CPCR: Contact-Prediction Clustering-based Routing in Large-scale Urban Delay Tolerant Networks

Wenjing Yang
School of Software
Beihang University
Beijing, P.R.China
Email: yangwenjing@sse.buaa.edu.cn

Haiquan Wang
School of Software
Beihang University
Beijing, P.R.China
Beijing Key Laboratory of
Network Technology Beijing, P.R.China
Email: whq@buaa.edu.cn

Jingtao Zhang
and Jiejie Zhao
School of Software
Beihang University
Beijing, P.R.China
Email: zjt@buaa.edu.cn
Email: zjj@buaa.edu.cn

***Abstract*—As increasing of the network scale, hierarchical routing methods perform better in scalability compared with flat strategies. However, maintaining the groups based on large amount of historical information is costly in a large dynamic network. We propose a hierarchical clustering-based routing method CPCR, which is decentralized and only makes use of contact information exchanges between nodes, to achieve scalability and limit resource consumption in large-scale urban Delay Tolerant Networks (DTNs). Via analyzing a large taxi trace data in Beijing, we discovery how to use an exponential distribution to model the contact characteristics between nodes and how to measure the similarity among such distributions. With such information, CPCR can obtain the contact probability in real-time for each nodal pair and ensure that nodes in the same cluster have higher contact strength during clustering. Accordingly, in CPCR, intra-cluster routing conducts via a direct delivery, while inter-cluster routing chooses to flood to the remaining clusters or only relays to the destination cluster adaptively. Our simulation results over the Beijing taxi traces show that the proposed method can achieve the efficacy of clustering and good scalability of routing.**

***Keywords*—DTN routing , clustering-based routing, contact distribution , inter-contact time , delay tolerant networks**

I. Introduction

Delay Tolerant Networks (DTNs) [**?**], [**?**], [**?**], [**?**] is a branch of networks, where communication links only exist temporarily with a highly dynamic topology. One of such examples is vehicular network where vehicles communicate with each other in order to disseminate data using opportunistic contacts [**?**], [**?**], [**?**], [**?**], [**?**], [**?**]. To communicate in such an challenging environments, numerous DTN routing methods have been proposed over the last decade.

Traditional flat routing methods, such as *Direct Delivery* (DD) [**?**] and *Epidemic Routing* [**?**], become not scalable to large-scale DTNs. Meanwhile, clustering-based approaches have long been considered as an effective approach to reduce network overhead and improve scalability in traditional mobile ad hoc networks [**?**], [**?**], [**?**], [**?**] with relatively stable topology and more communication opportunity. Clustering in DTNs is unique [**?**], [**?**], because the network topology is not always fully connected. As a result, it becomes much more challenging to acquire necessary information to formulate clusters and ensure their stability. When we consider an urban DTN, the nodal scale and mobility will further increase the cost of maintaining clusters.

Contact characteristics among nodes [**?**], [**?**], [**?**] have been used as one of key metrics for clustering in DTNs. A contact happens between two nodes when they appear in each other's communication range, which refers to a possible communication opportunity. Frequency and duration of historical contacts have been used as metrics to describe contact probability [**?**], [**?**], however they cannot provide very accurate estimation of the contact probability among nodes. Instead, the distribution of *inter-contact time* (ICT) has been shown to be powerful for describing contact characteristics [**?**], [**?**], [**?**]. ICT between two nodes directly affects the transmission delay between these two nodes, since a shorter ICT implies a greater opportunity to relay packet in a period of time.

In this paper, we consider how to design an efficient clustering-based routing scheme for large-scale urban DTNs. We first analyze a real-world urban vehicular tracing dataset, GPS traces from $12,096$ Beijing taxis in one week, to reveal the Inter-contact time distributions of mobile vehicles in a large city. Our study confirm that both the global ICT among all nodes and the individual ICTs between each nodal pair follow the exponential distributions. We demonstrate that the parameter estimation for ICTs of nodal pairs is more accuracy than the parameter fitting the global ICT distribution. Therefore, we design distributed clustering algorithm which groups nodes by the contact probability calculated from the ICT distribution for nodal pair in real-time, and propose a new clustering-based routing scheme based on this contact predication clustering.

To summarize, our contributions in this paper can be highlighted as follows:

- Via an in-depth analysis of the Beijing taxi data set, we demonstrate that it is applicable to estimate the parameter for the ICT exponential distribution of every nodal pair. Such pair-wise ICT distribution is more accurate then the global ICT distribution, which is commonly used in several DTN routing schemes.
- A hierarchical routing algorithm, CPCR, based on a distributed clustering via contact prediction is proposed. CPCR uses direct delivery and flooding strategy in intra-cluster and inter-cluster routing phases,

respectively.

- Extensive simulations are conducted with the Beijing taxi tracing data to evaluate the efficiency and scalability of the proposed clustering-based routing.

The rest of paper is organized as follows: Section II provides an overview of related work on DTN routing. Section III provides the theoretical basis of contact predicting and clustering via detailed analysis of the Beijing taxi traces. Section IV presents our clustering-based routing algorithm which leverages the estimated ICT distributions to clustering nodes into groups and routes the packages accordingly. Simulation results over the Beijing taxi traces are presented in Section V. Finally, Section VI concludes this paper.

II. Related Work

In 2004, Jain et al. [**?**] first investigate the routing issues in DTNs, where messages are to be moved end-to-end across a connectivity graph that is time-varying. They assume that the dynamics of DTNs is known in advance. Since then routing in DTNs has been heavily studied [**?**], [**?**] over the last decade. The routing performance is influenced by topology partitioning in DTNs. Based on the routing topology, we can categorize existing routing methods into *flat routing* or *hierarchical routing*. There are other ways to classify DTN routing methods, for more detailed please refer to related surveys such as [**?**], [**?**].

