On Balancing the Energy Consumption of Routing Protocols for Opportunistic Social Networks

Chen Yang, Radu Stoleru
Department of Computer Science and Engineering, Texas A&M University
{yangchen, stoleru}@cse.tamu.edu

***Abstract*—Routing protocols for opportunistic social network (OSN) utilize *popular* nodes to achieve good routing performance with low overhead. This, however, may result in a severe energy consumption imbalance. The existence of Transient Connected Components (TCC) further complicates the problem due to the randomness of TCC's topology. This paper investigates the energy consumption imbalance problem in OSNs with TCCs. We show that TCC-aware routing protocols, despite their superiority in routing performance, suffer from a more severe imbalance problem. We propose an Energy Consumption Balanced Routing protocol, which includes a new metric and a *routing protocol independent* mechanism. We analytically show that our protocol leads to the convergence of aggregate traffic carried by each node. Through simulation on real world traces, we show that our protocol reduces the energy imbalance by up to 31%, while maintaining comparable routing performance, and that our protocol independent mechanism balances the energy consumption of existing OSN routing protocols.**

I. Introduction

Opportunistic Social Networks (OSN) have gained increasing attention in recent years due to the widespread use of mobile cell phones that have powerful computing, sensing and communicating capabilities [1]. In OSNs, human held mobile devices communicate opportunistically due to the lack of end-to-end connectivity. Early research has focused on exploiting pairwise contact to disseminate information, while a recent study [20] has identified the wide existence of *Transient Connected Components* (TCCs) in OSNs and has revealed their positive impact on information dissemination.

Recent OSN routing protocols [2] [6] [9] achieve good delivery rate and delay performance with relatively low routing overhead by exploiting social structure of humans [16], such as identifying and utilizing "high quality" nodes. However, as a node with high quality metric is more likely to *receive a message and becomes a message carrier*, it inevitably carries a high *memory load* [4]. This results in higher buffer utilization, more received and transmitted messages, and consequently, a *higher energy consumption rate*. Using two real world mobility traces [3] [10], we show that the energy consumption imbalance problem indeed exists in social-based routing protocols. Moreover, protocols utilizing TCCs suffer from more severe memory load imbalance. Although utilizing popular nodes is critical for social-based routing protocols, over-utilization of these nodes may result in multiple detrimental effects on routing performance [16].

Existing OSN congestion control mechanisms [13] [8] [14] that are based solely on balancing buffer utilization do not guarantee energy consumption balance as popular nodes deliver messages much faster than typical nodes. This results in higher aggregated traffic flow and thus higher energy consumption. Moreover, even if popular nodes are not selected to carry messages, they still have a higher chance for relaying messages for other nodes within the same TCC when they are on shortest paths. As existing OSN congestion control mechanisms are TCC-unaware, they will suffer from this problem in networks with TCCs.

Due to the presence of TCC in OSNs, accurate analysis of energy consumption (stemming from node transmission/communication overhead) and finding good metrics for balancing energy consumption of routing protocols are extremely difficult. The main ideas we propose are the use of the node *memory load*, and the need for *energy-aware intra-TCC routing*. We propose a novel Memory Load-aware routing metric that combines both routing quality metric and memory load metric; and Energy-aware Intra-TCC routing to avoid hot zones within TCCs. Based on these, we propose our Energy Consumption Balanced Routing protocol that reduces energy consumption imbalance while maintaining routing performance. We analyze our protocol and show that after sufficient time, a node's average memory load rate converges, and implicitly the average energy consumption rate converges due to the demonstrated correlation between two of them. We evaluate our Energy Consumption Balanced Routing protocol using the Reality and UCSD mobility trace. Simulation results show that our protocol can significantly improve both energy balance and memory load balance while achieving comparable routing performance (PDR, PDD and data copy overhead) with existing social-based routing protocols. The contributions of this paper are as follows: (a) it analyzes the energy consumption imbalance in OSNs with TCCs and demonstrates that TCC-aware routing protocols suffer from a more severe memory load imbalance; (b) it proposes an Energy Consumption Balanced Routing protocol which provably guarantees the convergence of average memory load rate; (c) using real world mobility traces, it shows the effectiveness of proposed solutions when compared with state-of-art routing protocols.

II. Preliminaries

A. Transient Connected Component (TCC)

It is identified in [20] that, in addition to pairwise contacts, *Transient Connected Components* (TCCs) widely exist in OSNs. TCCs are temporally formed in OSNs where mobile

978-1-4673-8590-9/15/$31.00 ©2015 IEEE

devices are able to communicate in a multi-hop manner. Although the existence of TCCs increases the contact probability between nodes, we show that TCC-aware routing protocols may suffer from a more severe memory load imbalance problem, when compared to TCC-unaware routing protocols.

B. Social-based Routing Protocols

Most social-based routing protocols use different *quality metrics* to measure the contact capability of a given node. When a node encounters a new neighbor, these quality metrics are used by *forwarding strategies* to decide whether to forward a message or not. *Message carriers* are the nodes who hold the message for further forwarding (when TCC topology changes occur) or delivering to destinations. A node with high quality metric is therefore assumed to be a better *message carrier* than a node with lower quality metric.

1) Quality Metrics: The proposed quality metrics can be classified into two categories: *destination-independent* and *destination-dependent*. A destination-independent metric represents the contact capability of a node regardless of the destination of the message. This type of metric includes **Betweenness** [2] [6] and **CCP** [5]. On the other hand, a node may have different quality metric values, depending on different destinations. For example, **Dest. Frequency** [4] and **Contact Probability** [9] are destination-dependent. We briefly introduce **Betweenness** and **Dest. Frequency** which are used in this paper.

Betweenness: Consider a *contact graph* $G = (V, E)$ where an edge linking v_i and v_j means that v_i and v_j have met each other. The Betweenness c_i of node v_i is defined as: $c_i = \sum_j \sum_k \frac{g_{jk}(v_i)}{g_{jk}}$, where g_{jk} represents the total number of shortest paths from v_j to v_k, and $g_{jk}(v_i)$ is the number of shortest paths that contains node v_i.

Dest. Frequency: The destination frequency c_{ij} of node v_i to destination v_j is defined as $c_{ij} = \lambda_{ij}$, where λ_{ij} is the contact rate between node v_i and v_j.

2) Forwarding Strategies: We assume that the node who initiates the forwarding still keeps the message. We adopt the following forwarding strategies as they are widely used:

Compare and Forward: The node forwards the message to the encountered node if the latter one has a higher quality metric. When multiple nodes contact, the node selects the one with the highest quality metric. This strategy is widely adopted by previous research [6] [2] [9].

