RENA: Region-based Routing in Intermittently Connected Mobile Network

Hao Wen, Jia Liu, Chuang Lin, Fengyuan Ren
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China
wenhao, jliu, clin, renfy@csnet1.cs.tsinghua.edu.cn

Pan Li
Department of Electrical and Computer Engineering
Mississippi State University, Starkville, MS 39762, USA
li@ece.msstate.edu

Yuguang Fang
Department of Electrical and Computer Engineering
University of Florida, Florida 32611-6130, USA
fang@ufl.edu

ABSTRACT

Considering the constraints brought by mobility and resources, it is important for routing protocols to efficiently deliver data in Intermittently Connected Mobile Network (ICMN). Different from previous works that use the knowledge of previous encounters to predict the future contact, we propose a storage-friendly REgioN-bAsed protocol, namely, RENA, in this paper. Instead of using temporal information, RENA builds routing tables based on regional movement history, which avoids excessive storage for tracking encounter history. We validate the generality of RENA through time-variant community mobility model with parameters extracted from the MIT WLAN trace, and the vehicular network based on 8 bus routes of the city of Helsinki. The comprehensive simulation results show that RENA is not only storage-friendly but also more efficient than the epidemic routing, the restricted replication protocol SNW and the encounter-based protocol RAPID under various conditions.

Categories and Subject Descriptors

C.2 [**Computer Communication Network**]: Network Protocols—*Routing protocols*

General Terms

Algorithms, Design, Performance

Keywords

DTN, ICMN, Routing, Storage-friendly

MSWiM'09, October 26–29, 2009, Tenerife, Canary Islands, Spain.

1. INTRODUCTION

As a kind of challenged networks, Intermittently Connected Mobile Network (ICMN) is a delay-tolerant ad-hoc network that is made up of mobile nodes. Typical examples include ZebraNet [14], Vehicular Ad Hoc Network (VANET) [1], and Pocket Switched Network (PSN) [12]. In these networks, there are usually no supporting infrastructure (such as mesh nodes or base stations [22], [2]) for routing. Due to frequent network disruption and intermittent connections in ICMN, it is difficult to establish reliable end-to-end paths between mobile peers. Instead, the approach of Delay-Tolerant Networking (DTN) [13], which builds upon carry-and-forward between mobile nodes, is being widely applied to enable communication in such mobile environments. In addition, the constraints brought by mobility and resources in ICMN make the routing problem much more challenging, especially for resource-constrained devices such as sensor nodes or Bluetooth devices. Therefore, it is important for routing protocols to efficiently deliver data.

Previous work [13] suggests the benefit of gaining more knowledge about network conditions for routing decisions. Most DTN routing protocols use knowledge of previous encounters to forecast the future contacts [14], [5], [3], [1]. In these works, encounter history, a kind of *temporal* information, is taken as guidance to predict mobility: every node records other users when they encounter. For example, in the powerful RAPID protocol [1], it even need record the history of every forwarding operation to provide a more precise prediction. However, it costs excessive storage to track encounter history, which make RAPID suffer from degraded performance in storage-limited conditions.

Besides temporal information, it is also observed that the long-run trends in mobility widely reveal some spatial-related characteristics [7], [15], [9], [11]. Many WLAN traces indicate that the mobility of users demonstrates *location-preference* to some extent: users usually visit only parts of locations, such as restaurants or office buildings, and stay there for long periods. Meanwhile, they will commute back and forth among these locations resulting in *periodic re-appearance*. These traces also indicate *heterogeneity* in mobility, e.g., some nodes visit popular locations more often than others.

Although some mobility models are derived from these observations, there is little work to make use of spatial properties for efficient routing in ICMN. Since it is not cost effective to build routing tables based on encounter history in highly dynamic mobile network, a reasonable approach is to maximize the benefit of *macro-level* spatial information without having to constantly track and update encounter information.

In this paper, we propose a storage-friendly REgioN-bAsed protocol, i.e., RENA, in ICMN. Instead of using encounter information, RENA builds routing tables based on regional movement history, which avoids excessive storage for recoding encounter history. More specifically, this paper makes the following contributions:

1. We introduce a novel concept of *region-related* pattern extracted from the spatial property and take advantage of this macro-mobility information. To the best of our knowledge, RENA is the first region-based routing protocol proposed in delay-tolerant ICMN under realistic resource constraints.

2. Considering the tradeoff between the control packets and the storage consumption, we emphasize the necessity to design a *storage-friendly* protocol. Our experiments demonstrate that the delivery performance of RAPID [1] will significantly be degraded in resource constrained conditions if we consider storage occupied by control packets, while RENA achieves more stable performance since it only requires fixed storage for control packets.