Flat routing methods employ only a single routing strategy globally, while all nodes involved play similar roles. For example, *Epidemic Routing* [**?**] is a simple flooding method with zero knowledge of the network topology. Compared with some single-copy routing methods, such as *Direct Delivery* [**?**], *Randomized Routing* (RR) [**?**], Epidemic routing greatly improves delivery ratio but suffers from resource waste and even network congestion due to large data replication. In order to reduce the waste of resources, *Spray and Wait* routing [**?**] limits the amount of message copies in its spray phrase and lets node wait until contacted by the destination node in its wait phrase. But with such solutions, routing performance could still be poor in large-scale networks. Besides these flooding based schemes [**?**], [**?**], [**?**], there is also a set of DTN routing methods based on the contact information, where a smarter relay node selection is made. For example, *PRoPHET* [**?**] is a probability-based method, where each node maintains the encounter history with other nodes, and the routing decision is made based on the encounter probability. Similarly, *MaxProp* [**?**] records neighbor nodes and their contact times with the destination node so that a contact probability can be adopted as the rule to choose the next hop relay. By introducing knowledge oracles, this kind of routing methods can reduce resource consumption and improve delivery ratio as well. However, with the node scale expanding, these methods cost more storage and computation resources to maintain the global historical information of contacts. Hence, the scalability and performance of flat routing methods are limited, when the network scale increases and topology evolves all the time. In such scenario, flat routing method cannot ensure low delay and high delivery ratio, and the cost of requiring global knowledge sometime is too high to be efficient any more.

Hierarchical routing organizes the networks in groups via clustering techniques. It has been shown that hierarchical routing can achieve good scalability in large-scale mobile networks. However, restricted by the dynamic topology and frequently interruptions, hierarchical routing in DTNs is still in its early stages. Dang and Wu [**?**] propose a clustering based routing method based on online estimation of contact probabilities. Their clustering method can make sure that nodes within the same cluster have higher contact probability. Intra-cluster routing adopts directly delivery approach so that a node will not delivery packages until contacted by the destination node in the same cluster. For the intra-cluster routing, a relay mechanism similar with MaxProp is used. Liu and Wu [**?**] also design a hierarchical DTN routing algorithm based on a strict mobility model, where all nodes move according to strict repetitive patterns. Their method is only effective such a specific mobility model and could not adapt to other mobility pattern. Ahmed and Kanhere [**?**] adopt encounter frequencies of public transport networks to classify nodes. Nodes only relay messages to neighboring nodes which are in the same cluster of the destination node. Then epidemic routing is used to propagate messages inside the cluster. These clustering-based hierarchical routing methods can improve the scalability, but the resource consumption of collecting and maintaining historical information remains high. Whitbeck and Conan [**?**] also proposed a hybrid DTN-MANET routing (HYMAD) for dense and highly dynamic wireless networks. By periodically scanning of topology changes, HYMAD builds temporary disjoint groups of connected nodes as clusters. Then DTN approach is used between disjoint clusters while MANET routing is used within clusters. This method is decentralized and only makes use of topological information exchanges between the nodes, thus limit the storage and information exchange cost. However, the update cycle of the topology is crucial. If the cycle is too short, updating and maintaining the topology may be difficult. If the update cycle is too long, intra-cluster routing using MANET method may be invalid. Therefore, this method does not suitable for sparse DTNs.

In summary, the key challenge of hierarchical routing is the clustering method, especially in large scale DTNs. Efficient clustering method can form stable clusters, in which nodes are similar with each other, so that corresponding intra- and inter-cluster routing algorithms can be designed easily. Overall, with stable clustering hierarchical routing can improve the scalability and efficiency of routing in large scale DTNs.

III. Characteristics of Large-scale Urban DTNs

In this section, we mainly focus on revealing the inherent characteristics of contacts and possible clustering methods in large-scale urban DTNs by analyzing a real-world trace data. We believe that these are essential properties which affect DTN routing design and verification.

A. Trace Dataset: Beijing Taxi Traces

To study urban DTNs, we use a real-world GPS dataset which were generated by $12,096$ taxis in Beijing, China within one week (from June 13th to 19th, 2010). This dataset has been used by several key research and application programs of Intelligent Transportation Systems (ITS) in Beijing, China. The number of participated taxis ($12,096$ taxis) is 18% of

```
1378220 $JYJ 13301209475 20100619020107 116.424736 39.94133759 429211193 147244943 6 14 0 0 50#
1405539 $JYJ 13301209482 20100619020330 116.4188843 39.89599991 429189548 147077754 12 98 0 0 50#
1425910 $JYJ 13301214131 20100619020516 116.4018021 39.95977783 429126659 147312896 0 192 0 0 50#
1388608 $JYJ 13301214133 20100619020201 116.5081482 39.89743042 429517736 147082361 2 226 0 0 50#
1412095 $JYJ 13301214251 20100619020401 116.3109512 39.95031357 428791171 147277547 0 8 0 0 50#
1417843 $JYJ 13301214251 20100619020437 116.3108139 39.95037842 428790672 147277772 0 274 0 0 50#
1419486 $JYJ 13301214422 20100619020442 116.3469696 39.87646866 428924212 147005556 0 284 0 0 50#
1432012 $JYJ 13301214432 20100619020546 116.4670639 40.01408386 429366828 147512800 41 286 0 0 50#
1408123 $JYJ 13301214496 20100619020341 116.4458313 39.88336945 429288687 147031069 0 272 0 0 50#
1416259 $JYJ 13301214523 20100619020429 116.4451675 39.90390396 429286245 147106760 0 342 0 0 50#
1386793 $JYJ 13301214583 20100619020151 116.4356003 39.9047966 429251130 147110193 0 280 0 0 50#
1428516 $JYJ 13301214642 20100619020533 116.3618011 39.87991333 428979005 147018379 13 256 0 0 50#
1425962 $JYJ 13301214766 20100619020516 116.2493973 39.91648102 428564023 147152590 0 274 0 0 50#
1415042 $JYJ 13301214815 20100619020422 116.4914856 39.9526825 429456455 147286169 0 274 0 0 50#
1394361 $JYJ 13301251400 20100619020231 116.4216309 39.92809296 429199732 147196110 0 268 0 0 50#
1431423 $JYJ 13301254126 20100619020544 116.4662018 39.93974686 429363639 147238766 0 348 0 0 50#
1383948 $JYJ 13301254298 20100619020137 116.2619171 39.89905167 428610219 147088399 0 74 0 0 50#
```