Delegation Forwarding: The node forwards the message to the encountered node if the latter one has the highest quality metric among all nodes it previously met. This strategy was proposed in [4] to further reduce the number of data copies.

3) Routing Protocols: The combination of a quality metric and a forwarding strategy results in a routing protocol. If the protocol only selects *message carriers* from its one hop neighbors, we say it is a TCC-unaware routing protocol. If the protocol selects *message carriers* from all nodes within a TCC, we say it is a TCC-aware routing protocol. An example is shown in Figure 1, where the quality metric is shown beside each node. In this example, if v_1 uses a TCC-unaware protocol, it chooses the *next carrier* from the set $\{v_2, v_3\}$ and hence v_3 will be chosen. If it uses a TCC-aware protocol, it chooses from the set $\{v_2, \ldots, v_6\}$ and hence v_6 will be chosen. Notice that when v_6 is selected as the *next carrier*, intermediate node v_3 only *relays* the message to v_6 without adding it to its own buffer, i.e., v_3 will not become a message carrier. When the selected *next carrier* is multiple hops away, we assume that the node which initiates the forwarding calculates the shortest path and performs source routing. In the rest of the paper, we reserve the term "forwarding" for OSN forwarding, while using "relaying" for intra-TCC forwarding.

Trace	*MIT Reality*	*UCSD*
Device	Phone	PDA
Network type	Bluetooth	WiFi
No. devices	97	275
Duration (days)	246	77

TABLE I: Mobility traces [3] [10]

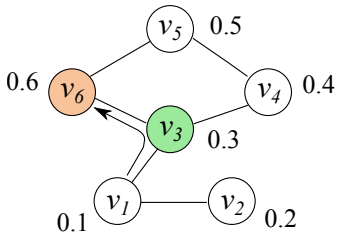


Fig. 1: TCC-unaware v.s. TCC-aware routing protocols.

C. Datasets

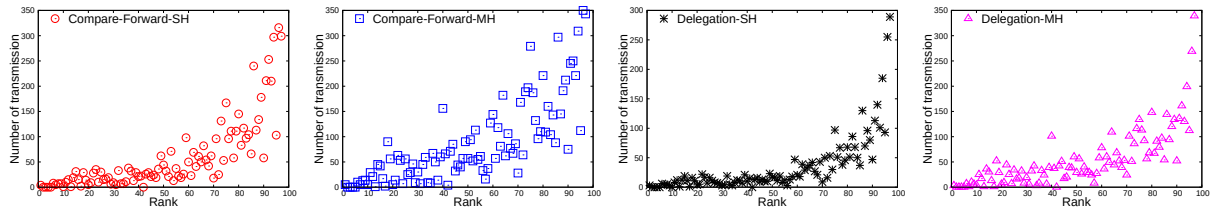
In this paper we use two mobility traces for both analysis and performance evaluation. The details are shown in Table I. As shown, the *MIT Reality* [3] trace consists of 97 users carrying cell phones. Users are selected from MIT students, staff and faculty members. The mobility trace contains Bluetooth sightings between devices over the course of nine months. The *UCSD* [10] have recorded the Access Point detection and association events for 275 hand-held PDAs distributed to students. The trace covers eleven weeks. We assume that there is a contact event between two users when the time windows in which they detect the same AP overlap.

III. Motivation And Memory Load Analysis

To motivate our research, we perform simulations for existing social-based routing protocols on two real world mobility traces. These empirical results demonstrate that a severe energy consumption imbalance indeed exists. As mentioned in Section 1, we argue that memory load [4] is a key indicator of energy consumption. We therefore theoretically analyze memory load, as an accurate analysis of energy consumption is difficult due to the presence of TCC (note that in a TCC, different routes exist between any pair of nodes). Our analysis shows that memory load is also severely imbalanced, and that routing protocols that utilize TCC suffer from more severe imbalance. We then validate our analytical results using simulations on real world traces.

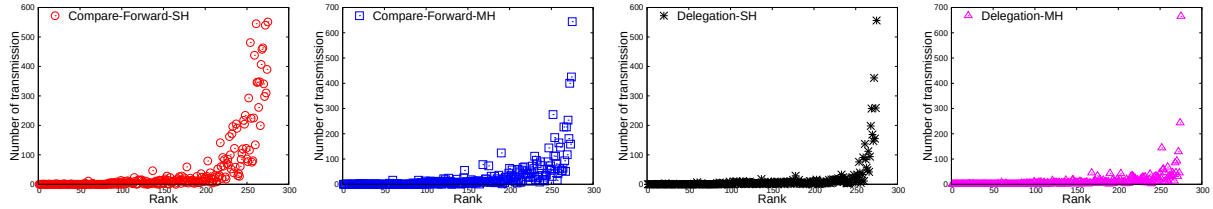
A. Motivation

Transmission and reception of messages is the main source for energy consumption in OSN/DTN routing protocols [17].



(a) Comp.-Forward (b) Comp.-Forward-TCC (c) Delegation (d) Delegation-TCC

Fig. 2: Energy consumption imbalance for two forwarding strategies in the Reality trace: TCC-unaware (a,c), TCC-aware (b,d).



(a) Comp.-Forward (b) Comp.-Forward-TCC (c) Delegation (d) Delegation-TCC

Fig. 3: Energy consumption imbalance for two forwarding strategies in the UCSD trace: TCC-unaware (a,c), TCC-aware (b,d).

The transmission and reception opportunity in an OSN with TCC mainly occurs in the following three scenarios: (a) a node in a TCC has a message and decides to forward that message to another node in the TCC (i.e., the next carrier or the destination); (b) a node in the TCC relays a message (initiated by a node, as explained in (a)) on a path, towards the next carrier (as decided by the initiating node) or the destination; (c) a node is selected as a message carrier or is the destination and thus receives a message. In case (a) nodes with higher quality metric have a higher chance for forwarding messages to destinations (since they receive more messages) but a lower chance for forwarding messages to the next carrier (since the number of nodes with even higher metric is small). In case (b) a node with higher quality metric has a lower chance to serve as an intermediate node, since it has a lower chance to meet other nodes that have higher quality metric than itself. On the other hand, a node with a lower quality metric has a higher chance to serve as an intermediate node when it is in a TCC. But since the node has a low quality metric, it does not meet other nodes as often as nodes with high quality metric (if the metric reflects the contact rate). Furthermore, whether a node is on a forwarding path in the TCC highly depends on the network topology, which is even harder to analyze or predict. In case (c) nodes with higher quality metric have higher chance to be selected as message carriers. Given the complexity of analyzing the transmission imbalance, we perform simulations to understand the energy consumption imbalance.