3. We validate the generality of RENA through two realistic simulation scenarios: time-variant community mobility model, whose parameters are extracted from MIT WLAN trace, and vehicular network based on 8 bus lines of the city of Helsinki.

The rest of this paper is organized as follows: We present the related work in Section 2. A region-based mobility pattern is proposed in Section 3. Then Section 4 describes the design details of the region-based routing protocol RENA. Simulation results are presented in Section 5. Finally, we discuss the feasibility of RENA and conclude the paper.

2. RELATED WORK

Epidemic routing is a simple and fast way to perform routing throughout the network [21]. However, it is extremely wasteful of limited resources, such as wireless bandwidth and storage space. An approach to reduce the overhead of epidemic routing is to distribute a bounded number of copies [19]. As they do not make use of gaining knowledge about network conditions, their performance is not satisfying under more realistic mobilities [1].

Since node contact/encounter patterns represent the temporal characteristics of the node mobility, many DTN routing protocols exploit past knowledge of encounters to optimize the delivery performance. They are differentiated from each other mainly by the type of history information chosen: the last encounter time [5], [3] ,[20], the encounter frequency [14], or the whole encounter history [1].

However, these protocols face a fundamental challenge of choosing the right time scale for estimation [16]. Their performance may be influenced by the power-law decay of inter contact times between mobile nodes [15], [4] or the "distance effect" [5]. Therefore, the long-run trends inherited in the mobility, such as spatial features, are difficult to capture from a short-scale temporal view. Even for the powerful protocol RAPID [1] that requires almost all contact history for estimation, its performance significantly deteriorates due to the expansion of history information.

To understand mobility empirically, there are many recent works on WLAN measurements which reveal the important spatial properties of the real-world wireless network users [7], [15], [9], [11]. Although these research works are mainly obtained from the deployed wireless access networks with fixed infrastructure, they do imply that spatial properties should be taken into account during the design of practical routing protocols.

MobySpace is perhaps the first routing protocol to make use of locations of landmarks (e.g., access point) to construct a virtual space [17]. Here, the mobility pattern of a node can be described by its visits to locations. However, their work mainly focuses on Power-law probability distributions, and assumes that nodes have full knowledge of mobility patterns and infinite resources, which is unpractical in realistic applications.

Recently, several social-based routing protocols are proposed [12], [10], [8]. In essence, temporal or spatial information is a kind of representation of the social context. BUBBLE, which is proposed in PSN, calculates node popularity and social community based on *encounter history* [12]. They evaluate the impact of community and centrality on forwarding, and propose a hybrid algorithm that selects high centrality nodes and community members of destination as relays.

3. REGION-BASED MOBILITY PATTERN

Our solution is motivated from a simple observation: location-preference, re-appearance and heterogeneity are usually observed as typical features not only from human beings but also from other species (such as Zebra) [14]. People or animals usually visit parts of locations, stay there for a period of time, then commute back and forth. The encounter history is effective to predict next encounter in a specific area. However, when users change their geographic areas, the encounter information obtained at previous locations may be no longer accurate as the guidance for new locations.

For example, students studying in the same building are likely to encounter each other frequently at study time. After class, they may go to different dormitories. It is difficult to estimate who will encounter again at new locations if only based on encounter history in the building, since two students study in the same class do not necessarily mean they live in the same dormitory. Although the statistical tracking information at a micro-level (i.e., on small time scales or within small distances) may be still useful, it is necessary to take the macro-level information into account in a realistic scenario.

In this section, we propose a region-based mobility framework to explain the macro-level abstraction. Then two important parameters are calculated for further protocol design.

3.1 Mobility Model

We consider m regions as mobility domain in the figure, enumerated as 0, 1, $\cdots$, m-2, m-1. A node (we use the terms "node", "user" and "peer" interchangeably to describe a mobile network user in this paper) moves according to a transition matrix $\mathbf{P}$: It moves from region i to another region j with probability p_{ij} or remain in the same region with probability p_{ii}. The steady-state probability at region i is π_i. Then this mobility model can be thought of as a discrete-time Markov chain where a possible state transition is made every τ unit time of a period.