Fig. 1. Examples of records in the Beijing taxi trace dataset.

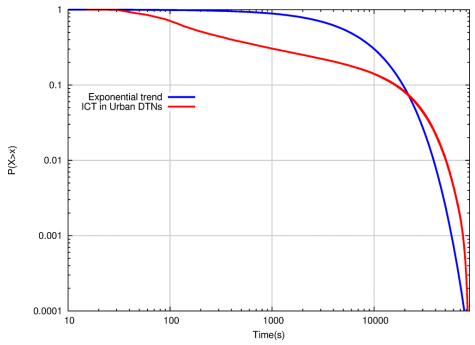


Fig. 2. CCDF of ICT in Beijing's urban DTNs

the total taxis in the city. Each taxi is equipped with a GPS device and upload its information (including location, speed, direction, et al.) about every 60 seconds. There are around 1.22×10^8 records in total. Figure 1 shows an example portion of traces from the dataset. For each row, the record includes a base station ID, company name, taxi ID (id), timestamp (t), current location (l, including both longitude and latitude), another location (out of 54 fixed locations), speed, acceleration, status of the taxi, event, and height. Out of this information, we only use the taxi ID, timestamp, and current location, i.e., (id, t, l), to generate the contacts among taxis for the study within this paper. Notice that GPS traces from taxis have been used recently for inferring human mobility [**?**] and modelling city-scale traffics [**?**]. Therefore, we believe that they are also suitable to characterize the contact patterns among vehicles in large-scale urban DTNs.

B. Contact Characteristics in Beijing Taxi Traces

The *inter-contact time* (ICT) is the interval time between two continuous contacts for same nodal pair. ICT implies the contacts frequency of two nodes. Researchers have focused on the study of ICT characteristics for different mobility patterns in DTNs, and lead to numerous ICT distribution results of both theoretical and real-world mobility models. Most ICT distribution models are created as exponential distributions. According to [**?**], [**?**], [**?**] , the ICT between two mobile nodes generates exponential distributions where the exponential parameter is related to the motion characteristics. Simulations in [**?**], [**?**] based on random waypoint and random direction mobility models confirm such claims. In the real-world mobility traces of vehicular networks [**?**], the ICT between two vehicles also follows an exponential distribution.

Figure 2 plots the CCDF of ICT in one day (June 13th, 210, $86,400$ seconds in total) of Beijing taxi trace records. From the very beginning of the plotting period (until $10,000s$), the plot is almost a straight line with a negative slope on log-log scale. The tail of the distribution decreases rapidly after this range. Such a fact indicates that the ICT distribution in urban DTNs is an exponential nature. Only about 15% of ICT last longer than $10,000s$ in the plot, which means that most of contacts in the network can reappear after a predictably short time. Similar conclusions have been found in another Shanghai taxi trace dataset studied by [**?**], [**?**]. The contact probability between any two nodes based on the global ICT distribution is

$$P(X > x) = e^{-\lambda t}, \tag{1}$$

where the exponential parameter λ reflects the contact strength between nodes. A static λ can be estimated for the whole network by fitting the global ICT distribution to the exponential function. This simple method can measure the contact strength intuitively and efficiently, and has been frequently used in many DTN routing methods.

In our recent work [**?**] , we also study the ICT distributions of individual nodal pairs and find that they follow exponential distributions too. Similar as [**?**], we have the following three basic assumptions on ICTs:

1) The probability that any two mobile nodes will make contact twice or more times in a small duration of time Δt is small.
2) The probability that two mobile nodes will accomplish one contact within Δt is approximated as $\lambda \Delta t$.
3) In the non-overlapping time durations, the contacts of any two mobile nodes are independent.

With these assumption, the ICT of a node pair i and j also follows the exponential distribution with exponential parameters λ_{ij}, according to:

$$\lambda_{ij} = n_{ij} / \sum ICT_{ij}, \tag{2}$$

where n_{ij} refers to the number of contacts between nodes i and j, and $\sum ICT_{ij}$ indicates the total ICTs between these two nodes. Hence, the contact probability of nodes i and j within time period t is

$$p_{ij}(t) = P_{ij}(X \leq t) = 1 - e^{-\frac{n_{ij}}{\sum ICT_{ij}} t}. \tag{3}$$

By fitting the samples of ICTs for a particular nodal pair with Equation 2, we can derive the exponential parameters λ_{ij}. The values of λ_{ij} in Beijing taxi dataset varies significantly from 0.0005 to 0.005. Figure 3 shows the exponential distributions fitting for four pairs of nodes. Clearly, the parameters for each nodal pair according to their historical contacts may improve the accuracy of forecasting contact probability than a unified globe λ. In addition, for Beijing taxi trace data on June 13th, the average contact time is $44.6s$ and the average ICT is $2904.3s$. These results are consistent with the three assumptions above.