We perform simulations using two mobility traces, Reality and UCSD. **Betweenness** is used as the quality metric. We pre-process the trace: we calculate each node's Betweenness metric and use it to rank nodes. 1,000 messages are generated with source and destination randomly chosen among all nodes within each trace. The deadline of each message is set to 24 hours. Once the message is delivered to the destination, it is eliminated from all nodes' buffers. We assume infinite communication bandwidth and buffer size. For each node, we record the total number of packet transmissions and receptions. We plot the *(rank, value)* pair for each node in Figure 2 and Figure 3. We use "-TCC" to indicate TCC-aware routing.

From Figure 2 and Figure 3, we can observe that the energy consumption imbalance exists. For TCC-unaware protocols, as shown in Figures 2a, 2c and Figures 3a, 3c, nodes with the top 10% Betweenness contribute up to 42% (74%) of the total energy consumption in the Reality (UCSD) trace. Although nodes within TCCs may serve as intermediate nodes and relay messages, the energy consumption imbalance for TCC-aware routing is still severe in spite of the added randomness. As shown in Figures 2b, 3b and Figures 2d, 3d, the top 10% nodes contribute up to 32% (57%) of the total energy consumption in the Reality (UCSD) trace. These results motivate our research for reducing the energy consumption imbalance.

We propose that a key indicator of energy consumption is memory load. Next, we present our theoretical analysis and empirical results for memory load imbalance, and demonstrate its correlation with energy consumption.

B. Memory Load Imbalance Analysis

1) Network Model and Assumptions: Consider a network with a set of N nodes. Each node v_i has a quality metric c_i. We model the contact process of the entire network as a homogeneous Poisson process $\{T_n\}_{n\geq 0}$ [12], where $\{T_n - T_{n-1}\}_{n\geq 1}$ are i.i.d exponential variables with mean $1/\lambda$. During each contact event at time T_n, a set of node V_n meet with each other, where $|V_n| = S_n$. Each node v_i generates messages according to a renewal process $\{M_n^i\}_{n\geq 0}$, independent of the contact process, with mean inter-arrival time $1/\mu_i$, and message size $\{D_n^i\}_{n\geq 0}$. Each message has a deadline L, i.e. the n-th message is expired after $M_n^i + L$. We consider Compare and Forward as the forwarding strategy since it is the simplest and commonly used strategy. During each contact, only the node with the highest metric may receive a copy of

the message.

We make the following assumptions to facilitate our analysis. We assume that c_i are distributed in $(0, 1]$ interval [4]. Notice that it is usually the relative order of metrics between two nodes that decides the message forwarding. So this assumption does not affect the analysis of load imbalance. Without loss of generality, let c_i be the i-th least metric, i.e. $c_1 < \cdots < c_N$. The duration of each contact is assumed to be negligible but sufficient for all data transfer, which is a common assumption in many DTN analysis [12]. S_n is shown to follow a geometric distribution [20]. We can therefore vary $E[S_n]$ to analyze TCC-aware and TCC-unaware routing protocols. We assume that the data sizes $\{D_n^i\}_{n\geq 0}$ are i.i.d random variables and are independent of both $\{M_n^i\}_{n\geq 0}$ and $\{T_n\}_{n\geq 0}$.

2) Memory Load Analysis: We model the memory load of a node v_j incurred by messages from v_i as a renewal reward process. For the n-th message from v_i, the "reward" $R_{i,j}^n$ for v_j is a random variable $\mathbb{1}_{\delta_{i,j}^n} D_n^i$, where $\mathbb{1}_{\delta_{i,j}^n}$ indicates whether v_j receives a copy of the message before it expires. The total reward, i.e. the memory load, from v_i at time t is therefore $R_{i,j}(t) = \sum_{n=0}^{N_i(t)-1} R_{i,j}^n$, where $N_i(t) = \max\{n : M_n^i < t\}$ is the number of generated messages before t. The total memory load of node v_j is $R_j(t) = \sum_{i:i<j} R_{i,j}(t)$. By Elementary Renewal Theorem [15], we know that $\lim_{t\to\infty} E[R_{i,j}(t)]/t = \mu_i E[R_{i,j}^n] = \mu_i P(\delta_{i,j}^n) E[D_n^i]$. Therefore, the time average of the memory load of v_j is

$$\bar{R_j} = \lim_{t\to\infty} \frac{1}{t} E[R_j(t)] = \sum_{i:i<j} \mu_i P(\delta_{i,j}^n) E[D_n^i] \qquad (1)$$

Now we compute the probability of event $\delta_{i,j}^n$. Notice that since $\{T_n\}_{n\geq 0}$ has intensity λ, the number of contacts N_C during a message's lifetime L is a Poisson random variable with mean λL. Therefore, $P(\delta_{i,j}^n | N_C) = 1 - \prod_{k=1}^{N_C}(1 - P(\delta_{i,j,k}^n))$, where $\delta_{i,j,k}^n$ denotes the event of v_j *receives a copy of* v_i*'s* n*-th message during the* k*-th contact (with set* $V_{(k)}$*) after the message generation*. Notice that $\delta_{i,j,k}^n$ occurs, if both v_i and v_j are in $V_{(k)}$ and v_j has the highest metric. Assuming that each node v_i has probability p_i to be in the set V_k, then if N is sufficiently large, $P(\delta_{i,j,k}^n | S_{(k)}) = p_i p_j (1 - p_i - \sum_{m\geq j}^{N} p_m)^{S_{(k)}-2}$, for $S_{(k)} \geq 2$. We consider the case where the probability p_i is proportional to the node's quality metric, i.e. $p_i = c_i / \sum_l c_l$. This corresponds to many quality metrics where the node with a high total contact rate has a high quality metric [4]. $P(\delta_{i,j,k}^n)$ and $P(\delta_{i,j}^n)$ can then be calculated using total probability formula given the distribution of N_C and $S_{(k)}$, i.e. $P(\delta_{i,j,k}^n) = \sum_{l=2}^{\infty} Pr(S_{(k)} = l) p_i p_j (1 - p_i - \sum_{m\geq j}^{N} p_m)^{l-2}$, $P(\delta_{i,j}^n) = \sum_{l=0}^{\infty} Pr(N_C = l)(1 - \prod_{k=1}^{l}(1 - P(\delta_{i,j,k}^n)))$. Equation 1 can be evaluated when μ_i and $E[D_n^i]$ are known.

As a case study, let us assume that all nodes generate messages at the same rate and with the same expected data size, i.e. $\mu_i = 1$, $E[D_n^i] = 1$ for all i. Node v_i has a quality metric $c_i = i/N$, for $i = 1, \ldots, N$. Suppose λ is 10 times μ_i, i.e., $\lambda = 10$; and that L is 24 times inter-message event time, i.e., $L = 24$. We plot the time average of memory load $\bar{R_j}$ for each node v_j in Figure 4. It is clear that the memory load imbalance exists. Nodes with high quality metric incur much higher memory load than nodes with low quality metric. Moreover, and more importantly, we notice that the imbalance is increased with the increase of $E[S_k]$, i.e. the mean size of the encounter node set. *Since in our analysis a larger encounter node set mimics TCC-aware routing behavior, we argue that TCC-aware routing protocol results in more severe memory load imbalance*.

