

3R: Fine-grained Encounter-based Routing in Delay Tolerant Networks

Long Vu, Quang Do, Klara Nahrstedt
Department of Computer Science, University of Illinois
Email: {longvu2, quangdo2, klara}@illinois.edu

Abstract—In this paper, we first characterize the fine-grained encounter pattern among mobile users found in a large-scale Bluetooth trace collected by 123 participants at the University of Illinois campus from March to August 2010. Our characterization results show that the fine-grained encounter pattern is regular and predictable. We then present 3R routing protocol, which leverages the regularity of fine-grained encounter pattern among mobile nodes to maximize message delivery probability while preserving message delivery deadline. We evaluate and compare 3R with Prophet and Epidemic routing protocols over the collected trace. Evaluation results show that 3R outperforms other alternatives considerably by improving message delivery while reducing message overhead.

I. INTRODUCTION

It is well known that in reality people usually visit regular places [1], [2], where people make regular social contacts [3] for their daily activities. Exploiting the regularity of contact pattern¹ to expedite message forwarding in Delay Tolerant Networks (DTN) has drawn significant attention from research community [4], [5], [6], [7], [8], [9], [10]. However, existing DTN routing protocols so far have not exploited the regularity of encounter pattern efficiently for improving message forwarding.

The first class of DTN routing protocols was Epidemic routing [11], [12], where the message was flooded to all nodes in the network until the message was delivered at the receiver. On one hand, Epidemic routing provided the nearly optimal message delivery if one intermittently connected path ever existed [6]. On the other hand, the regularity of encounter pattern among mobile nodes in the network was not exploited and thus Epidemic routing usually incurred broadcast storms of messages.

The second class of DTN routing protocols focused on mobile nodes whose movement schedules were fixed [4], [5] to forward message. This approach was to assume the non-randomness in the movement of nodes, and leveraged the non-randomness to optimize the message routing. As a result, this approach could only be applicable if nodes did not change their movement schedules, which might not be always true in many real world scenarios.

The third class of DTN routing protocols only exploited the coarse-grained encounter information of mobile nodes such as Prophet[6], EBR[8], PASR[9], MEED[13], FER[14], and MaxProp[15]. Particularly, these protocols

compressed/summarized all encounter information between a node pair into a coarse-grained piece of information called encounter probability, and used it to route messages. However, this summarized encounter probability represented neither the trend nor the specific characteristics of encounter pattern between a particular node pair. For example, two node pairs had the same compressed encounter probability but they might have different encounter patterns: one pair might meet during weekdays while the other might meet only at weekend. Therefore, using only the coarse-grained encounter probability might result in inefficient message forwarding.

The fourth class of DTN routing protocols leveraged encounter information of people within one community to route messages [16], [17], [18], [19], [20]. For these protocols, a graph was used to represent a community in which one vertex (or node) was one community member. The edge between a node pair existed if the two corresponding community members met at least once. The node degree was used to select message forwarder, where the node with a higher node degree was preferable. Essentially, node degree was a coarse-grained encounter information, which was similar to the compressed encounter probability. As a result, this class of routing protocols had similar drawbacks as those of the third class of DTN routing protocols above.

In summary, since previous DTN routing protocols used only the coarse-grained encounter information, they may not fully exploit encounter information for message forwarding. In reality, the movement behavior of people, and hence their encounter patterns, may differ between weekday and weekend, or even among different time slots within the same day. Thus, the coarse-grained encounter information may not capture the regularity of encounter pattern among people. So, knowledge of the fine-grained encounter information is needed for an improved DTN routing solution.

In this paper, we first characterize the fine-grained encounter pattern found in our encounter trace collected by 123 participants in University of Illinois campus. Our characterization results show that the fine-grained encounter pattern is regular and predictable. Then, we present the 3R routing protocol², which leverages the fine-grained encounter information among mobile nodes to improve message delivery. Particularly, 3R divides encounter time into a

¹In this paper, terms “contact” and “encounter” are used interchangeably.

²3R stands for fine-grained encounter-based Routing.

finer granularity of type of day and time slot. Then, 3R estimates encounter probability with respect to this fine-grained time division. Finally, the fine-grained encounter probability is used in message forwarding. This paper has following contributions:

- 1) We show that the fine-grained encounter pattern found in our large-scale encounter trace is regular and predictable. To the best of our knowledge, we are the first to characterize the regularity of fine-grained encounter pattern from a real large-scale encounter trace.
- 2) We exploit the regularity of fine-grained encounter pattern in the design of 3R routing protocol to maximize message delivery probability and preserve message delivery deadline.
- 3) We evaluate and compare 3R with Epidemic [11] and Prophet [6] routing protocols over the collected encounter trace. Evaluation results show that 3R outperforms other alternatives considerably.

This paper is organized as follows. We first characterize the regularity of the fine-grained encounter pattern found in real encounter trace in Section II. Then, we present the 3R protocol in Section III. In Section IV, we compare the performance of 3R with Epidemic and Prophet routing protocols. Finally, we conclude the paper in Section V.