To further confirm that the method based on pair-wise ICT distributions can characterize the real contact pattern better than the aggregated global ICT distribution, we also use the coefficient of determination R^2 [**?**] to determine the similarity degree of modelled ICT curves and real observed curves from dataset. Note that R^2 is a number between 0 and 1.0, and used to describe how well a regression line fits a set of data. An R^2

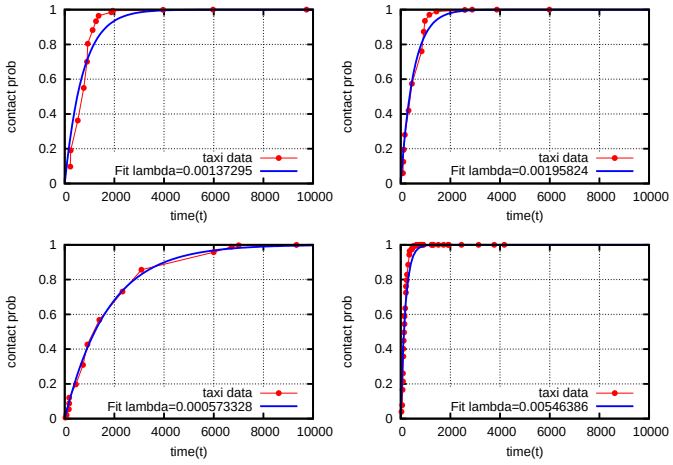
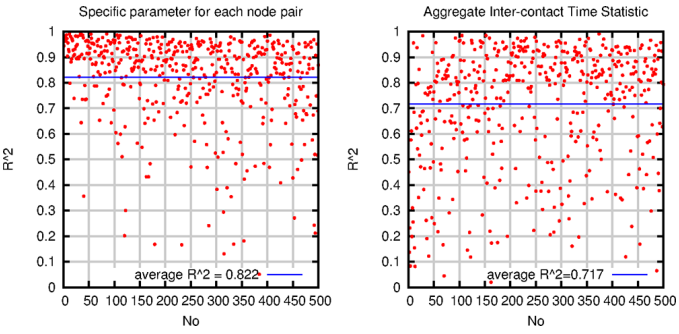


Fig. 3. Pairwise ICT distributions of four nodal pairs in Beijing taxi dataset. Red dots are the real sample data from the dataset while blue curves are the fitted exponential distributions.

near 1.0 indicates that a regression line fits the data perfectly, while an R^2 closer to 0 indicates that a regression line does not fit the data very well.



(a) the pair-wise ICT method (b) the global ICT method

Fig. 4. Comparison of similarities R^2 of the two estimation methods. Here, red dots are the values of R^2 for 500 node pairs, while the blue line shows the average value of these R^2.

For the global ICT method, we attract the λ based on all contact records, and result $\lambda = 0.00077527$. For the pair-wise ICT method, we use Equation 2 to learn the individual λ_{ij} for each nodal pair. Then we calculate R^2 between the estimated distributions based on λ or λ_{ij} with the real distributions from traces of individual nodal pairs in the dataset. Figure 4 shows plots of such comparison over 500 randomly generated nodal pairs. Clearly, R^2 differs for each pair of nodes. The blue line in both plots are the average R^2. The global ICT method has an average of R^2 at 0.717 while the pair-wise one is 0.822. This confirm that using pair-wise ICT distribution can fit the real records better and thus give better contact probability over the global fitting method. Therefore, in our proposed clustering method, we use a distributed algorithm to estimate λ_{ij} for every nodal pair.

IV. Contact-Prediction Clustering-based Routing (CPCR)

Now we are ready to describe the proposed *Contact-Prediction Clustering-based Routing* method (CPCR). CPCR leverages the estimated pair-wise ICT distribution (via Equation 2) to predict the contact probability among nodes (using Equation 3), and uses such probabilities to form clusters. Based on the clustering, CPCR perform its intra- and inter-cluster routing strategies, respectively.

A. Data Structure and Notations

We first review basic data structures and notations used by CPCR. Every node creates or updates its local information upon different events, e.g., happen to contact, periodic update, or record timeout. The following data structures are used at each node.

$$\begin{cases} Node = (\text{id, cid, ContactTable, GatewayTable, ConnectionList, Mes} \\ ContactTable = \{ContactRecord_1, \cdots, ContactRecord_n\} \\ GatewayTable = \{GatewayRecord_1, \cdots, GatewayRecord_m\} \\ ConnectionList = \{Connection_1, \cdots, Connection_o\} \\ MessageList = \{MassageRecord_1, \cdots, MassageRecord_p\} \\ ContactRecord = (id, cid, contactTimes, \sum ICT, contactProb, ti \\ GatewayRecord = (cid, gid, timeout) \\ MessageRecord = (message, FloodingClusters) \\ Connection = (remoteEndNode, message) \end{cases}$$

Table I gives all notations used here.

TABLE I. Notations used in our data structure.