3.2 Hitting time

We first consider the expected hitting time HT of a node in region i arrives at region j for the first time. In order to obtain this parameter, we make state j an absorbing state and other states transient states, then reset the transition probabilities from state j as

$$p_{jj} = 1, p_{ji} = 0(i \neq j). \quad (1)$$

Now we can extract the matrix $\mathbf{P_T}$ that specifies only the transition probabilities from transient states into transient states. If we define s_{ik} to be the expected number of time periods that the Markov chain is in region k, given that it starts in region i, the matrix $\mathbf{S}$ can be expressed as

$$\mathbf{S} = (\mathbf{I} - \mathbf{P}_T)^{-1}. \quad (2)$$

More details about calculating matrix $\mathbf{P_T}$ and $\mathbf{S}$ could be found in [18].

Since the expected time to enter in absorbing state j from transient state i is equal to the expected time spent in all transient states before enters in state j from transient state i, we can get HT as

$$HT_{ij} = \tau \sum_{k \neq j} s_{ik}. \quad (3)$$

3.3 Return time

Next consider another important metric: the expected return time RT of a node back to a region j after it leaves region j. In fact, RT_j is equal to the expected hitting time to region j from all j's adjacent regions:

$$RT_j = \sum_{k \neq j} \frac{p_{jk}}{1 - p_{jj}} HT_{kj}. \quad (4)$$

4. PROTOCOL DESIGN

In this section, we will present the design details for our region-based protocol RENA. In the following discussion, we assume that nodes are equipped with either GPS or other localization methods to obtain a geographic location. Note that we do not require accurate location that is necessary for geographic routing protocols.

4.1 Protocol Overview

The main idea of RENA is to find the popular regions for destination node and distribute copies of packets inside those popular regions. However, we need define region firstly. In this paper, we simply adopt a square coverage in the movement area as a region. Then the whole area is divided to m equal-sized grid cells shown as in Fig. 1 (m=16).

Given source node x and destination node y, RENA chooses candidate regions for y in terms of estimated delivery delay $D_{ij}(x, y)$, which includes the routing time $R_{ij}(x)$ from current region i of node x to candidate region j and the waiting time $W_j(y)$ of the relay node before encountering node y inside the region j:

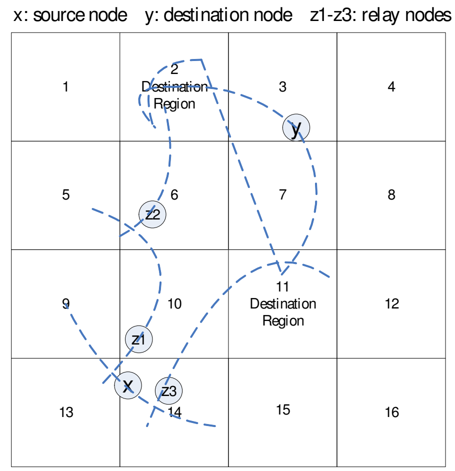

Figure 1: RENA overview

$$D_{ij}(x, y) = R_{ij}(x) + W_j(y), \quad (5)$$

where $R_{ij}(x)$ and $W_j(y)$ are both derived from the transition matrix (see Sec. 4.3 for details).

In this paper, we choose delivery delay, not encounter probability, as an optimized metric. The reason for this is that most existing probability-driven approaches have an incidental rather than an intentional effect on delay related metrics [1]. For example in Fig. 1, selected region 2 is the most popular for node y but far from current location of source node x (i.e., region 14), while region 11 is not frequently visited by node y but may be still a feasible choice in terms of shorter routing time $R_{ij}(x)$.

When L regions are chosen as the candidate destinations (Sec. 4.3), the source node will send one copy to every chosen region, respectively. The relay node is chosen according to its expected hitting time HT_{kj} from the current region k to destination region j: the node that has smaller hitting time to destination region j would be a better relay node. Thus in Fig. 1, when relay node z_1 determines that z_2 has a smaller hitting time to region 2, z_2 is chosen as a new rely node and z_1 will *delete* the copy.

After the relay node reaches one of the destination regions, e.g., z_3 reaches region 11 in Fig. 1, it will trigger a spray-and-search distribution: First, the relay node will start a binary spray to W-1 nodes in the spray phase; Second in the search phase, W nodes carrying copies will only forward their own copies to those with a smaller return time RT_j until encountering the destination node.

Essentially, RENA forwards L copies to L destination regions and distributes W copies inside every chosen region. Then the total number of copies of packets is bounded by LW. Since replication strategy and storage management policy are carefully designed based on spatial property, RENA is a storage-friendly protocol that limits replicating only inside selected regions instead of the whole area.