II. CHARACTERIZING REGULARITY OF FINE-GRAINED ENCOUNTER PATTERN IN REAL ENCOUNTER TRACE

A. Collection of Real Encounter Trace

Recently, we deployed a scanning system named UIM (i.e., University of Illinois Movement) on Google Android phones [21]. UIM has a Wifi scanner and a Bluetooth scanner. The former periodically (i.e., every 30 minutes) captures MAC addresses of Wifi access points, while the latter periodically (i.e., every 60 seconds) captures MAC addresses of Bluetooth-enabled devices in proximity of experiment phones. Then, 123 experiment phones were carried by 123 participants from March to August 2010 in University of Illinois campus. Participants include faculties, grads, undergrads, and staffs. In this paper, we use the terms person and phone, Bluetooth and BT interchangeably. We also focus only on the collected BT trace as shown in Table I, in which each row is created by one BT scan, and has a scan time η and a set Φ of scanned BT devices. Our collected BT trace is a **fine-grained encounter trace** with respect to time and location since: (1) each row is appended to the trace every 60 seconds (e.g., one BT scan), and (2) the BT transmission range of the phone is about 10 meters, thus if the phone can scan a device, then the phone and the device stay in the same geographical location. Since the BT trace is in the format of a table, which is similar to a relation in the Relational Algebra, we use Relational Algebra [22] to manipulate the BT trace. Henceforth, we use the terms table and relation, row and tuple interchangeably.

η	Φ
03/08/10 09:20	n_1, n_3
03/08/10 09:21	n_1, n_3
03/08/10 09:22	n_1
03/08/10 13:50	n_4, n_9
03/14/10 08:14	n_1, n_3, n_8

Table I
BT TRACE B_p OF PHONE p

Formally, let B_p be the collected BT trace of the experiment phone p . The relation B_p has multiple tuples: $B_p = \{b_1, b_2, b_3, \dots, b_k, \dots, b_{|B_p|}\}$. Each tuple $b_k \in B_p$ is in the format of $b_k = \langle \eta_k, \Phi_k \rangle$, where η_k is the scan time of the BT scan and Φ_k is a set of BT MACs returned from that scan. So, we have $\Phi_k = \{n_1, n_2, \dots, n_j, \dots\}$, in which n_j is the j^{th} BT MAC scanned by the BT scanner of p during the entire experiment period. Let Δ_p be the set of all BT MACs scanned by the BT scanner for the entire experiment period. In Table I, $\Delta_p = \{n_1, n_3, n_4, n_8, n_9\}$.

Definition of “fine-grained encounter”: Let $enc(p, n, t)$ be the predicate, which represents that two devices p and n have a fine-grained encounter at time t (or the carriers of the two devices have a social contact at time t). We define $enc(p, n, t)$ as follows:

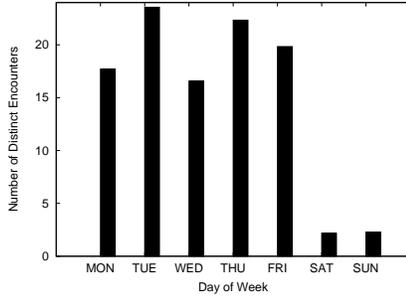
- 1) For a phone p and a device n , $enc(p, n, t) \Leftrightarrow (n \in \Phi_i) \wedge (\eta_i = t) \wedge (b_i = \langle \eta_i, \Phi_i \rangle) \wedge (b_i \in B_p)$. Intuitively, the phone p has one encounter with the device n at time t if n belongs to a tuple in B_p , whose scan time is t . In Table I, p and n_1 have an encounter at “03/08/10 09:20”.
- 2) For a phone p and two devices n_1, n_2 , $enc(n_1, n_2, t) \Leftrightarrow enc(p, n_1, t) \wedge enc(p, n_2, t)$. Intuitively, two devices n_1 and n_2 meet at time t , if they both exist in one tuple of B_p , whose scan time is t . In Table I, n_4 and n_9 meet at “03/08/10 13:50”.

Next, we characterize the fine-grained encounter pattern found in our BT trace.

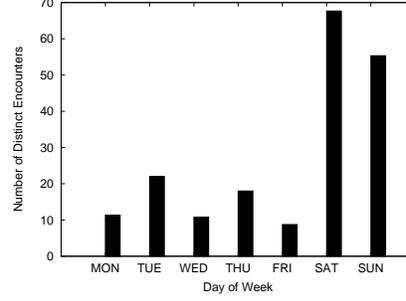
B. Classification of Fine-grained Encounter Pattern

We observe four different fine-grained encounter patterns in the collected BT traces of 123 participants as shown in Figures 1 and 2. In these figures, for a particular phone p , we calculate the average number of unique BT MACs scanned by p for each day of week.

Figure 1(a) shows the first and the most common encounter pattern in our data set, where people have a considerably higher number of encounters during the weekday than the weekend. This pattern is common since in university campus most professors and students have classes and meetings where they meet other people during the weekday. On the other hand, professors and students usually travel less and thus they meet less number of people at the weekend. Figure 1(b) shows the second encounter pattern, which is totally opposite to the first pattern above. People who belong

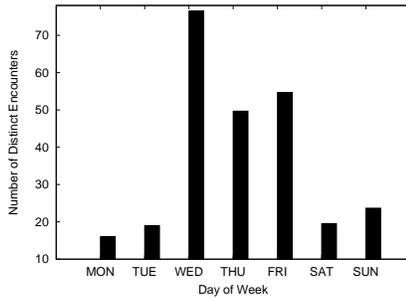


(a) Encounter number decreases at the weekend

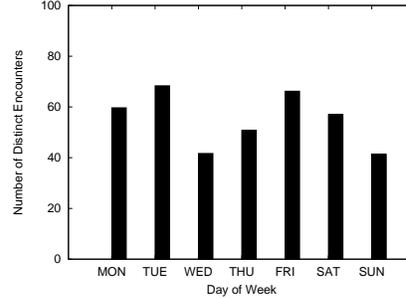


(b) Encounter number increases at the weekend

Figure 1. Encounter number changes significantly between weekday and weekend



(a) Encounter number grows high in midweek



(b) Encounter number stays in a small range

Figure 2. Encounter number grows high in midweek or stays in a small range

to this pattern have more social contacts at the weekend and less during the weekday. We find that in our collected trace of 123 participants, more than 95% of participants exhibit the first and the second encounter patterns.