Symbol	Description
id	unique identifier of a node
gid	unique identifier of gateway node id to another cluster
cid	unique identifier of cluster which contains the node
$ContactTable$	a table stores contact records
$GatewayTable$	a table stores gateway records
$ConnectionList$	a list stores all connections, which are established, when nodes contacts
$MessageList$	a list stores messages and their historical info
$timer$	a timestamp to remind nodes of periodic updating
$ContactRecord$	contacts to other nodes and their contact-probability
$GatewayRecord$	cluster id, the gateway node id to the cluster, and a timeout field
$Connection$	message information and the other contacting node
$remoteEndNode$	the other contacting node
$message$	data needs to transmit
$FloodingClusters$	a collection of clusters with the message copy
$contactTimes$	contact times between two nodes
$timestamp$	time record, when connection is up or down
$\sum ICT$ or $\sum ICT_{ij}$	total ICT between two nodes (i and j)
$contactProb$	contact probability between nodes
$timeout$	a timestamp to decide whether a record expires

B. Clustering Criteria

CPCR utilizes $p_{ij}(t)$, contact probability for nodes i and j in time period t, as the clustering metric, since it implies contact strength between the two nodes. Hereafter, $p_{ij}(t)$ can be simplified as p_{ij}. The clustering criteria is given as follows:

Criterion 1 We group nodes with contact probability higher than a probability threshold η into clusters.

Criterion 2 If a node conforms with the Criterion 1 for two or more clusters, the node will join the most stable cluster.

Accordingly, a cluster can be represented as a collection of nodes as follows:

$$C = \{ID_k | k = 1, 2, \ldots, n; \forall i, j \in \{1, 2, 3, \ldots, n\}, p_{ij}(t) > \eta\}. \quad (4)$$

Assume based on Criterion 1 node i belongs to two cluster C_m and C_n, i.e., $p_{ij} > \eta, p_{ik} > \eta$ for $\forall j, k, j \in C_m, k \in C_n$. In this case, we use the minimum contact probability in a cluster to represent the stability of that cluster. By comparing $\min(p_{m_1 m_2})$ and $\min(p_{n_1 n_2})$, $\forall m_1, m_2 \in C_m$ and $\forall n_1, n_2 \in C_n$, node i will join the cluster with higher minimum probability.

Note that two parameters t and η can affect the contact probability and thus affect the results of clustering. When η is a constant, a smaller t decreases $p_{ij}(t)$ so that fewer nodes will meet the clustering criterion. Thus, smaller but more stable clusters will be generated. As a consequence, the overhead of intra-cluster routing and clustering maintenance will decrease whereas the overhead of inter-cluster routing will increase. On the other hand, increasing η can increase the contact strength in clusters and guarantee the efficiency of intra-cluster routing, whereas, more clustering fragments will occur. Overall, reasonable values of t and η need to be configured for tradeoff between clustering and routing.

C. Distributed Clustering based on Contact Perdiction

CPCR use the above clustering criteria to perform the clustering among mobile nodes. The assumption that the network topology remains static in the initial clustering phase in MANET applications, does not apply to DTNs where nodes move randomly over time. Therefore, the clustering algorithm in DTNs should be distributed, heuristic and self-maintaining. In CPCR, nodes cooperatively execute the clustering algorithm and communicate with other nodes. The concept of cluster head, which acts as a coordinator in most clustering designs, is not introduced in CPCR. In our opinion, nodes are homogeneous because of their similar communication and computation abilities in DTNs. Moreover, it is difficult to avoid redundant bottlenecks when routing with cluster heads. The detailed overall clustering method is given in Algorithm 1 which includes information exchange upon contact, periodic updating of $ContactTable$ and $GatewayTable$, and deleting related gateway records when they expire.

Initialization: Initially, each node generates a cluster only including itself. The node also sets $ContactTable$ and $GatewayTable$ empty.

Happen to Contact: When two nodes move into each other's communication range, two nodes will execute the process shown in Algorithm 2.

First, the cluster information of the other node is required. Upon contacting with node j, node i will add or update the contact record with node j and re-calculate the contact probability with node j by Equation 3. If $cid_i = cid_j$, node i add or update the record in $GatewayTable_i$. Node i will decide whether or not node j can be a better gateway to cluster cid_m, when $cid_m \neq cid_j$, as follows:

1) Node identifies the maximum contact probability to cluster cid_j in $ContactTable_j$, which is denoted as:

$$p_j^{cid_m} = MAX\{p_{jk} | \forall k, cid_k = cid_m\}.$$

2) If $p_{ij} \times p_j^{cid_m} > p_{i,gid_i^{cid_m}} \times p_i^{cid_m}$, node j can be a better gateway for node i to cluster cid_m, node i will update the relevant record and reset the gateway timeout.
3) In case that node i has no gateway to cluster cid_m, node j will be set as the gateway only if node j has any record which belongs to cluster cid_m.

If $cid_i \neq cid_j$ (i.e., i and j belong to different clusters), node i will decide whether to join cluster cid_j. The function $whetherToJoinOtherCluster(OtherNodes)$ are decided by criteria given in Section IV-B.

Periodic Updating: To obtain a more precise real-time prediction, the algorithm needs to regularly update the records in $ContactTable$ and $GatewayTable$. The periodic record updating procedure is shown in Algorithm 3.