	CnF-SH	CnF-MH	Del-SH	Del-MH
Reality	0.95	0.79	0.96	0.81
UCSD	0.94	0.90	0.96	0.94

TABLE II: r between energy consumption and memory load

3) Empirical Results: We run simulations using parameters introduced previously. In addition to Betweenness, we also use **Dest. Freq.** as quality metric. We record the value of memory load and plot $(rank, value)$ pairs. For Dest. Frequency, we use average rank to represent each node.

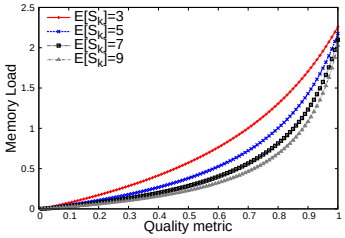


Fig. 4: Memory imbalance analysis

Results for Betweenness are shown in Figure 5a and Figure 5b. We can see that the memory load increases with the node's Betweenness metric, and that the trend fits our analytical results. The node with high Betweenness metric receives many more messages, when compared to typical nodes. Furthermore, when comparing the same forwarding strategy with different TCC-awareness, we can see that *TCC-aware routing makes the memory load imbalance more severe*. This again fits our analysis in the previous section.

Table II presents the Pearson correlation coefficient r between energy consumption and memory load. It is clear that they are positively correlated. We therefore consider memory load as an indicator for energy consumption.

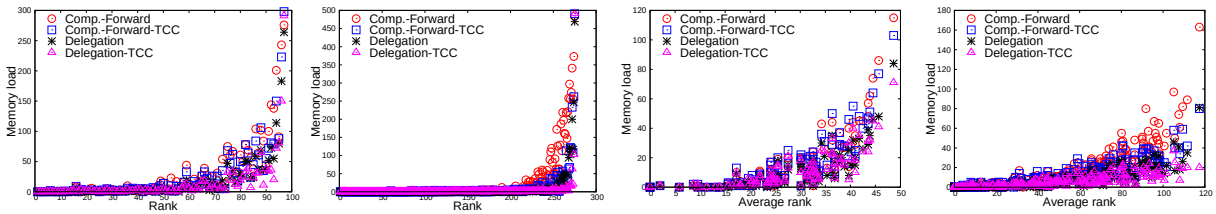
In Figure 5c and Figure 5d we can see that when using destination-dependent metric, nodes with high average contact rates receive many more messages. Since destination-dependent metrics have similar imbalance performance, we focus on destination-independent metrics in the rest of this paper.

IV. Energy Consumption Balanced Routing In Opportunistic Social Networks

In this section, we present our Energy Consumption Balanced Routing protocol for OSN with TCCs. We consider routing a single message m from source v_s to destination v_d. The message has a generation time t_m and a deadline T, i.e. the message expires after time $t_m + T$.

A. Main Idea

Our main ideas are to give higher priority to under-utilized nodes when selecting message carriers and to perform intra-TCC routing in an energy balanced manner. We jointly



(a) Betweenness, Reality (b) Betweenness, UCSD (c) Dest. Freq, Reality (d) Dest. Freq, UCSD

Fig. 5: Memory load imbalance for two quality metrics in the MIT Reality trace (a,c) and the UCSD trace (b,d)

consider both *quality* metric and *memory load quota* metric when selecting message carriers, as we demonstrated in the previous section that memory load is a key indicator for energy consumption in OSN. The quality metric reflects the capability of a node to deliver the message within a short period of time. Always selecting high quality metric nodes as message carriers can maximize the PDR but results in severe energy consumption imbalance. We propose a *memory load quota* metric to indicate if the node is over-utilized or under-utilized. If a node is already over-utilized, we avoid selecting it as message carrier in the near future. We propose a Memory Load-aware routing metric by combining both metrics, which is then used as input to the TCC-aware Comp. and Forward strategy. We also propose to use *Energy-aware Intra-TCC routing* to route messages within a TCC. As popular nodes have a greater chance to lie on shortest paths between other nodes in a TCC, it is necessary to avoid selecting paths that contain over-utilized nodes when routing within a TCC. We present the details of our protocol in the following sections.

B. Memory Load-Aware Routing Metric

We associate with each node v_i a *memory load quota* q_i. At time $t = 0$, q_i are set to q_{init} for all v_i. When v_i receives a message and adds the message to its buffer, q_i is updated by $q_i \leftarrow q_i - q_{cost}$ where q_{cost} is a constant. We denote the *memory load quota* of v_i at time t as $q_i(t)$. Notice that $q_i(t)$ reflects the utilization of node v_i until moment t. If node v_i has a significantly lower $q_i(t)$ when compared to other nodes, we consider that it is over-utilized.

Now we define the Memory Load-aware routing metric.

Definition 1. *The Memory Load-aware routing metric $r_i(t)$ for node v_i at time t is: $r_i(t) = c_i + \rho \cdot q_i(t)$, where $\boldsymbol{c_i}$ is node $\boldsymbol{v_i}$'s quality metric, $\rho > 0$ is a control variable.*

When routing a specific message m, we use a *message-specific Memory Load-aware routing* metric, defined by

$$r_{i,m} = r_i(t_m) = c_i + \rho \cdot q_i(t_m)$$

where t_m is the generation time of message m.

Notice that $r_{i,m}$ is an evaluation of $r_i(t)$ at moment t_m. We choose to use a message-specific metric to reduce the number of data copies. In fact, if we evaluate the metric at the exact moment when making forwarding decisions, unnecessary data copies may be created to balance the memory load among nodes. We show through simulations that our protocol has comparable data copy overhead with other protocols.

Algorithm 1 BetRank Calculation For v_i.

```
1:  Upon receiving timestamped node metric list L
2:  Update myBet
3:  k ← 0
4:  for all entry in L do
5:      myEntry ← myList.get(entry.nodeID)
6:      if myEntry.t < entry.t then
7:          myEntry.bet = entry.bet
8:          myEntry.t = entry.t
9:  for all entry in myList do
10:     if entry.bet < myBet then
11:         k ← k + 1
12: myBetRank = k/N
```

In order to keep track of q_i, each node has to maintain a *memory load quota table*, which includes a series of $(t, q_i(t))$ pairs. Entries are purged if they are older than a certain threshold, which we set to the message's deadline T.

C. BetRank Metric

In this paper we propose a new node quality metric called *BetRank*.