Figure 2(a) shows the third encounter pattern, where the number of encounters stays high only in the midweek. This pattern may belong to people who have a strictly working schedule since they always have the busiest schedule in the midweek. Figure 2(b) shows the last encounter pattern in which the number of encounters stays in a relatively small range for all days. People belong to this encounter pattern meet a similar number of other people everyday, regardless of weekday or weekend.

In conclusion, the characterization results show that the encounter patterns of more than 95% participants depend on type of day, where the number of encounters differs significantly between weekday and weekend.

C. Regularity of Fine-grained Encounter Pattern

To characterize the regularity of fine-grained encounter pattern for a particular participant, the participant must have a long enough collected BT trace. Therefore, we select a set of 50 phones from 123 experiment phones, in which each selected phone collected from 20 to 50 days of BT trace. Let d_i be the number of days the selected phone p_i collected BT trace and let B_{p_i} be the collected BT trace, we

have: $1 \leq i \leq 50$ and $20 \leq d_i \leq 50$. Let Δ_{p_i} be the set of all unique Bluetooth MACs scanned by the BT scanner of p_i for d_i days. The next step is to calculate the number of regular fine-grained encounters found in B_{p_i} . To this end, we use the “time slot” and “support value” as follows.

First, we obtain the fine granularity of time by dividing a day into time slots of size τ hours. For example, with $\tau = 6(h)$, we have following time slots ([00:00:06:00), [06:00:12:00), [12:00:18:00), [18:00:24:00)) for one day. For a scanned BT MAC $n_j \in \Delta_{p_i}$, let d_{ij} be the number of days p_i and n_j meet, in which the encounters happen at the same time slot of these d_{ij} days. For example, p_i and n_j may meet every weekday during the [8AM:10AM) time slot. In our context, **a fine-grained encounter between the phone p_i and a scanned Bluetooth MAC $n_j \in \Delta_{p_i}$ is a “regular fine-grained encounter” if $d_{ij} \geq d_i \cdot \epsilon$** , where ϵ is the threshold or the support value. In other words, a fine-grained encounter is a regular fine-grained encounter if p_i and n_j meet at the same time slot for at least $d_i \cdot \epsilon$ days during the experiment period of d_i days. Next, we vary the values of τ and ϵ to characterize the regular fine-grained encounters found in the trace of 50 participants. Henceforth, we use the terms “encounter” and “fine-grained encounter” interchangeably.

In Figure 3(a), we fix $\tau = 6(h)$ and vary ϵ from 0.4 to 0.7. We find that when ϵ increases the number of participants

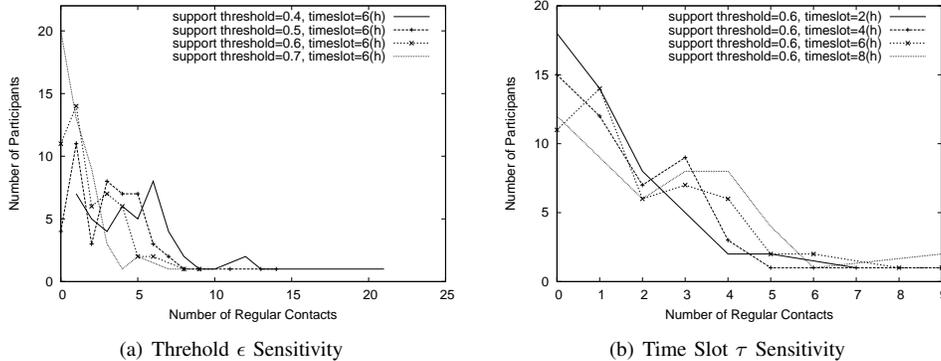


Figure 3. Regularity of the fine-grained encounter pattern

with a smaller number of regular encounters increases. This result is intuitive since with a higher support value ϵ , an encounter needs to happen at the same time slot more frequently to be considered regular. For $\epsilon = 0.7$, 20 (out of 50) participants have no regular encounter. Notice that $\epsilon = 0.7$ is a very high support value as shown in the following example. For a month (e.g., $d_i = 30$), we have 8 days of the weekend and 22 weekdays. With $\epsilon = 0.7$, an encounter at the weekdays is regular only if it happens at the same time slot in $30 \cdot 0.7 = 21$ (days). For $\epsilon = 0.5$, Figure 3(a) shows that 45 participants have at least 1 regular encounters and only 5 participants have no regular encounter.

Next, we fix $\epsilon = 0.6$ and vary τ from 2(h) to 8(h). In Figure 3(b), we find that when τ increases, the number of participants with more regular encounters increases. This is because with a larger time slot, the definition of regular encounter becomes relaxed. For example, with $\tau = 2$ (h), two people must meet during this 2 hour time slot to be considered regular encounter. However, with $\tau = 8$ (h), the encounter may happen anytime during 8 hours, which occurs with a higher probability in reality. Figure 3(b) also shows that when τ varies from 2(h) to 8(h), from 33 to 38 participants have at least 1 regular encounter.

In conclusion, for different values of τ and ϵ , the encounters found in the trace of 50 participants are regular.

D. Discussion

Our studies in Section II-B and Section II-C show that the fine-grained encounter pattern can be characterized from the fine-grained encounter trace. More importantly, we find that the fine-grained encounter pattern: (1) depends on the type of day, particularly encounter pattern of more than 95% participants differs significantly between weekday and weekend, and (2) is regular and predictable.