First, the local node i updates p_{ij} in $ContactTable_i$ with the latest $contactTimes$ and $\sum ICT$. Then, it will go through clustering criteria check. If the node passes Criterion 1, it will remain in the original cluster. Updating $ContactTable$ may change node i's probability to the other clusters. Therefore, node i will also update $GatewayTable_i$ to set the new or better gateway to other clusters. The gateway in $GatewayTable_i$ must in the same cluster of node i. If the node does not pass Criterion 1, it will leave its original cluster and construct

Algorithm 2 Contacting

```
for all Connection ∈ ConnectionsList do
  OtherNodes ⇐ Connection.remoteEndNode
  ContactRecord ⇐ ContactTable[OtherNodes]
  if Connection = established then
    ContactRecord.∑ICT+ = CurrentTime − ContactRecord.timestamp
    ContactRecord.contactTimes++
    ContactRecord.contactProb = 1 − exp{−(ContactRecord.contactTimes / ContactRecord.∑ICT) t}
    if cid = OtherNodes.cid then
      isBetterGateway(OtherNodes)
    else {cid ≠ OtherNodes.cid}
      whetherToJoinOtherCluster(OtherNodes)
  else {Connection = destoried}
    ContactRecord.timestamp = CurrentTime
Gateway-Updating()
```

Algorithm 3 Period-Updating

```
for all ContactRecord ∈ ContactTable do
  ContactRecord.contactProb = 1 − exp{−(ContactRecord.contactTimes / ContactRecord.∑ICT) t}
for all ContactRecord.cid == node.cid do
  FLAG ⇐ true
  if ContactRecord.contactProb < η then
    FLAGE ⇐ false, Break
  if FLAG = false then
    Node.cid ⇐ newClusterID()
    Node.GatewayTable ⇐ NULL
  else
    Gateway − Updating()
```

Algorithm 1 Clustering Algorithm

Initialization: $timer \Leftarrow CurrentTime$, $ContactTable \Leftarrow NULL$, $GatewayTable \Leftarrow NULL$, $cid \Leftarrow id$
while $Simulation \neq end$ **do**
  if Contacting **then**
    Contacting()
  else $\{CurrentTime - timer > t\}$
    Period-Updating()
    for all $GatewayRecord \in GatewayTable$ **do**
      if GatewayRecord.isTimeout **then**
        $GatewayTable.remove(GatewayRecord)$
    $timer \Leftarrow CurrentTime$

a new isolated cluster which only includes itself. In such a situation, the node will keep its own $ContactTable$, while $GatewayTable$ will be set empty, since the old gateway records are meaningless to the new established cluster.

D. Clustering-based Routing Scheme

Based on dynamically formed clusters, our clustering-based routing procedure can be divided into intra- and inter cluster routing procedures. If destination node is in the same cluster of a node, a single-copy routing strategy will be executed as intra-cluster routing algorithm. The local node will not deliver the packet until contacted by the destination node. Considering high contact probability inside a cluster, such direct delivery can simplify route decision making and ensure acceptable delay in the intra-cluster routing. If the local node and the destination belong to the different clusters, inter-cluster flooding approach will be used. Every cluster is considered as an abstract "node". Therefore, it ensures that a cluster can only accept no more than one copy of a message. Since CPCR is event-driven, the routing procedure will be executed when contacts happen. When node i contacts with node j, node i will traverse every message to determine whether or not to relay the messages. The routing procedure can be described as Algorithm 4.

Algorithm 4 Clustering-based Routing

if $Connection = established$ **then**
  $remoteNode \Leftarrow Connection.remoteNode$
  for all $message \in MassageList$ **do**
    $destination \Leftarrow message.destination$
    if $destination = remoteNode$ **then**
      $sendAndDelete(message)$, CONTINUE
    if $belongToSameCluster(this, destination)$ **then**
      CONTINUE
    if $belongToSameCluster(destination, remoteNode)$ **then**
      $sendAndDelete(message)$
    else $\{belongToSameCluster(this, remoteNode)\ \}$
      if $isGateway(remoteNode, destination.cid)$ **then**
        $sendAndDelete(message)$
    else
      $copyAndDelete(message)$

If the message's destination is node j, node i will send the message to node j. Then, the message will be deleted (as $sendAndDelete(message)$). If the destination of the message is in the same cluster of local node, message will not be relayed until the local node meets the destination node, which can be considered as intra-cluster routing. For inter-cluster routing where the remote node j is not the destination and not in the same cluster of the local node, node j can be a relay point if it satisfies one of following circumstances:

1) The target cluster of the message is the same as the cluster of node j.
2) The node j is the gateway node from the cluster of node i to the target cluster. As the definition for gateway, we can figure out that node j belongs to the same cluster as the local node.
3) Three nodes (i, j, and the destination node) belong to three different clusters. In this case, local node will send a copy to the remote node, only if there is no copy in the cluster of node j. When message has been sent, node i will store a record that the $cluster_j$ has a copy of the message, avoid of sending a message to the same cluster again.

We determine whether two nodes are in the same cluster by explore the $ContactTable$ and $GatewayTable$ of the local node. If we can neither find a record for a destination in $ContactTable$, nor find a gateway to the target cluster, inter-cluster procedure for the third condition will be executed.

V. Performance Evaluation

In this section, we present results from three set of simulation experiments over Beijing taxi dataset to evaluate the efficiency of proposed clustering and routing method CPCR and its scalability. We first evaluate the clustering performance and determine the parameter settings of (t, η). We then compare delivery ratio and overhead of three single-strategy routing protocols with CPCR in the large-scale urban DTNs scenario to demonstrate its routing performance. The three compared routing methods are *Epidemic Routing* [**?**], *Direct Delivery* (DD) [**?**] and *PRoPHET*. Notice that Epidemic and DD are used in the inter-cluster and intra-cluster phases of CPCR respectively. PRoPHET is a two-hop forwarding strategy whose forwarding strategy is also based on the estimated contact probability. Last, with different network scales, we reveal the scalability of all methods.