Definition 2. *BetRank c_i for node v_i: Suppose node v_i has the k-th least Betweenness among N nodes, then its BetRank is defined as $c_i = k/N$.*

Notice that on one hand BetRank preserves the relative order of Betweenness metric between different nodes. Thus BetRank and Betweenness will have identical routing performance. On the other hand, BetRank enforces node's quality metric to be uniformly distributed in $(0, 1]$. We will show, through analysis, that when Memory Load-aware routing metric uses BetRank as node quality metric, the memory load consumption rate for different nodes converges.

In OSNs, the Betweenness metric can be estimated through the Ego network betweenness [2]. Here, we introduce a simple algorithm to accurately estimate the rank of each node's metric, as shown in Algorithm 1. Each node maintains a list of timestamped Betweenness values of all nodes, including itself, and periodically disseminates this information to its neighbors. By updating this information (line 4 to 8), each node counts the number of nodes that have smaller Betweenness metric than itself (line 9 to 11), which leads to an estimation of rank. A tie can be broken by the node ID. Due to space limitation, the evaluation of BetRank is in [19].

Algorithm 2 Energy Consumption Balanced Forwarding Algorithm For v_i.

1: **for all** message m in v_i's message queue **do**
2: **if** $m.Dest \in TCC$ **then**
3: $m.Path \leftarrow EnergyAwareIntraTCC(v_i, m.Dest)$
4: $v_i.Send(m)$
5: **else**
6: Find $r_{j,m} = max\{r_{k,m}, v_k \in TCC\}$
7: **if** $r_{j,m} > r_{i,m}$ and $m \notin v_j$'s msg. queue **then**
8: $m.Path \leftarrow EnergyAwareIntraTCC(v_i, v_j)$
9: $v_i.Send(m)$

D. Energy-aware Intra-TCC Routing

Although controlling memory load is key to energy consumption balance, it is not sufficient for a TCC-aware routing protocol. In a TCC-aware routing protocol, a node that initiates forwarding has to find a path to the next message carrier. Controlling memory load keeps popular nodes from constantly being selected as message carriers. However, popular nodes that have good connectivity may always lie on the shortest path between other nodes within TCC. Energy imbalance may also incur, as popular nodes have to relay more messages.

To address this problem, we propose an *Energy-aware Intra-TCC routing*. We represent a TCC as a graph $G = (V, E)$, where V is the set of nodes, and E is the set of edges. Suppose that each node v_i has consumed energy $e_c(i)$. We associate with each edge $(v_i, v_j) \in E$ a weight $w_{ij} = \frac{1}{2}(e_c(i) + e_c(j))$. Then the shortest path algorithm applied to the weighted graph prefers nodes with high remaining energy, and thus avoids nodes that have already been over-utilized.

E. Energy Consumption Balanced Routing Protocol

Here we present our Energy Consumption Balanced Routing protocol, which mainly consists of the following three parts.

1) Beacon Exchange: In addition to neighbor discovery, each node broadcasts a *beacon* periodically. Each *beacon* includes a timestamped metric list, consumed energy, neighbor list, message list and a *memory load quota table*. By collecting other node's beacons, each node is able to maintain the topology of the current TCC and learn other's local information.

2) Message Forwarding: When a node v_i detects a TCC topology change, it invokes the forwarding algorithm presented in Algorithm 2. The node first checks if a given message can be delivered to the destination (line 2 to 4). If not, it then finds the node with the maximum Memory Load-aware routing metric in the current TCC (line 6). If the found node has higher metric than itself and does not have the message, it forwards the message to the found node (line 7 to 9). If the selected message carrier or the destination is multiple hops away in the same TCC, the node performs *Energy-aware Intra-TCC routing* as presented in Section IV-D.

3) Message Receiving: When a node v_i receives a message, then: (a) if it is the final destination of the message, it simply adds the message to its received message queue; (b) if it is the intermediate destination (i.e. a selected message carrier) of the message, the node adds the message to its local buffer, updates q_i, and adds a new entry $(t, q_i(t))$ to its *memory load quota table*; (c) if it is the intermediate hop, the node extracts the next hop from the message and transmits the message.

F. Routing Protocol Analysis

In this section we analyze our Energy Consumption Balanced Routing protocol. We show that our protocol leads to the convergence of the memory load quota consumption rate for nodes with different quality metrics.

Since we assume that all nodes have initial memory load quota $q_i(0) = q_{init}$, then $r_i(0) = c_i + \rho \cdot q_{init}$. $r_i(0)$ and c_i have the same rank for v_i. However, since the memory load quota consumption rates (which we call quota consumption rate) are different, $r_i(t)$ has different ranks at time $t > 0$. We denote the rank of $r_i(t)$ among N nodes as $\phi_i(t)$.

We notice that given the same message generation rate, memory load is mainly decided by the rank of the metrics. Therefore, if a node's metric c has rank x, its memory load is a function of x, which we denote as $f_r(x)$. The quota consumption rate $g(x)$ is then proportional to the memory load, i.e. $g(x) = \alpha \cdot f_r(x)$, where $\alpha > 0$ is a constant. Since $f_r(x)$ is an increasing function, $g(x)$ is also an increasing function. Then, the quota consumption rate $\sigma_i(t)$ for node v_i at time t is $\sigma_i(t) = g(\phi_i(t)) = \alpha \cdot f_r(\phi_i(t))$ and the average quota consumption rate $\overline{\sigma}(t)$ for all nodes at time t is:

$$\overline{\sigma}(t) = \frac{1}{N}\sum_{i=1}^{N}\sigma_i(t) = \frac{1}{N}\sum_{i=1}^{N}g(\phi_i(t)) = \frac{1}{N}\sum_{i=1}^{N}g(i) = \overline{\sigma} \tag{2}$$

where the third equality is due to $\{\phi_i(t), i = 1, \ldots, N\} = \{1, 2, \ldots, N\}$, and $\overline{\sigma}$ is a constant.

Now we consider the average quota consumption rate for a given node v_i, defined as $\overline{\sigma_i}(t) = \frac{1}{t}\int_0^t \sigma_i(\tau)\mathrm{d}\tau$. We will show that $\overline{\sigma_i}(t)$ converges to average quota consumption rate (i.e., $\overline{\sigma}$) for all nodes.

The Memory Load-aware routing metric for node v_i is $r_i(t) = c_i + \rho \cdot q_{init} - \rho \cdot \int_0^t \sigma_i(\tau)\mathrm{d}\tau$. Notice that the change of rank among different $r_i(t)$ depends on the third term in the above equation, as the second term is the same for all nodes.