Although it is believed that people exhibit regular movement patterns for their daily activities [1], [2] and their encounter patterns are regular [3], to the best of our knowledge we are the first to characterize the regularity of fine-grained encounter pattern from a large-scale real encounter trace.

Next, we exploit the regularity of fine-grained encounter pattern in the design of the 3R routing protocol.

III. 3R: FINE-GRAINED ENCOUNTER-BASED ROUTING

In this section, we first present the overview of 3R. Second, we present how 3R bootstraps and maintains the set of encounters. Third, we present how to estimate encounter probability from the past encounter trace and how to construct the routing table. Finally, we present how 3R uses routing table to route the message.

Name	Description
m	Message transmitted in network
s	Sender of m
r	Receiver of m
n	A mobile node
D_m	Delivery deadline of message m
T_m	Time at which n delivers m to r
P_t^n	Probability that n delivers m to r during $[t, D_m]$
R_n	Routing table of node n
ν_i	Type of day, $\nu_1 = \textit{weekend}$, $\nu_2 = \textit{weekday}$
τ_j	Time slot j^{th} in one day
m_ν	Type of day of m when m is routed from s to r
ρ	Time duration (e.g., $\rho = [08 : 00, 10 : 00]$)
Δ_n	Set of mobile nodes n has met so far

Table II
NOTATIONS USED FOR THE DESIGN OF 3R

A. Protocol Overview

1) *Scenario*: In this paper, we focus on the scenario where the mobile user carries his mobile phone during his daily activities. The phone runs the BT scanner that captures the BT MAC addresses of Bluetooth-enabled devices in the proximity of the phone and stores the collected BT MAC addresses in the memory of the phone. In other words, the phone captures the social encounters between this mobile user and other people he meets. Since a typical person has regular movement pattern and meets regular set of people for his daily activities, his mobile phone captures the regular encounters between him and others. 3R uses the

past encounter trace to predict future encounter, and thus can find the best message forwarder to transmit the message from the sender to the receiver.

2) *System Model*: We consider a network formed by the peer-to-peer (P2P) connectivity among mobile devices when they meet. Since the direct path between the sender and the receiver may not exist instantly, our network is a Delay Tolerant Network. Each node has a mandatory P2P interface such as Wifi, Bluetooth, etc. In our network, the encounter pattern among nodes is regular. Data messages forwarded among mobile nodes are text messages or short video clips.

3) *Protocol Overview*: Table II shows notations used in following sections. For a message m , when m is sent out by the sender s , s sets the message delivery deadline D_m . Objective of 3R is to route m from s to the receiver r by delivery deadline D_m . We formulate 3R protocol as an optimization problem, which maximizes the delivery probability of m to r and meets the delivery deadline D_m :

$$\begin{aligned} & \max P^n_t \\ & \text{s.t. } T_m \leq D_m \end{aligned} \quad (1)$$

3R obtains the objective function for the constraint of Equation 1 as follows. Let $|\Delta_n|$ be the number of unique mobile nodes the node n has met so far. At the time t in the routing process, assuming that m is carried by a mobile n_1 and n_1 meets n_2 , if $P^{n_2}_t > P^{n_1}_t$, then n_1 transmits m to n_2 , which will carry and forward m towards r . Node n_2 will then find a node with a better delivery probability to transmit m towards r . If $P^{n_2}_t = P^{n_1}_t$, then the node with the greater value of $|\Delta_n|$ will be chosen as the next message forwarder since it has a higher probability to directly reach the receiver r or to meet nodes with higher delivery probability. By doing this, 3R maximizes the delivery probability of m by the delivery deadline D_m .

Each node n has a routing table R_n , which is derived based on the regularity of past encounter pattern between n and nodes n has met so far. R_n is constructed in a totally distributed manner and only depends on the local encounter trace of n . Then, R_n is used to find the better message forwarder in the routing process. Finally, 3R keeps only a single copy of the message m in the network during the routing process to reduce message overhead.

B. Bootstrapping 3R

Node n has a BT scanner that captures the encounters between n and other nodes. After several weeks since n starts capturing encounters, n has collected enough data to infer the regular encounter pattern. Then, n uses this collected encounter trace to construct the first routing table R_n . Notice that after the first routing table is constructed, the BT scanner continues to collect more encounters, which is used to update R_n periodically later. The routing table can be updated periodically (e.g., weekly).

C. Updating Δ_n

Node n keeps a set of encounters Δ_n that includes all nodes n has met so far. Anytime n meets a new node n_1 , n adds n_1 to Δ_n . The size of the set Δ_n is $|\Delta_n|$, which is the number of unique nodes n has met so far. $|\Delta_n|$ is similar to the node degree in the social graph used in the previous works to select the next message forwarder [16], [17], [23]. In the following sections, we present how 3R uses $|\Delta_n|$ in the message forwarding process.

D. Estimating Encounter Probability

As presented in Section II-B and Section II-C, the regularity of encounter pattern depends on type of day (in a week) and time slot (in a day). To capture these characteristics, we classify the scan time in BT trace into type of day and time slot. The intuition of this time classification is that people usually follow their daily scheduled routines and thus a person may meet different sets of people in different days (weekday vs. weekend) and at different time slots in the same day. To this end, we convert the collected BT trace B_n of n in Table I into the converted trace B'_n as shown in Table III. Particularly, we add two columns: type of day ν and time slot τ for each row in Table I. We have two types of day: weekend and weekday. The number of time slots in a day depends on the size of the time slot τ . For example, if $\tau = 1(\text{h})$, we have 24 time slots for a day. In the converted format, $B'_n = \langle b_1, b_2, \dots, b_k, \dots, b_{|B'_n|} \rangle$, with $1 \leq k \leq |B'_n|$.