Table 1: Region transition matrix for node y

Region	2	3	7	11
2	2	0	1	0
3	1	0	0	0
7	0	1	0	1
11	0	0	1	0

Table 2: Transition probability matrix for node y

Region	2	3	7	11
2	0.67	0	0.33	0
3	1	0	0	0
7	0	0.5	0	0.5
11	0	0	1	0

4.2 Region transition matrix

Since we assume that every node has a coarse geographic location, it can obtain the current region where it locates. Every τ unit time, it will track the current region and record into region transition matrix $\mathbf{B}$, where b_{ij} indicates transfer times from region i to region j. For example, if tracking history of node y is 3-2-2-2-7-11-7-3, its region transition matrix is shown as Table 1. Here, node y does not record the sequence, and only update corresponding b_{ij}.

According to region transition matrix $\mathbf{B}$, we can obtain transition probability matrix $\mathbf{P}$ as

$$p_{ij} = b_{ij} / \sum_{j=1}^{m} b_{ij}. \tag{6}$$

For example, the corresponding transition probability matrix $\mathbf{P}$ of Table 1 is Table 2.

Then according to calculation in 3.1, we could obtain hitting time HT_{ij} of node y from region i to region j and its return time RT_j to region j. Then nodes will flood their return time matrix $\mathbf{RT}$ in the whole network. Since RT is only related to the number of regions m, this flooding transmission only consumes limited storage and bandwidth. Different from RT, only hitting time values HT related to current region will be exchanged between encounter nodes (see Sec. 4.6).

4.3 Choosing destination regions

When source node x chooses candidate regions for destination node y, it is better to find proper ones that not only are frequently visited by y but also could be quickly reached from the current location of x. However, without global network knowledge, it is almost impossible to give an accurate prediction in a delay-tolerant mobile network. Instead, RENA operates based on a coarse but an efficient estimation of routing time and waiting time.

In terms of the routing time $R_{ij}(x)$ from current region i of node x to candidate region j, the source node builds *a local view* from its historical encounter nodes and itself:

$$R_{ij}(x) = \frac{HT_{ij}(x) + \sum_{i=1}^{en} HT_{ij}(z_i)}{en+1}, \tag{7}$$

where node z_i indicates the i-th encounter node of x whose HT_{ij} is not equal to infinity and en means encounter times between x and other nodes. Here the same encounter node may be repeatedly calculated, i.e., $z_i = z_j$ when $i \neq j$, because we argue that frequently encounter nodes should have greater weight to the local view of a source node.

In terms of the waiting time before encountering destination node inside the region j, we simply take the return time of destination node as estimation value, namely,

$$W_j(y) = RT_j(y) \tag{8}$$

Based on (7) and (8), source node x can obtain estimated delay to destination node y:

$$D_{ij}(x,y) = R_{ij}(x) + W_j(y) \tag{9}$$

Then we arrange estimated delays of all regions in ascending order as:

$$D'_{i1}(x,y) \le ... \le D'_{ik}(x,y) \le ... \le D'_{im}(x,y) \tag{10}$$

To avoid unnecessary storage consumption and propagation overhead, we choose L destination regions or copies based on the Time-To-Live (TTL) for packets. Since the actual delivery time $D_{ij}(x,y)$ distributions are hard to obtain, we here simply assume that they are exponentially distributed. Then the minimum L are selected according to

$$\left(\sum_{k=1}^{L} \frac{1}{D'_{ik}(x,y)}\right)^{-1} \le \mathrm{TTL} \tag{11}$$

Here in (11), we have used the following result: When the encounter times between destination nodes and different L copies are exponentially distributed with means $\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, ..., \frac{1}{\lambda_L}$, respectively, the expected minimum delivery delay that destination node encounters one of the copies is

$$\frac{1}{\lambda} = \frac{1}{\lambda_1 + ... + \lambda_L} = \left(\sum_{k=1}^{L} \lambda_k\right)^{-1} \tag{12}$$

Note that we assume an exponential distribution of delivery time since realistic motilities are very difficult to model. However, simulation results show that this tractable simplification performs well in different realistic scenarios.

4.4 Choosing relay nodes

After L destinations regions are decided, the next step is to find relay nodes to forward these L copies to chosen regions. Relay nodes are chosen according to the expected hitting time HT_{kj} from the current region k to destination region j:

1. When two nodes encounter in the region k ($k \neq j$), they will exchange and compare their hitting time values HT_{kj}.

2. Copies will be *forwarded* to the node that has a smaller HT_{kj}. The other node that has a bigger HT_{kj} *does not* keep the copy anymore.

4.5 Spray and search

When the relay node z that carries copies reaches the destination region j, it will trigger a Spray-And-Search (SAS) distribution in order to improve the encounter probability with destination node y:

1. Spray phase: the relay node z will start a binary replication of W copies to distinct relay nodes. When any node z_1 that has $c > 1$ copies encounters another node z_2 without those copies, z_1 hands over to z_2 $\frac{c}{2}$ copies and keeps half of them. This process will continue until $c = 1$ or destination node y is encountered.