Since we assume that c_i are uniformly distributed in $(0, 1]$, after time $t = 0$ the first change of rank only occurs between the nodes v_{N-1} and v_N as $r_N(t)$ decreases the fastest (with rate $-\rho \cdot g(N)$). Therefore, we first consider node v_{N-1} and v_N at time $t = 0$. At time $t = 0$, node v_{N-1} and v_N's metric $r_{N-1}(0)$ and $r_N(0)$ have the rank $N-1$ and N, respectively. Consider a small time interval $[0, \delta t)$, during which the rank is not changed. Then we have:

$$r_{N-1}(\delta t) = c_{N-1} + \rho \cdot q_{init} - \rho \cdot g(N-1) \cdot \delta t$$
$$r_N(\delta t) = c_N + \rho \cdot q_{init} - \rho \cdot g(N) \cdot \delta t$$

Since $r_N(\delta t)$ is decreasing at a higher rate, at time δt_1, we have $r_{N-1}(\delta t_1) = r_N(\delta t_1)$, where δt_1 is given by $\delta t_1 = \frac{c_N - c_{N-1}}{\rho \cdot (g(N) - g(N-1))}$. After δt_1, node v_{N-1} and v_N can only have the same quota consumption rate: $\frac{1}{2} \cdot (g(N-1) + g(N))$.

Then since $r_{N-1}(t)$ and $r_N(t)$ are decreasing together at a rate higher than $r_{N-2}(t)$, we have $r_{N-2}(t) = r_{N-1}(t) =$

$r_N(t)$ at time δt_2, which is given by:

$$\delta t_2 = \frac{\frac{1}{2}(c_N + c_{N-1}) - c_{N-2}}{\rho \cdot (\frac{1}{2}(g(N) + g(N-1)) - g(N-2))}$$

If we denote $\tilde{c}_k = \frac{1}{N-k+1}\sum_{i=k}^{N} c_i$ and $\tilde{g}(k) = \frac{1}{N-k+1}\sum_{i=k}^{N} g(i)$ for $k = 1, \ldots, N-1$, while $\tilde{c}_N = c_N$ and $\tilde{g}(N) = g(N)$. By observing the pattern, we can write the expression for δt_{N-i} for node v_i as:

$$\delta t_{N-i} = \frac{c_{i+1} - c_i}{\rho \cdot (\tilde{g}(i+1) - g(i))},\ i = 1, \ldots, N-1$$

Therefore, node v_i has quota consumption rate $\sigma_i(t) = g(i)$ in time interval $[0, \delta t_{N-i}]$, and $\sigma_i(t) = \tilde{g}(k)$ in time interval $(\delta t_{N-k}, \delta t_{N-k+1}]$ for $k = i, i-1, \ldots, 2$. After time δt_{N-1}, every node has the same quota consumption rate $\overline{\sigma}$.

Let $\delta t_0 = \delta t_1$, we can write the average quota consumption rate $\overline{\sigma_i}(t)$ for node v_i for sufficiently large t as

$$\frac{1}{t}\left(g(i)\delta t_{N-i} + \sum_{k=2}^{i} \tilde{g}(k)(\delta t_{N-k+1} - \delta t_{N-k}) + \int_{\delta t_{N-1}}^{t} \overline{\sigma} \mathrm{d}\tau\right)$$

for $i = 1, \ldots, N$. As the first two terms are finite, we have

$$\lim_{t\to\infty} \overline{\sigma_i}(t) = \overline{\sigma}$$

This result demonstrates that after sufficient time, all nodes have the same quota consumption rate. *In the next section we will show that the energy consumption imbalance will be reduced, due to the convergence of quota consumption rate and its strong correlation with energy consumption.*

V. Performance Evaluation

We evaluate our Energy Consumption Balanced Routing protocol through simulations using two mobility traces (MIT Reality and UCSD), and a custom trace-driven simulator. For UCSD, we use the first two weeks to reduce simulation time. We assume infinite communication bandwidth and infinite buffer size for each node. 50,000 and 10,000 messages are generated for each simulation run for MIT Reality and UCSD trace, respectively, with the sources and destinations selected randomly from nodes within each trace. Different number of messages are used due to the fact that traces have different duration. The first half of the trace is used as a warmup period, when each protocol estimates the node quality metric they use. We set $q_{init} = 1$, $q_{cost} = 4 \times 10^{-5}$ for Reality and 2×10^{-4} for UCSD. q_{cost} is selected such that all nodes' memory load quota remain positive at the end of simulation.

Our simulator implements a similar energy module as TheONE [7], i.e. we consider both packet transmission and reception. Notice that our analysis in Section III-B has shown that energy consumption is proportional to the expected data size. It is therefore sufficient to consider the number of messages transmitted and received for a node as an indication of its energy consumption [17]. We omit devices' energy consumption for baseline activity and scanning, as we focus on the energy consumption incurred by routing protocols.

We use the message deadline as a simulation parameter, varying from 1 hour up to 24 hours. Since deadline is a typical parameter for evaluating routing performance, we explore its impact on energy consumption imbalance as well. We empirically set $\rho = 15$ for the Reality trace and $\rho = 4$ for the UCSD trace (ρ is selected such that it maintains comparable data copy overhead with other protocols while reduces energy consumption imbalance). Five simulations are run, with random seeds for statistical significance.

We compare our Energy Consumption Balanced Routing protocol with **Compare-Forward**, **TccRoute** [20], **FairRoute** [13], and **Epidemic** routing [18]. TccRoute is a state-of-art TCC-aware routing protocol that uses CCP as node quality metric. FairRoute is a routing protocol that achieves fairness among nodes in OSN. *Aggregated interaction strength* is used as quality metric, and a fairness mechanism based on queue length is proposed to prevent popular nodes from receiving excessive amount of traffic. A node forwards a message only if the encountered node has higher quality metric and smaller queue length. For fairness, we implement a TCC-aware FairRoute because [20] demonstrates that TCC-awareness improves routing performance, and because our protocol is also TCC-aware. Epidemic is essentially a flooding-based protocol, and is typically used as the baseline comparison for routing performance. In this paper, we use it as baseline for energy consumption imbalance, since nodes' opportunity to be message carriers solely depends on mobility.