ν	τ	η	Φ
weekday	[09:10]	03/08/10 09:20	n_1, n_3
weekday	[09:10]	03/08/10 09:21	n_1, n_3
weekday	[09:10]	03/08/10 09:22	n_1
weekday	[13:14]	03/08/10 13:50	n_4, n_9
weekday	[08:09]	03/14/10 08:14	n_1, n_3, n_8

Table III
CONVERTED BT TRACE B'_n FOR TIME SLOT SIZE OF 1 HOUR

Let $p^{n'}_{ij}$ be the encounter probability between node n and node $n' \in \Delta_n$ for the type of day ν_i and the time slot τ_j . $p^{n'}_{ij}$ is calculated as follows:

$$p^{n'}_{ij} = \frac{|\sigma_{\varphi=\{\nu_k=\nu_i, \tau_k=\tau_j, n' \in \Phi_k\}} B'_n|}{|\sigma_{\varphi=\{\nu_k=\nu_i, \tau_k=\tau_j\}} B'_n|} \quad (2)$$

In Equation 2, the selection operation σ is performed over all tuples $b_k = \langle \nu_k, \tau_k, \eta_k, \Phi_k \rangle \in B'_n$. Particularly, $|\sigma_{\varphi=\{\nu_k=\nu_i, \tau_k=\tau_j, n' \in \Phi_k\}} B'_n|$ is the number of tuples b_k , which belongs to the type of day ν_i and time slot τ_j and $n' \in \Phi_k$. Similarly, $|\sigma_{\varphi=\{\nu_k=\nu_i, \tau_k=\tau_j\}} B'_n|$ is the number of tuples b_k , which belongs to the type of day ν_i and time slot τ_j . The next step is using the estimated encounter probability to construct the routing table R_n .

ν	τ	n_1	n_2	n_3	n_4
weekday	(08:00,09:00]	0.4	0.2	0.1	0
weekday	(09:00,10:00]	0.1	0	0.6	0
weekday	(10:00,11:00]	0	0	0.5	0.6
weekday	(12:00,13:00]	0	0	0.8	0
...
weekend	(08:00,09:00]	0	0.8	0	0
weekend	(09:00,10:00]	0	0	0.2	0.7
...

Table IV
EXAMPLE OF R_n FOR TIME SLOT SIZE OF 1 HOUR

E. Constructing Routing Table R_n

Table IV shows an example of a routing table R_n . R_n is a relation of $|\Delta_n| + 2$ attributes. The first two attributes are the type of day ν and time slot τ , and the next $|\Delta_n|$ attributes are nodes n has met so far. Each tuple is one routing entry, whose first two values are type of day ν and time slot τ . The last $|\Delta_n|$ values of a tuple are the encounter probabilities between n and the corresponding mobile node for the type of day and time slot. The column corresponds to the node n' in R_n essentially represents all encounter probabilities between n and n' for all types of day and time slots. In Table IV, for the type of day “weekday” and the time slot (08 : 00,09 : 00] the nodes n and n_1 meet with the probability of 0.4. The first tuple in this relation is $\langle weekday, (08 : 00,09 : 00], 0.4, 0.2, 0.1, 0 \rangle$. The column of n_2 in Table IV includes encounter probabilities between n and n_2 for all types of day and time slots. We will present how to construct the routing table R_n in the next step.

For each node $n_1 \in \Delta_n$, we create the set of queries $\chi^{n_1} = \{X_1, X_2, \dots, X_k, \dots, X_{|\chi^{n_1}|}\}$ in which $X_k = \{\nu_k, \tau_k\}$ where $\nu_k \in \Upsilon = \{weekday, weekend\}$, and $\tau_k \in \Sigma$. The set Σ is constructed based on the size of time slot τ . For example, if $\tau = 1(h)$, then a day has 24 slots and $\Sigma = \{(00 : 00, 01 : 00], (01 : 00, 02 : 00], \dots, (23 : 00, 24 : 00]\}$. Next, we apply Equation 2 for each query $X_k \in \chi^{n_1}$ to calculate the encounter probabilities between n and nodes n_1 . After having the encounter probabilities for all queries in χ^{n_1} , we update the columns of n_1 in R_n with the new encounter probabilities. Notice that the same sets of Υ and Σ are used for all nodes in Δ_n when we construct R_n .

The relation R_n is in the format of $R_n = \{e_1, e_2, \dots, e_{|R_n|}\}$ with $|\Upsilon| \cdot |\Sigma|$ tuples. Particularly, $|\Upsilon| = 2$ and $|\Sigma|$ depends on τ . For example, with $\tau = 1(h)$, $|\Sigma| = 24$ and R_n has 48 tuples. Each tuple $e_k \in R_n$ is in the format of $e_k = \langle \nu_i, \tau_j, p^{n_1}_{ij}, p^{n_2}_{ij}, \dots, p^{n_u}_{ij}, \dots, p^{n_{|\Delta_n|}}_{ij} \rangle$, where $p^{n_u}_{ij}$ is the encounter probability between n and n_u for the type of day ν_i and during the time slot τ_j . Here, $1 \leq u \leq |\Delta_n|$ and $0 \leq p^{n_u}_{ij} \leq 1$. Notice that R_n is constructed locally by node n and R_n only depends on the BT trace collected by n .

1) *Extensible Routing Table*: Over time, node n may meet new nodes. Therefore, R_n is an extensible routing table

that grows when new nodes are added into the set Δ_n .