2. Search phase: when any one of W nodes z_3 carrying one copy encounters another node z_4 without that copy, it will forward its own copy to z_4 if z_4 has a smaller return time RT_j.

Here, we limit the number of copies inside regions W to control the total number of copies $L \times W$. The spray phase aims at "jump-start" spreading in a quick manner and the search phase works to find better relays that are more likely to stay inside region j. Note that the SAS only performs in the destination region j to avoid burdening propagation in the whole area. Therefore, when any relay node in the process moves out of region j, its SAS will suspend. The SAS will resume when related copies are relayed back (maybe by new relay nodes) to region j later.

4.6 Updating policy

Since it is almost impractical to obtain the global system state, RENA chooses to exploit the value of delayed and inaccurate estimation. Although it is helpful for routing decision with more knowledge of network conditions, the routing protocol should consider the tradeoff between performance gain and system overhead, thereby update control information in an efficient manner. RENA achieves this goal by propagating only necessary control packets.

In RENA, every node updates region transition matrix every τ unit time, but only updates transition probability matrix, hitting time matrix and return time matrix when encounter events happen. For efficiency, a node z_1 executes the following operations when encountering a peer z_2 in region k:

1. Directly forwards data destined to z_2.

2. Exchanges ACKs of delivery packets that are not beyond TTL and delete useless copies according to ACKs.

3. Exchanges the returning time matrix **RT** of its historical encounter nodes and itself.

4. Exchanges the hitting time values HT_{kj} from current region k to other regions.

5. Forwards copies to z_2 if z_2 is a better relay node as discussed in Sec. 4.4.

6. Execute the SAS distribution if necessary.

Here ACKs are used to avoid the unnecessary overheads and the information of third and fourth steps are the decision-making basis for the last two steps. Although the updating information may be stale in such a dynamic network, this distributed policy proves to be efficient through comprehensive simulations.

4.7 Storage management

For the multi-copy replication mechanism, it is crucial to design storage policy to ensure that the total storage consumption is small and controlled. RENA generates only a small, carefully chosen number of LW copies (every W copies distributed in one selected region). Besides data packets and ACKs, node in RENA mainly records returning time matrix **RT** of all users and its own hitting time matrix **HT**. In our simulation, the total number of copies LW is restricted to 10% of all n nodes. The results show that average storage per region consumed by control data in RENA is about 650 Bytes.

In addition, we manage packets according to their redundancy and importance degree:

1. Acknowledged packets are immediately deleted.

2. Every τ unit time, remove packets and ACKs that have reached the threshold of TTL.

3. If the storage is close to overflow state, copies in the search phase are firstly dropped followed by copies in the spray phase. Then we further drop the copies that have not reached destination region if still necessary.

5. EVALUATION

In this section, we conduct simulation study to evaluate our RENA and other protocols.

5.1 Protocol Comparisons

We compare RENA to three other routing protocols: Epidemic routing [21], Spray and Wait [19], and RAPID [1]. A brief review of each algorithm is given below.

Epidemic routing [21]: Whenever a node encounters another node, the two nodes exchange all packets that they do not have in common.

Spray and Wait (SNW) [19]: It distributes only a limited number of copies each to a different relay. Each copy is then carried all the way to the destination node by the designated relay. It restricts the number of replications to be equal to about 10%-15% of all n nodes.

RAPID [1]: It treats DTN routing as a resource allocation problem that translates the routing metric into per-packet utilities which determine how packets should be replicated. In order to estimate the utility given the routing metric, the exchanged control packets include the quantity and location of replicas of a packet and the average size of past transfers. As far as we know, results in [1] show that RAPID is one of the most powerful protocols in terms of both delivery rate and delivery delay.

To evaluate the effect of control packets, we also provide the results if we only count the storage consumed by data packets and set *unlimited buffer* for control packets. Note that these additional results are only used to *verify storage friendliness* of control packets since the assumption of unlimited buffer is not practical in reality.

5.2 Experiment setups

We implement four routing protocols using the Opportunistic Network Environment (ONE) simulator [6]. We compare our protocol with other routing algorithms in terms of delivery rate and delay with variations of TTL, storage capacity, and the number of nodes. The simulation time is set 12 hours.