All routing methods are implemented in Opportunistic Networking Environment (ONE)[**?**] simulator which is designed for evaluating DTN routing and application protocols. It allows users to create scenarios based upon different synthetic movement models and real-world traces and offers a framework

TABLE II. PARAMETER SETTINGS OF CLUSTERING EVALUATION

Parameter	Value
Simulation Time	$4,000s$
Buffer Size	2.5MB
Message Size	$50 - 100kb$
Transmission range	$200m$
Inter message creation time	$5s$
Bandwidth	$256k$
Node Number	$1,000$ or $1,500$
Area	$20 \times 20\ km^2$
η	$0.15, 0.3$
t	$200s, 400s$

for implementing routing and application protocols (already including six well-known routing protocols). All simulation scenarios are extracted from the Beijing taxi dataset we described in Section III-A and imported to ONE as movement models.

A. Clustering Evaluation

In this section, we aim to evaluate the performance of our clustering method by using the average scale of clusters (i.e., the average node numbers in a cluster). Note that the average scale of clusters is affected by clustering rules and is an important criterion for evaluating clustering performance. Whether or when the cluster becoming stable is another criterion for evaluation. Quantifying a certain clustering criterion in different scenarios is not easy. For example, in a small-scale network with high node density, small-scale clusters may be better than large clusters for message delivery. McDonald and Znati [**?**] proposes a method of comparing the clustering performance when change its own factor. We adapt their approach to evaluate our clustering algorithm. Simulation parameters of our clustering algorithm are shown in Table II, in which the variable parameters are the node number, the contact probability threshold η and the contact forecast time t.

The average scale of clusters reflects the granularity of clustering, and the curve of the change of the average scale of clusters can indicate whether a cluster can become stable. Two node scales (i.e. node numbers), namely, $1,000$ and $1,500$, are chosen to compare the clustering performance by changing a single parameter, contact probability threshold or contact forecast time. Figures 5 and Figure 6 illustrate the change of the average scale of the cluster with time.

Figure 5 illustrates the curve of the cluster scale variation by changing η from 0.3 to 0.15, at a fixed forecast time of $400s$, for both network scales. A larger network scale leads lager average scale of clusters. High node density increase the opportunity to contact so that the contact probability between nodes becomes higher than η, which also increases the degree of convergence. The η influences the average scale of clusters directly. In Figure 5 (left), the peak value of the scale is 13.6 with $\eta = 0.15$, while the peak value is 9.4 with $\eta = 0.3$. Similar trends can be found in the right subfigure of Figure 5.

Figure 6 uses the same contact threshold ($\eta = 0.3$), but its forecast time t changes from $400s$ to $200s$. In Figure 6 (left), when $t = 200$, the cluster scale peak value is 6.8, and when $t = 400s$, the peak reaches 9.4. This result is due to the fact that forecast time increases the contact probability value, which causes more nodes to satisfy the clustering criteria and

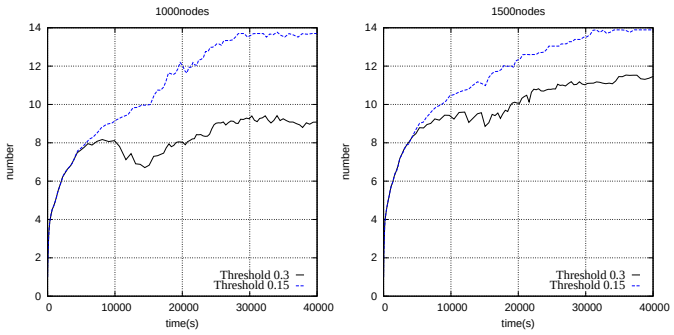


Fig. 5. Average scale of clusters for different η (left: 1,000 nodes; right: 1,500 nodes)

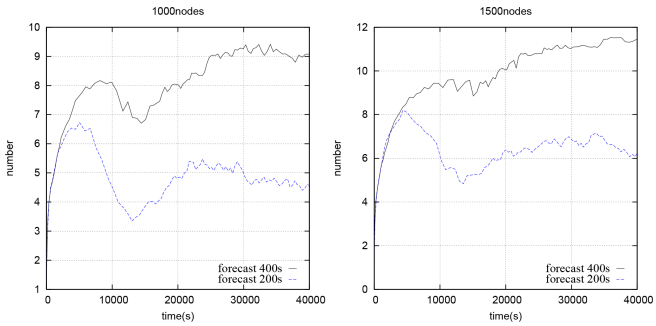


Fig. 6. Average scale of clusters for different t (left: 1,000 nodes; right: 1,500 nodes)

join clusters. Similar conclusion for the case with $1,500$ nodes (right subfigure in Figure 6).

In both cases, the curves can achieve stability after $2,000s$. It shows that the proposed clustering algorithm can cluster the network effectively and stably. Overall, the clustering phase can organize the network into clusters which have certain degree of granularity and achieve stability, which benefits the routing task inside and across the clusters in large-scale networks.

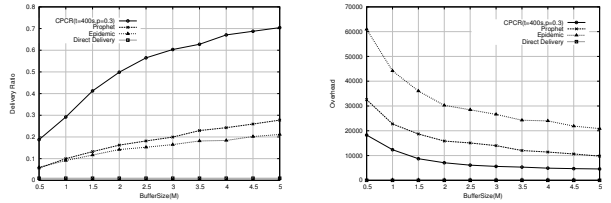
B. Routing Performance

In this set of simulations, we randomly select traces of $3,000$ vehicles from the central area ($20km \times 20km$) of Beijing. As revealed in [**?**], the central area of the city has numerous hotspot areas, which we believe is an appropriate scenario for the clustering based-routing. We investigate delivery ratio and overhead of different routing protocols under different buffer sizes. Detail parameter settings are shown in Table III.