2) *Finer Grain Routing Table*: Classifying time into type of day and time slot does capture the regularity of people movement. However, there are cases where a finer grain routing table becomes preferable. For example, a student may always attend a class from 10AM to 11AM every Tuesday, a professor may always give a lecture from 3PM to 4PM every Friday. For these cases, we can classify time into a finer granularity, for example time can be classified into day of week such as Monday, Tuesday, etc. (rather than type of day $\{weekday, weekend\}$), and time slot of size τ . For the finer time classification, we have a finer grain routing table with more routing entries. For example, for $\tau = 1(h)$, we have $7 \cdot 24 = 168$ routing entries. However, there are two important tradeoffs. First, the finer classification requires more data to be collected (or a longer training time) for more accurate encounter probabilities in the construction of R_n . Second, the finer classification works better only for people whose encounter patterns are strictly repeated; as a result, for people with a more relaxed encounter patterns, a finer classification may result in inefficient message forwarding.

F. Forwarding Message

Given the routing table R_n , node n uses R_n to transmit the message m from the sender to the receiver.

In the routing process, node n is preferable as the next message forwarder if n provides m a higher delivery probability to the receiver r by the message delivery deadline D_m . Therefore, at time t when node n_1 (assuming that n_1 is carrying message m) meets node n_2 , n_1 calculates its delivery probability $P^{n_1}_t$, which is the probability n_1 delivers m to r during the time period $[t, D_m]$. Similarly, n_2 calculates its delivery probability $P^{n_2}_t$. If $P^{n_1}_t < P^{n_2}_t$, m is transmitted from n_1 to n_2 and then n_2 becomes the next forwarder of m . If $P^{n_1}_t = P^{n_2}_t$, $|\Delta_{n_1}|$ and $|\Delta_{n_2}|$ are compared and the node with a greater size of Δ_n will be the next forwarder. The intuition is as follows: the node that has met more nodes in the past will likely meet more nodes in the future, and thus that node can deliver m to r with a higher probability. The next step is to use R_n to calculate P^n_t .

When m is sent at the sender s , s obtains the type of day $m_\nu \in \{weekday, weekend\}$ and appends m_ν to m . At time t , m_ν is retrieved by n and n creates a relation E by performing a selection operation over R_n as follows:

$$E = \sigma_{\varphi=\{\nu=m_\nu, \tau \in \Sigma'\}} R_n \quad (3)$$

In Equation 3, φ is the condition of the selection operation over R_n , which basically filters out irrelevant tuples. In particular, relation E consists of only the tuples of R_n that have the type of day m_ν and the time slot in the set Σ' , which is created as follows. For a time slot τ_k , let τ^s_k be the starting time of τ_k and τ^e_k be the ending time of τ_k . For example, if $\tau_k = (08 : 00, 10 : 00]$, then we have

$\tau^s_k = 08 : 00$ and $\tau^e_k = 10 : 00$. For the duration $[t, D_m]$, we have $\Sigma' = \{\tau_k : \tau^s_k \geq t, \tau^e_k \leq D_m\}$. For example, in Table IV, if $m_\nu = \text{weekday}$, $t = 08 : 00$, and $D_m = 11 : 00$, then the relation E consists of the first three tuples as shown in Table V.

ν	τ	n_1	n_2	n_3	n_4
weekday	(08:00-09:00]	0.4	0.2	0.1	0
weekday	(09:00-10:00]	0.1	0	0.6	0
weekday	(10:00-11:00]	0	0	0.5	0.6

Table V

RELATION E WITH $m_\nu = \text{weekday}$, $t = 08 : 00$, AND $D_m = 11 : 00$

Given the relation E obtained by applying the Equation 3 for the type of day m_ν and time slots during the period $[t, D_m]$, we then create a relation S by performing a projection operation over E as follows:

$$S = \pi_{n_u=r}(E) \quad (4)$$

In Equation 4, we obtain the relation S by extracting the attribute $n_u = r$ from the relation E . In other words, the table S has only one column, which consists of the encounter probabilities between node n and the receiver r obtained by projecting the column of r in relation E . For example, in Table IV, if $m_\nu = \text{weekday}$, $t = 08 : 00$, $D_m = 11 : 00$, and $r = n_3$, then $S = \{0.1, 0.6, 0.5\}$ as shown in Table VI. Since S has only one column, we use the term ‘‘set S ’’, ‘‘relation S ’’, and ‘‘table S ’’ interchangeably in following sections. Formally, we have $S = \{p_1, p_2, p_3, \dots, p_{|S|}\}$, where $0 \leq p_j \leq 1$, $1 \leq j \leq |S|$. p_j represents the encounter probability between n and r during the time slot j^{th} and $p_j = 0$ means n and r have no encounter during the j^{th} time slot in the past.

$n_3 = r$
0.1
0.6
0.5

Table VI

RELATION S WITH $m_\nu = \text{weekday}$, $t = 08 : 00$, $D_m = 11 : 00$, AND $r = n_3$

Notice that the order of elements in S corresponds to the order of time slots in relation E . That is, p_1 is the encounter probability between n and r during the first time slot after time t , and $p_{|S|}$ is the encounter probability between n and r during the last time slot before D_m . The set S is then used to calculate P^n_t , the probability that n delivers m at r during the period $[t, D_m]$, as follows:

$$P^n_t = \sum_{j=1}^{|S|} p_j^d \quad (5)$$

In Equation 5, p_j^d is the delivery probability that n successfully delivers m to r at time slot j^{th} . p_j^d is calculated based on encounter probability in set S as follows. We observe that node n only delivers m to r at time slot j^{th} if n fails to deliver m to r in the first $(j-1)$ time slots and n meets r at time slot j^{th} . This happens only if n and r have no encounters during the first $(j-1)$ time slots and n and r encounter during the j^{th} time slot. So, we have:

$$p_j^d = \left\{ \prod_{k=1}^{j-1} (1 - p_k) \right\} \cdot p_j \quad (6)$$

The Equation 6 is used to calculate p_j^d for each time slot j where p_k is taken from the set S . Then, P^n_t is calculated accordingly by using the Equation 5. Notice that $P^n_t = 0$ if n has not met r so far.