Table 3: Main parameters of two scenarios

Item	Range	Default value
TTL	30-180 minutes	180 minutes
Storage	50-300 KB	100 KB
Packet generating interval	20-50 s	30s
The number of nodes (TVC model)	10-100	50
The number of nodes (Bus model)	8-80	40

For connectivity, we assume interpersonal communication (10m range, 2 Mbps) between mobile users using Bluetooth devices, such as mobile phones. Every t interval time, one mobile user generates one packet of 1 KB to a random destination.

For fairness of comparison, the numbers of copies of RENA and SNW are both restricted to *10%* of all n nodes. All nodes in RENA record the current region every 5 seconds. Other parameters are given in the specific mobility model.

5.3 Scenario 1: Time-variant Community Mobility Model

Time-Variant Community mobility (TVC) model is proposed to capture spatial and temporal correlations of realistic mobility [11]. It defines communities to serve as popular locations for the nodes, and implement time periods in which the nodes move differently to induce periodic behavior. To create a more realistic scenario, we adopt *trace-driven* TVC model, referred to as model 1 in [11], which displays similar behavior to the MIT WLAN trace in terms of the hitting and the return time. The communities in this model are essentially randomly determined and not fixed, which make nodes actually move around the whole simulation area.

In this model, edge length of simulation area is 1000 meters with community size of 100×100 m^2 and edge length of region is chosen 250 meters. Then the number of regions is 16. Main simulation parameters are listed in Table 3. The default values are adopted if not mentioned. We do not list other mobility-related parameters of TVC model here, which can be easily obtained from [11].

5.4 Scenario 2: Bus Mobility Model

We also choose a typical vehicular delay-tolerant network based on the city area of Helsinki with 10000×10000 m^2. The whole city area has 4 main districts. Additionally, three overlapping districts are defined to simulate movements between the center A and other districts, and one district to cover the whole simulation area. Every district is assigned its own bus route, respectively.

Total n buses are evenly distributed in this 8 bus routes and travel across these predefined bus routes according to a periodic schedule ([6]). The buses move at 7-10m/s with a 10-30s waiting time at each bus stop. RENA generates 16 regions with unit size 2500×2500 m^2. Main simulation parameters are listed in Table 3 and the default values are adopted if not specified.

5.5 Results

5.5.1 The Impact of TTL

Figure 2 shows the delivery rate of each protocol as TTL for data packets increase from 30 to 180 minutes in scenario 1. With this increase in TTL, we can see a decrease in delivery rate only for epidemic routing. What is striking is that RAPID-ODP (Only Data Packets are restricted by offered storage) significantly outperforms RAPID: It demonstrates that the storage consumption of control packets largely affects the performance of RAPID, since its control packets almost occupy more than 98% of storage in simulation. Although RAPID can achieve satisfactory performance in the DieselNet with 40 GB of storage [1], where control packets are comparatively small enough to ignore, we conclude that RAPID may be not a good choice in resource-constrained devices, e.g., sensor nodes or Bluetooth devices.

In contrast with RAPID, RENA only needs a limited storage (11 KB) for control packets and achieves almost the same delivery rate with RENA-ODP. More importantly, it performs better than SNW and RAPID in a stable manner when TTL increases, and its delivery rate is even within 7% of RAPID-ODP when TTL is bigger than 90.

Figure 3 shows the average delay of the delivered packets of each protocol in scenario 1. It is observed that RAPID and RAPID-ODP have smaller delay than others as a result of more accurate estimation. RENA either achieves almost the same delay with SNW but bigger delivery rate ($TTL < 150$), or achieves smaller delay than SNW and similar delivery rate ($TTL \geq 150$). Note that the average delay is calculated from the delivered packet so we need consider both average delay and delivery rate to make a fair comparison.

Figure 4 and 5 show the delivery performance as TTL for data packets increases in bus network scenario. Compared with TVC model scenario, we observe similar trends in terms of both performance metrics. However, SNW achieves much lower delivery rate than RENA and RAPID-ODP: even for 180 minutes of TTL time, the delivery rate of SNW is only 0.561. The reason is that SNW uses random distribution and passive waiting which are not effective in large-scale group mobility situation. In contrast, RENA and RAPID-ODP make use of broadcasting control packets to construct a local predication and active transmission.