Figure 7(a) shows that most protocols have higher delivery ratio with increased buffer size. The proposed routing algorithm yields a higher delivery ratio than the other single-strategy protocols because in such a large-scale scenario, clustering strategy groups nodes into different clusters, which makes the network more coarse-grained. As a result, from a single node's point of view, the network scale becomes much smaller. Furthermore, a node can make its forwarding decision inside or outside of the cluster more wisely which not only saves the buffer of local nodes, but also tends to the route with higher contact probability.

TABLE III. PARAMETER SETTING IN SIMULATION ONE

Simulation Time	$10,000s$
Area	$20 \times 20\ km^2$
Nodes Number	$3,000$
Transmit Range	$200m$
Package Size	Random $(50, 100)$k
Buffer Size	$[0.5, 1, \cdots, 5]$M
Routing Methods	CPCR ($t = 400$,$\eta = 0.3$), Epidemic, PRoPHET, DD



(a) (b)

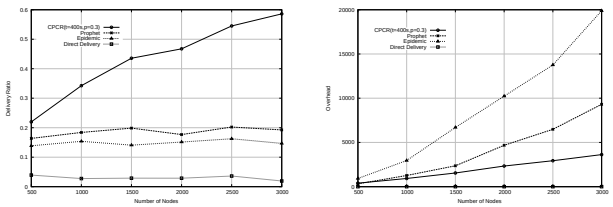
Fig. 7. Delivery ratio and overhead of all methods with different buffer sizes

Overheads are shown in Figure 7(b), which indicates the average number of copies generated by the network until a packet is successfully delivered. Direct Delivery is a single copy protocol which dose not produce additional copy of packets. Epidemic Protocol duplicates a copy whenever a node contact with others, thereby producing the highest overhead among the protocols. In the clustering-based routing framework, nodes generate copies only when they meet the other nodes in different clusters where no copy of the specific packet exists, thereby controlling the overhead. On the other hand, the overhead of PROpHET Protocol is two times greater than than that of the proposed algorithm because of its high forwarding probability and one-hop decision making. As a result, the clustering-based routing algorithm yields a really low overhead between those of PROpHET and Direct Delivery Protocol, and has considerable performance in large-scale real-world urban networks.

C. Scalability

Scalability is one of the major consideration of routing protocols in large-scale DTN networks. The network scale is mainly changed by the number of nodes within the network. A greater number of nodes involved in the network indicates greater overheads since many additional copies of packets and messages to synchronous routing information are generated. In this case, the network may encounter a bottleneck and the performance of most protocols may stagnate or deteriorate. In this set of simulations, the scalability of protocols is compared by varying the number of nodes from 500 to $3,000$. The other parameters are remaining the same as those in Table III.

Figure 8(a) shows the delivery ratio in the network with different numbers of nodes. The performances of the single-strategy routing protocols do not improve as the number of nodes increases. In general, increasing nodes density can bring more contact opportunities which should lead to higher delivery ratio. However, if the network scale is large enough, the buffer capacity becomes limited during the single-strategy routing because of their overall forwarding behaviors, which limit the increase of delivery ratio. The clustering-based routing algorithm can adapt to the increasing scale. If the clustering parameters are constant $(t = 400, \eta = 0.3)$, the number of



(a) (b)

Fig. 8. Delivery ratio and overhead of all methods with different network scales

clusters does not change considerably when the network scale increases. Instead, the size of each cluster, which is denoted by number of nodes inside each cluster, increases. Therefore, from a single node's point of view, the buffer capacities of clusters increase. When executing inter-cluster routing, each cluster is considered a bigger "node". Thus, the network maintains the small number of "nodes" which have a higher buffer capacity, thereby improving routing performances.

Figure 8(b) reveals the superior scalability of the proposed framework based on the other metric, namely, overhead growth rate. When the network is sparse, overhead of single-strategy protocols is close to that of clustering-based routing. However, as the number of nodes increases, overheads of those protocols, expect that of Direct Delivery, become dramatically high. On the other hand, the proposed algorithm can restrict its overhead growth rate. The reason is similar to the one we described above for delivery ratio. In large-scale urban DTNs, the proposed algorithm can help reduce network overhead and achieve scalability thereby outperforming traditional single-strategy routing protocols.

VI. CONCLUSION

In this paper, we analyzed the contact characteristics of large-scale urban DTNs over a real-world trace dataset from Beijing. Distribution of ICT exhibits an exponential nature in such kind of urban DTN networks. To improve the scalability of routing protocols, we proposed a clustering-based routing algorithm by using contact probability estimated from ICT distributions as the clustering metric. Extensive simulation results demonstrated that the performance and scalability of the proposed framework exceed those of traditional single-strategy protocols. There are several possible future extensions of this study. First, the algorithm can be evaluated from the aspect of clustering in terms of its stability and maintenance overhead. Second, the algorithm can be further improved to be more adaptive by understanding the influence of parameters. Last, real-world deployment of the proposed method can be used to evaluate its effectiveness and suitability for real-life application. We leave all these as our future work.

VII. ACKNOWLEDGMENT

This research has been partially supported by the US National Science Foundation (NSF) under Grant No. CNS-1319915 and CNS-1343355, the National Natural Science Foundation of China (NSFC) under Grant No.61300173 and No. 61170295, the Project of Aeronautical Science Foundation of China under Grant No.2013ZC51026 and No.2011ZC51024, the Fundamental Research Funds for the

Central Universities under Grant No. YWF-12-LXGY-001, and the State Key Laboratory Software Development Environment and Network Information and Computing Center of Beihang University.