1) *Overnight Message*: These are two cases we need to consider. For the first case, m is routed overnight from the weekday to the weekend and via versa. In this case, during the routing process, n updates m_ν with the current type of day. For the second case, m is routed overnight from one day to the next day. In this case, the selection operation over R_n in Equation 3 needs to be changed so that the set Σ' includes all necessary time slots of these two days. Moreover, the time slots in set E needs to be re-ordered so that time slots of the next day should be after the ones of the day before. Then, Equation 4 remains unchanged.

2) *Distributed Routing Solution*: Objective function of the Equation 1 is obtained in the routing process since the calculation of P^n_t takes D_m into account. Notice that the routing decision is made in a totally distributed manner since P^n_t is calculated by using only the local routing table R_n and does not rely on other nodes in the network. Moreover, only one copy of the message m is kept among mobile nodes in the network during the routing process.

IV. EVALUATION

A. Setting

We select a set Ω of 9 BT traces collected by phones carried by 9 grads from the same research group in the department of Computer Science, University of Illinois from March 1, 2010 to March 20, 2010. From Ω , we have a set D of 100 unique BT MACs, including these 9 experiment phones.

We then use D to evaluate and compare the performance of 3R with Epidemic[11] and Prophet[6] routing protocols. Epidemic routing is basically a flooding-based scheme which floods the message from the current forwarder to any new encountered nodes, which have not carried the message. This protocol provides a high delivery probability; however, it incurs a high message overhead. Meanwhile, Prophet uses the compressed encounter probability between pairs of nodes to select the next message forwarder rather than classifying time into type of day and time slot like what 3R does. For a

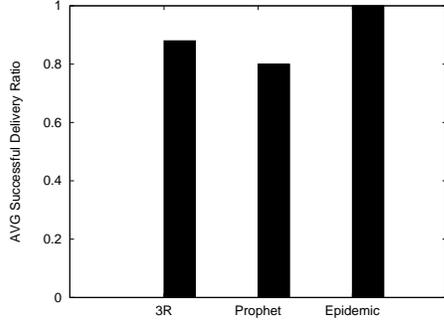


Figure 4. Comparison of Average Successful Delivery Ratio

fair comparison, Prophet only uses one copy of the message in the forwarding process. Since the calculated probability is not for type of day and time slot, Prophet does not capture the detailed regularity of encounter pattern.

To compare the performance of 3R with Epidemic and Prophet routing protocols, we create a set of 100 (sender,receiver) pairs $\Gamma = \{(s_i, r_i) : 1 \leq i \leq 100\}$, in which s_i is the sender, r_i is the receiver, and $|\Gamma| = 100$. For a pair of (s_i, r_i) , we first select $s_i \in \Omega$ at random (notice that we have $|\Omega| = 9$). Then, we select a random day $d \in [03/01, 03/19]$ during the experiment period, let $D^d_{s_i}$ be the set of BT records collected by s_i during the day d . Notice that records of $D^d_{s_i}$ are sorted increasingly according to the scan time ($D^d_{s_i}$ is in the format of the Table I). Let R be the set BT MACs extracted from the last 30 records in $D^d_{s_i}$. Formally, $R = \{n_i : n_i \in \Phi_j, b_j = \langle \eta_j, \Phi_j \rangle \in D^d_{s_i}, |D^d_{s_i}| - 30 \leq j \leq |D^d_{s_i}|\}$. Finally, the receiver r_i is selected at random from the set R .

There are two reasons to select the sender s_i and the receiver r_i as above. First, as presented in a survey of 300 faculties and students[24], the delay that people can tolerate is at most 10 hours, depending on the DTN networking applications[24]. We also observe that since the set D consists of BT MACs collected by participants in the same research group, nodes in D usually meet during the working period from 8AM to 6PM. So, by selecting r_i from the last 30 records of $D^d_{s_i}$, we basically set the deadline for the message transmission at the end of the office hour (i.e., around 6PM) and thus the delivery deadline is in the range of 8 to 10 hours. We believe this delivery deadline is realistic. Second, since r_i is selected based on the set $D^d_{s_i}$, s_i always can deliver the message m to r_i if the Epidemic routing protocol is used. In other words, we expect that Epidemic routing will have 100% delivery ratio and the shortest message delivery time since it floods the message m to all possible nodes in the network. So, Epidemic routing is used as the base line in the comparison. 3R learns the regularity of encounter patterns (e.g., construct the routing table) for all 100 nodes by using the collected traces from 03/01/2010 to 03/19/2010. Then, 3R uses the constructed

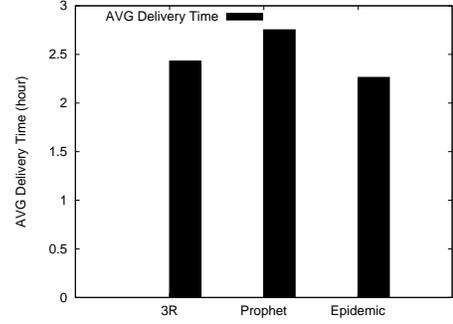


Figure 5. Comparison of Average Delivery Time

routing tables to forward messages.

Our comparison focuses on metrics Average Successful Delivery Ratio, Average Delivery Time, and Average Message Overhead. For each metric, we perform the routing for all 100 (sender,receiver) pairs and plot the average.