5.5.2 The Impact of Storage

Figures 6 and 7 show results as the storage increases from 50 to 300 KB in scenario 1. It is observed that RAPID and epidemic routing are both sensitive to storage restriction: they gets higher delivery rate with the increase of storage. Again, a significant gap between RAPID and RAPID-ODP is observed. Compared with RAPID and epidemic routing, RENA demonstrates its storage friendliness through a stable performance in different storage limitations. It is more notable that RENA achieves much smaller, at least 15%,

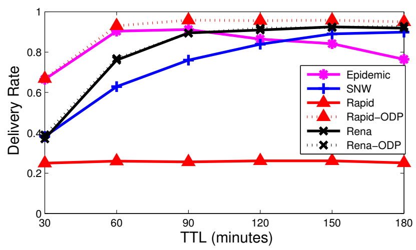

Figure 2: TVC model - Delivery rate of (TTL)

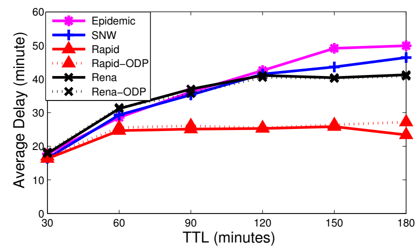

Figure 3: TVC model - Average delay (TTL)

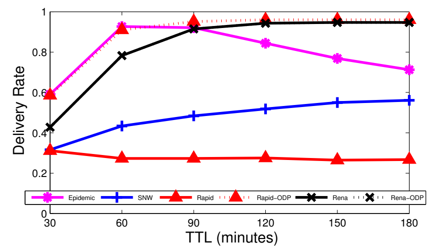

Figure 4: Bus model - Delivery rate (TTL)

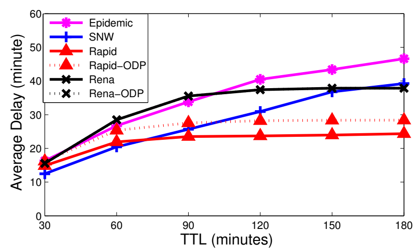

Figure 5: Bus model - Average delay (TTL)

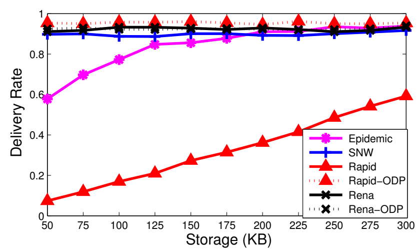

Figure 6: TVC model - Delivery rate (Storage)

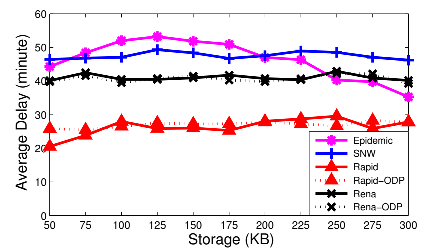

Figure 7: TVC model - Average delay (Storage)

delay than SNW while it obtains higher reliability.

Figures 8 and 9 display again the storage friendliness of RENA and the storage sensitivity of RAPID and epidemic routing. Since we have 16 regions in this setup, the control packets totally consume about 10K storage. Only in the 50k storage condition, there is an obvious difference between RENA and RENA-ODP. Although SNW is also storage friendly, its performance is much worse than RENA due to its passive waiting. Different from RENA and SNW, RAPID and epidemic routing are markedly sensitive to storage restriction.

We also observe an interesting event that the results of RENA, SNW and RAPID-ODP are more stable than that in scenario 1: they obtain almost the same results under various storage conditions except RENA in the 50k storage condition, probably due to the periodicity in the bus network.

5.5.3 *The Impact of Traffic Load*

When we vary packet generating interval time from 20s (high traffic) to 50s (low traffic) in both scenarios, it is observed that epidemic routing and RAPID are obviously sensitive to traffic load while RENA and SNW keep providing a stable performance. Again, RENA not only achieves delivery rate even close to RAPID-ODP but also smaller delivery delay than SNW and epidemic routing, while the delivery rate of RAPID still has significant difference from RAPID-ODP.

5.6 Non-Markov Mobility

Although RENA is proposed based on Markov mobility model, it could achieve feasibility in most scenarios. When motilities of users do not follow Markov property, RENA could still work well if there are spatial-related characteristics that widely exist in real-life motility. Even in the extreme case without any location-preference property, such as random mobility models, RENA could naturally transfer to SNW: when there are no candidate destination regions to choose, RENA simply chooses its current region as destination and immediately starts binary spray. In other words, the delivery performance of SNW is *the lower bound* of RENA.

6. CONCLUSION

In this paper, we introduce a novel concept of region-related pattern extracted from the spatial property in ICMN. Then we take advantage of this macro-mobility information and propose a region-based routing protocol RENA. Compared with epidemic routing and restricted replication protocol SNW, RENA achieves not only higher delivery rate but also shorter delivery delay only at the cost of a relatively low storage consumed by control packets. In terms of storage and feasibility, RENA is more storage-friendly and adaptive than encounter-based RAPID. Therefore, we conclude that RENA may be a better choice for resource-constrained devices in ICMN.