A. Evaluation Metrics

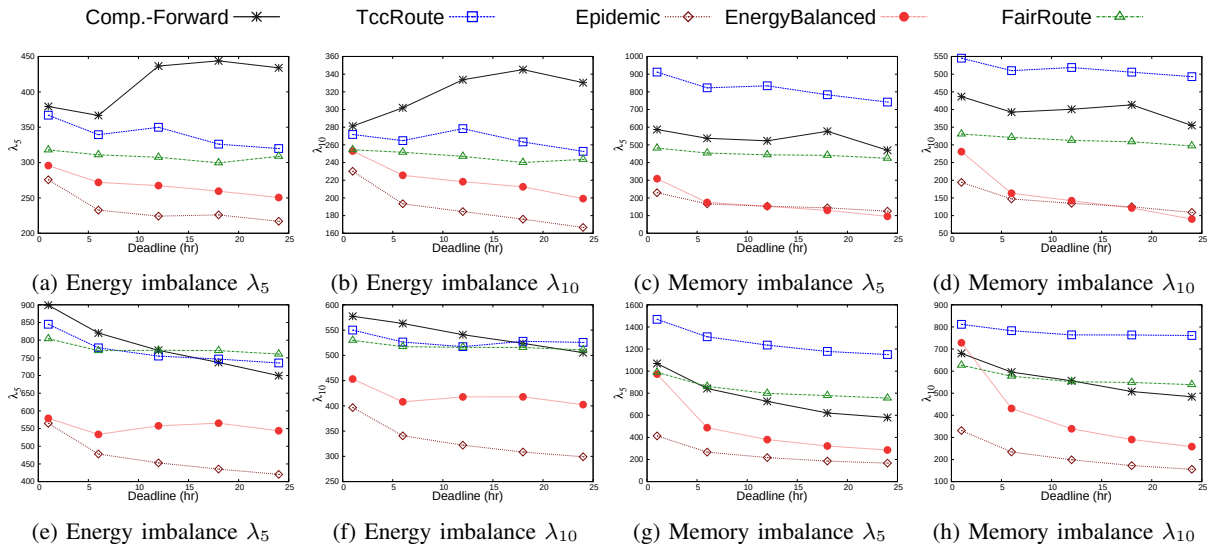
We evaluate the energy consumption imbalance, the memory load imbalance and routing performance. We use L_i to denote either energy consumption or memory load for v_i.

For evaluating energy imbalance, we use the **percent imbalance** metric. This metric is typically used to measure load imbalance in distributed processing systems [11]. Since these two problems are similar, this metric is suitable for both. The metric is defined as: $\lambda_k = \left(\frac{\tilde{L}_k}{\overline{L}} - 1\right) \times 100\%$, where $\tilde{L}_k$ is the average value of top $k\%$ L_i values and $\overline{L} = \frac{1}{N}\sum_i L_i$. *The load imbalance is more severe if the percent imbalance metric is high*. We show the results for $k = 5, 10$ for both energy consumption imbalance and memory load imbalance.

In addition to the percent imbalance metric, we also consider traditional routing performance metrics, including the **Packet Delivery Ratio** (PDR), the **Packet Delivery Delay** (PDD) and **Data Copy** (the number of copies created for each message, indicating the overhead of the routing protocol).

B. Energy Consumption and Memory Load Imbalance Results

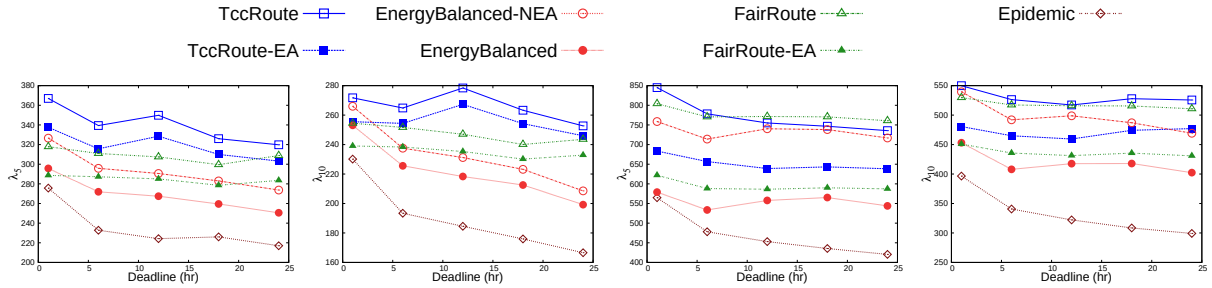
In this section we present the imbalance results, shown in Figure 6 for Reality and UCSD traces. The ultimate goal for our protocol is to achieve better energy consumption balance while maintaining routing performance. From Figure 6 we can see that, as expected, Epidemic achieves the best balance performance among all routing protocols, for both energy consumption and memory load balance. This is due to the fact that Epidemic floods messages to the entire population of the nodes. From Figures 6a, 6b, 6e and 6f we can see that our protocol is the second best in terms of energy consumption balance (only worse than Epidemic) for all deadlines. Compared to the FairRoute protocol, our protocol achieves up to



(a) Energy imbalance λ_5 (b) Energy imbalance λ_{10} (c) Memory imbalance λ_5 (d) Memory imbalance λ_{10}

(e) Energy imbalance λ_5 (f) Energy imbalance λ_{10} (g) Memory imbalance λ_5 (h) Memory imbalance λ_{10}

Fig. 6: Energy consumption and memory load imbalance: (a)-(d) Reality; (e)-(h) UCSD



(a) Energy imbalance λ_5 (b) Energy imbalance λ_{10} (c) Energy imbalance λ_5 (d) Energy imbalance λ_{10}

Fig. 7: Applying Energy-aware intra-TCC routing technique to other routing protocols: (a) (b) Reality, (c) (d) UCSD

19% and 18% reduction of energy consumption imbalance in the Reality trace for λ_5 and λ_{10}, respectively; while in UCSD, our protocol achieves up to 31% and 21% reduction for λ_5 and λ_{10}, respectively. This result shows that our protocol reduces the energy consumption imbalance.

On the other hand, we notice that our protocol performs significantly better for memory load imbalance reduction, as shown in Figures 6c, 6d, 6g and 6h. This result is expected, as we explicitly minimize the memory load imbalance.

