A. Pietilänen and C. Diot, "Dissemination in Opportunistic Social Networks: The Role of Temporal Communities," in *ACM MobiHoc*, Hilton Head, SC, Jun. 2012, pp. 165–174.

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

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

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

[26] 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*, Hong Kong, China, Sep. 2013, pp. 47–52.

[27] V. Srinivasan, M. Motani, and W. T. Ooi, "Analysis and implications of student contact patterns derived from campus schedules," in *ACM Mobicom*, Los Angeles, CA, Sep. 2006, pp. 86–97.

[28] K. Lee, S. Hong, S. Kim, I. Rhee, and S. Chong, "SLAW: self-similar least-action human walk," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 515–529, Apr. 2012.

[29] K. Rasul, "Community-Based Forwarding for Low-Capacity Pocket Switched Networks," Master's thesis, University of Saskatchewan, 2013.