7. ACKNOWLEDGMENTS

This work is supported in part by the National Grand Fundamental Research 973 Program of China under Grant No. 2006CB303000 and 2010CB328105; the National Natu-

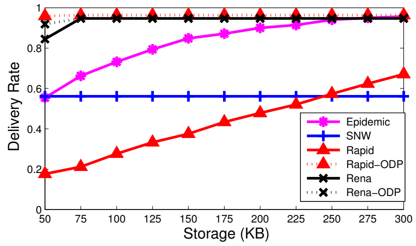

Figure 8: Bus model - Delivery rate (Storage)

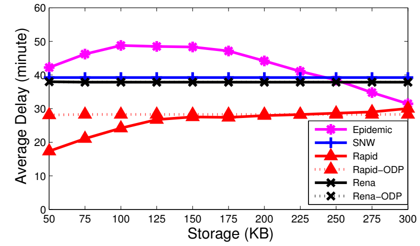

Figure 9: Bus model - Average delay (Storage)

ral Science Foundation of China under Grant No. 60673187, 60872055 and 60773138. The first author also wish to thank Chinese Government Graduate Student Overseas Study Program provided by CSC, and students in Wireless Networks Laboratory of University of Florida.

8. REFERENCES

[1] A. Balasubramanian, B. N. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. In *ACM Sigcomm*, Japan, Jan. 2007.

[2] N. Banerjee, M. D. Corner, D. F. Towsley, and B. N. Levine. Relays, base stations, and meshes: enhancing mobile networks with infrastructure. In *Proc. MOBICOM 2008*, pages 81–91, California, Sept. 2008.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE INFOCOM*, Spain, 2006.

[4] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *Proc. IEEE INFOCOM*, Spain, Apr. 2006.

[5] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *Proc. ACM MobiHoc*, pages 257–266, MD, USA, June 2003.

[6] F. Ekman, A. Keranen, J. Karvo, and J. Ott. Working day movement model. In *SIGMOBILE Mobility Models*, Hong Kong, May 2008.

[7] J. Ghosh, M. Beal, H. Ngo, and C. Qiao. On profiling mobility and predicting locations of campus-wide wireless network users. In *Proc. REALMAN*, pages 55–62, Italy, May 2006.

[8] J. Ghosh, S. J. Philip, and C. Qiao. Sociological orbit aware location approximation and routing (solar) in manet. In *Proc. IEEE Broadnets*, pages 641–650, Boston, USA, Oct. 2005.

[9] W. Hsu, D. Dutta, and A. Helmy. Extended abstract: Mining behavioral groups in large wireless lans. In *Proc. ACM MOBICOM*, pages 338–341, Canada, Sept. 2007.

[10] W. Hsu, D. Dutta, and A. Helmy. Profile-cast: behavior-aware mobile networking. *ACM SIGMOBILE Mobile Computing and Communications Review*, 12:52–54, Jan. 2008.

[11] W. Hsu, A. Spyropoulos, K. Psounis, and A. Helmy. Modeling time-variant user mobility in wireless mobile networks. In *Proc. IEEE Infocom*, Alaska, May 2007.

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

[13] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant networking. In *Proc. SIGCOMM*, pages 145–158, Portland, USA, Jan. 2004.

[14] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS-X*, pages 96–107, USA, 2002.

[15] T. Karagiannis, J. L. Boudec, and M. Vojnovic. Power law and exponential decay of inter contact times between mobile devices. In *Proc. ACM MOBICOM*, pages 183–194, Canada, Sept. 2007.

[16] J. Karvo and J. Ott. Time scales and delay-tolerant routing protocols. In *Proc. ACM Chants Workshop*, pages 33–40, California, USA, Sept. 2008.

[17] J. Leguay, T. Friedman, and V. Conan. Dtn routing in a mobility pattern space. In *Proc. ACM Chants Workshop*, pages 276–283, USA, Aug. 2005.

[18] S. M. Ross. *Introduction to Probability Models*. Academic Press, Orlando, FL, 2006.

[19] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proc. ACM WDTN*, pages 252–259, USA, Aug. 2005.

[20] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Trans. Netw.*, 16:77–90, Feb. 2008.

[21] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke, 2000.

[22] W. Zhao, Y. Chen, M. Ammar, M. D. Corner, B. N. Levine, and E. Zegura. Capacity enhancement using throwboxes in dtns. In *Proc. IEEE MASS*, pages 31–40, Vancouver, Canada, Oct. 2006.