B. Evaluation Result

Figure 4 shows that Epidemic outperforms 3R and Prophet in terms of Average Message Delivery Ratio due to its flooding-based nature. Since the receiver is chosen in the same day of the sender, Epidemic routing can always deliver the message to the receiver. Meanwhile, 3R obtains 89% of successful delivery since it exploits the regularity of people movement to forward the message, which is not exploited by Prophet. This figure also shows that Prophet only obtains 78% successful delivery.

Figure 5 shows that Epidemic routing obtains the shortest delivery time of 2.3 hours. Correspondingly, 3R obtains 2.43 hours and Prophet obtains 2.75 hours. Notice that this figure is only for delivered messages. In other words, 89% of messages delivered by 3R and 78% of messages delivered by Prophet are taken into calculation for this plot.

Figure 6 shows that while 3R needs 2.2 messages, Prophet needs 2.75 messages and Epidemic needs 22 messages to send one message m successfully. So, Epidemic incurs 10 times of message overhead in comparison to 3R.

C. Discussion

Evaluation results show that Epidemic routing is slightly better than 3R for message delivery ratio and message delivery time. However, 3R significantly outperforms Epidemic for message overhead.

Evaluation results also show that 3R slightly outperforms Prophet. This is because the trace used for the comparison was collected by 9 grads, who: (1) were in the same research group and met each other almost everyday, since grads usually came to their labs even at the weekend, and (2) usually only visit a limited number of places (e.g., home and lab) and make a small number of social contacts in their daily activities. Therefore, the encounter patterns of

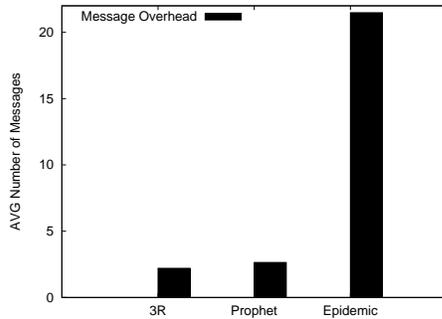


Figure 6. Comparison of Average Message Overhead

these grads may not be significantly different for different types of day and different time slots in the same day. In other words, the average encounter probability may be acceptable to represent the encounter patterns of these grads. As a result, the performance of Prophet becomes comparable to that of 3R since Prophet exploits the average encounter probability in forwarding message. We believe that if the trace is collected by people whose encounter patterns and micro-mobility show regularity on a finer granularity (e.g., hourly basis, daily basis), then Prophet would perform much worse but 3R would do much better.

V. CONCLUSION

Characterization study from the real collected BT trace shows that the fine-grained encounter pattern of people is regular. 3R exploits the regularity of fine-grained encounter pattern and provides a totally distributed routing solution to expedite message routing in Delay Tolerant Networks. We evaluate and compare 3R with Epidemic and Prophet routing protocols over the real fine-grained encounter trace and evaluation results show that 3R outperforms other alternatives.

REFERENCES

- [1] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility pattern," *Nature*, vol. 453, pp. 779–782, June 2008.
- [2] J.-K. Lee and J. C. Hou, "Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application," in *Mobihoc*, 2006.
- [3] L. McNamara, C. Mascolo, and L. Capra, "Media sharing based on colocation prediction in urban transport," in *Mobicom*, 2008.
- [4] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Mobihoc*, 2004.
- [5] S. Jain, K. Fall, and R. Patra, "Routing in delay tolerant network," in *Sigcomm*, 2004.
- [6] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," in *Mobihoc*, 2003.
- [7] N. Sastry, K. Sollins, and J. Crowcroft, "Delivery properties of human social networks," in *Infocom*, 2009.
- [8] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in dtns," in *Infocom*, 2009.
- [9] Z. Guo, B. Wang, and J.-H. Cui, "Prediction assisted single-copy routing in underwater delay tolerant networks," in *Globecomm*, 2010.
- [10] H. Cai and D. Y. Eun, "Aging rules: What does the past tell about the future in mobile ad-hoc networks?" in *MobiHoc*, 2009.
- [11] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," in *Technical Report CS-200006*, Duke University, 2000.
- [12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *the SIGCOMM workshop on Delay-tolerant networking*, 2005.
- [13] E. P. C. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward, "Practical routing in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 6, 2007.
- [14] Q. Yuan, I. Cardei, and J. Wu, "Predict and relay: An efficient routing in disruption-tolerant networks," in *Mobihoc*, 2009.
- [15] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Max-prop: Routing for vehicle-based disruption-tolerant networks," in *Infocomm*, 2006.
- [16] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *MobiHoc*, 2008.
- [17] T. N. Dinh, Y. Xuan, and M. T. Thai, "Towards social-aware routing in dynamic communication networks," in *the 28th IEEE International Performance Computing and Communications Conference (IPCCC)*, 2009.
- [18] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in *Mobihoc*, 2009.
- [19] F. Li and J. Wu, "Localcom: A community-based epidemic forwarding scheme in disruption-tolerant networks," in *SECON*, 2009.
- [20] J. Niu, X. Zhou, K. Wang, and J. Ma, "A data transmission scheme for community-based opportunistic networks," in *WiCOM*, 2009.
- [21] L. Vu, K. Nahrstedt, S. Retika, and I. Gupta, "Joint bluetooth/wifi scanning framework for characterizing and leveraging people movement in university campus," in *MSWiM*, 2010.
- [22] "Relational algebra." http://en.wikipedia.org/wiki/relational_algebra.
- [23] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Mobihoc*, 2007.
- [24] S. Gaito, E. Pagani, and G. P. Rossi, "Opportunistic forwarding in workplaces," in *ACM WOSN*, 2009.