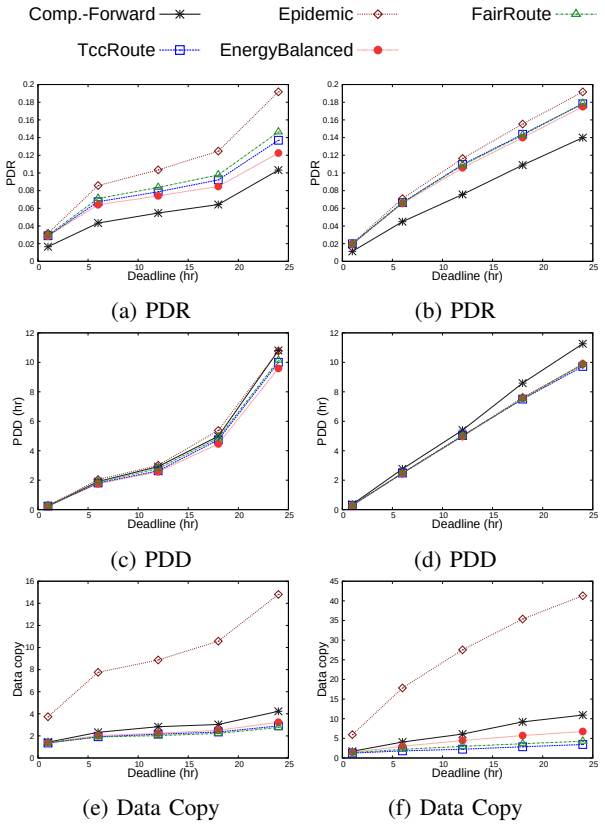
As our *Energy-aware Intra-TCC routing* (Section IV-D) is independent of the OSN forwarding decision, it is interesting to see how it might impact other TCC-aware routing protocols. To this end, we implement TccRoute and FairRoute with *Energy-aware Intra-TCC routing*, which are then named as TccRoute-EA and FairRoute-EA. We also implement our protocol without the *Energy-aware Intra-TCC routing*, named as EnergyBalanced-NEA. Since intra-TCC routing does not affect memory load, we only present the results for energy imbalance, as are shown in Figure 7. First, we note that even EnergyBalanced-NEA outperforms TccRoute and FairRoute in most cases, demonstrating that our memory load controlling mechanism is effective. Second, all TCC-aware routing protocol consistently benefit from *Energy-aware Intra-TCC routing*, as we can see that the imbalance is reduced for all deadlines. Finally, we note that our protocol still outperforms TccRoute-EA and FairRoute-EA in most cases.

C. Routing Performance Results

In this section we present the routing performance results as shown in Figure 8. Notice that we achieve comparable routing performance with other TCC-aware social-based routing protocols, in terms of PDR, PDD and Data copy. PDR is slightly decreased in Reality, while it is comparable in UCSD. Our protocol achieves the same PDD as other protocols for both traces. As for data copy overhead, our protocol has comparable overhead for Reality while slightly increased overhead for UCSD. These results demonstrate a trade-off between PDR, data copy and the energy imbalance. Reducing imbalance results in decreased PDR when maintaining overhead, or increased overhead when the PDR is maintained.

VI. State of the Art

Routing protocols for OSNs have been an active research area recently. Many protocols use different quality metrics to measure node's contact capability, and use different forwarding strategies to make routing decisions. Contact probabilities [9],



(a) PDR (b) PDR

(c) PDD (d) PDD

(e) Data Copy (f) Data Copy

Fig. 8: Routing performance: (a), (c) and (e) for the Reality trace; (b), (d) and (f) for the UCSD trace.

centrality [2] [5] [20], community structures [6] were explored to develop appropriate quality metrics. Most solutions are based on Compare-and-Forward strategy, and thus have a highly imbalanced energy consumption. A comprehensive review of OSN routing protocols can be found in [16].

However, the energy consumption imbalance problem has not yet been thoroughly explored. In [4], the authors mention the cost imbalance problem for delegation forwarding without the consideration of TCC. FairRoute [13] proposes to use queue length to limit the traffic received by popular nodes. A node would forward a message only if the encountered node has higher quality metric and smaller queue length. Similarly, in [8] the authors use buffer space advertisements to avoid storage congestion. In [14], a congestion control framework is proposed to offload traffic from congested individual nodes or network parts to other parts of the network. The framework mainly consists of a set of utilities and an adaptive replication rate. Although congestion control mechanism has positive impact on traffic balance, it does not necessarily guarantee that the aggregate traffic, which is directly related to energy consumption, is also balanced among nodes. Popular nodes are able to deliver messages quickly to destinations, and those messages are therefore purged from their buffer, which makes room for the new messages to be received and the aggregate traffic may still be high. These protocols also do not consider TCC and ignore the fact that popular nodes may also have higher chance to relay messages for other nodes within a TCC. Our proposed solution tries to directly balance the aggregate traffic and avoid hot-zones within TCCs.

VII. Conclusions

In this paper we demonstrate the existence of the energy consumption imbalance problem for OSN routing protocols. We propose the Energy Consumption Balanced Routing protocol which explicitly minimizes the memory load imbalance and avoids over-utilizing nodes when performing intra-TCC routing. We show that our protocol achieves a more balanced energy consumption while maintaining comparable routing performance. In order to explicitly minimize energy consumption imbalance, we will need a more clear understanding of TCC's topology, in addition to its size distribution. By jointly consider both node's memory load and its probability of relaying packets in TCC, we can have a more accurate modeling of energy consumption. We leave this as future work.

Acknowledgment

This work was funded by NSF grants #1127449, #1145858.

References

[1] M. Conti and M. Kumar. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, January 2010.
[2] E. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. MobiHoc '07.
[3] N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, March 2006.
[4] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. MobiHoc '08.
[5] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: A social network perspective. MobiHoc '09.
[6] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay tolerant networks. MobiHoc '08.
[7] A. Keränen, J. Ott, and T. Kärkkäinen. The one simulator for dtn protocol evaluation. Simutools '09.
[8] J. Lakkakorpi, M. Pitkänen, and J. Ott. Using buffer space advertisements to avoid congestion in mobile opportunistic dtns. WWIC' 11.
[9] A. Lindgren, A. Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.
[10] M. McNett and G. Voelker. Access and mobility of wireless pda users. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(2):40–55, April 2005.
[11] O. Pearce, T. Gamblin, B. de Supinski, M. Schulz, and N. Amato. Quantifying the effectiveness of load balance algorithms. ICS '12.
[12] A. Picu, T. Spyropoulos, and T. Hossmann. An analysis of the information spreading delay in heterogeneous mobility dtns. WoWMoM' 12.
[13] J.M. Pujol, A.L. Toledo, and P. Rodriguez. Fair routing in delay tolerant networks. In *INFOCOM 2009, IEEE*, Infocom' 09.
[14] M. Radenkovic and A. Grundy. Efficient and adaptive congestion control for heterogeneous delay-tolerant networks. *Ad Hoc Networks*, 10(7):1322 – 1345, 2012.
[15] S.I. Resnick. *Adventures in Stochastic Processes*. 1992.
[16] M.R. Schurgot, C. Comaniciu, and K. Jaffres-Runser. Beyond traditional dtn routing: social networks for opportunistic communication. *Communications Magazine, IEEE*, 50(7):155–162, July 2012.
[17] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. WDTN '05.
[18] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University, 2000.
[19] C. Yang and R. Stoleru. On balancing the energy consumption of routing protocols for opportunistic social networks. Technical report, Texas A&M University, 2015.
[20] X. Zhang and G. Cao. Efficient data forwarding in mobile social networks with diverse connectivity characteristics. ICDCS '14.