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,zjj@buaa.edu.cn

***Abstract*—Routing in Public Transport Networks is particularly challenging due to the high mobility, rapidly changing topology. Though clustering of nodes can aid forwarding decision in these Delay Tolerant Networks (DTNs), centralized clustering process is extremely costly in a large network. Moreover, it is critical to choose proper metrics to cluster nodes. In this paper, we propose a contact-predict clustering-based routing algorithm, CPCR, for large-scale urban DTNs. This algorithm computes the probability for each node pair and cluster that nodes distributively into higher contact-strength clusters using nodal local contact information. Depending on the clusters, intra-cluster and inter-cluster routing strategies will be adopted. Extensive simulations indicate that CPCR maintains relatively stable clusters and enhances the routing performance. The results demonstrate that clustering-based routing algorithm is rational and promising for large-scale urban DTNs.**

***Keywords*—clustering, inter-contact time ,contact prediction ,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, in which, vehicles communicate with each other in order to disseminate data using opportunistic contacts.

In this paper, we propose an clustering-based routing scheme for large-scale urban DTNs. We first analyze a real-world urban vehicular tracing data set to reveal the Inter-contact time distributions of mobile vehicles in a large city. Our study confirm that both the global ICT and the individual ICT between each nodal pair follow the exponential distributions. It is demonstrated that the distribution for ICT of nodal pairs is more accuracy than the global ICT distribution. Therefore, a 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 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

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 formulate clusters and ensure their

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.

stability. When we consider an urban DTN, the nodal scale and mobility will further increase the cost of maintaining clusters.

In summary, the key challenge of hierarchical routing is the clustering method, especially in large scale DTNs. To adapt to the scenario of large scale urban DTNs and solve the routing problem, a cluster-based routing algorithm is proposed.

III. Characteristics of Large-scale Urban DTNs

In this section, we introduce the data set of taxies in Beijing, and analysis the Inter contact time property of the data set. 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 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. For each row, the record includes a base station ID, company name, taxi ID (id), time stamp (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. We only utilize the taxi ID, time stamp, and current location, i.e., (id, t, l), to generate the contacts among taxis for the study. Note 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

In the real-world mobility traces of vehicular networks [**?**], the ICT between two vehicles follows an exponential distribution[**?**], [**?**], [**?**].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 statistical 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 Paper [**?**], authors studied the ICT distributions of individual nodal pairs and found that 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

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

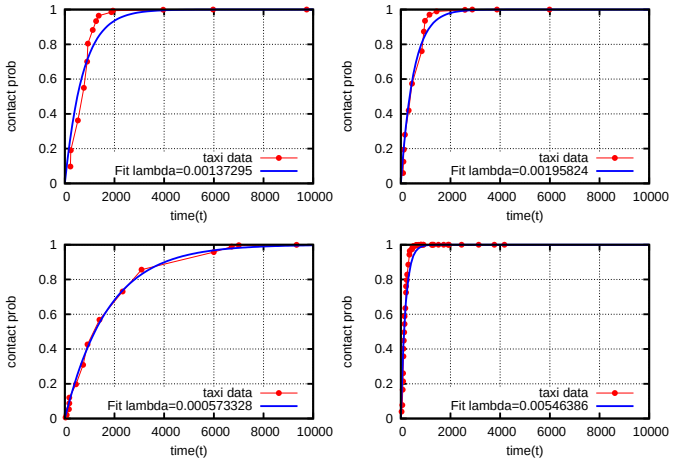
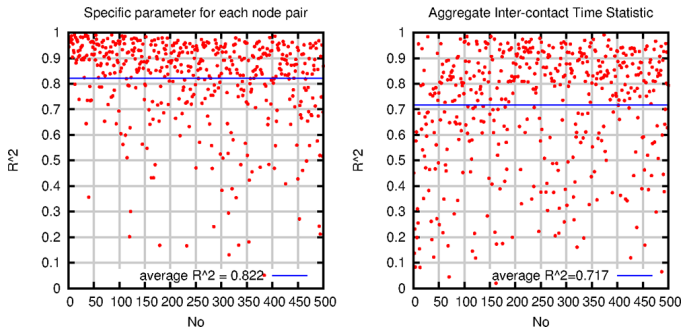


Fig. 1. Pairwise ICT distributions of four nodal pairs in Beijing taxi dataset. Red dots are the real sample data from the dataset while blue curves are the fitted exponential distributions.

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 1 shows the exponential distributions fitting for four pairs of nodes.

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 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 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. 2. 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 2 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, the contact probability between nodes can by computed by the local information, including the contact time and sum of ICTs, which is beneficial to our distribute clustering and routing process.

IV. Clustering and Routing Desgin

Now we are ready to describe the *Contact-Prediction Clustering-based Routing* (CPCR) algorithm. The 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. In this section, the clustering criteria and routing process are given. Since the process is in a distribute way, we introduce the event-driven process of information-exchanging to from cluster and deliver messages in detail.

A. Notation Definition

Basic data structures and notations in CPCR is reviewed as follows.

$$Node :< id, cid, \{CR\}, \{GR\}, \{Msg\}, \{MLog\} > \quad (4)$$

$$CR :< id, cid, encounters, T_{disconnect}, \sum ICT, prob > \quad (5)$$

$$GR :< cid, id, prob, T_{established} > \quad (6)$$

$$Msg :< mid, id, content, ttl > \quad (7)$$

$$MLog :< mid, \{< cid, T_{delivery} >\}, status, > \quad (8)$$

Where id is the node ID, cid is its cluster ID, CR represents a contact record, GR donates the gateway records. Msg donates the message carried by the node. For each node, a timer is set to update the those information above periodically. In a CR, it records the encounter times by field $encounters$ and the sum of ICTs by $\sum ICT$. It also records the timestamp of latest disconnection as $T_{disconnect}$ in order to figure out next ICT. Every message also contains an ID, say mid. In addition, when every message is delivered, the $\{MLog\}$ will store the cid and the current time to figure out which cluster this message has been delivered to, et al.

B. Clustering Criteria

CPCR utilizes $prob_{ij}(t)$, contact probability for nodes i and j in time period t, as the clustering metric. Hereafter, $prob_{ij}(t)$ 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. To simplify, we define the cluster with larger minim of contact probability in cluster is more stable.

C. Routing Strategy

Based on dynamically formed clusters, our clustering-based routing procedure can be divided into intra-cluster 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 can control the message copies in a cluster.

D. Event-driven process to form clusters and delivery messages

Since the packets are delivered only when there exists connections between nodes, the following processes ,including updating the contact and gateway records, adopting the messages need to be delivered and send those message and finally updating the logs of messages and the massages,are triggered off when a connection is established, as algorithm IV-D. Meanwhile, when a connection is dismissed, it will update the CR with remote node of the connection and set the $T_{disconnect}$ to the current time.

Algorithm 1 algorithm when a connection is established

```
updateCRs();
if this.Cluster! = remoteNode.Cluster then
    joinOrNotRemoteCluster();
updateGRs();
msgs = getValidMsgs();
delivery(msgs);
updateMLogs();
updateMsgs();
```

update CRs: When node N_i contacts with node N_j, CRs will be updated. If $\exists CR =< N_j, , . > \in \{CR\}_{N_i}$, the $encounters$ adds on 1 and $\sum ICT$ adds by the difference between the $T_disconnect$ and current time. Otherwise, if $CR =< N_j, , . > \notin \{CR\}_{N_i}$, a new $CR =< N_i, 1, 0, 0, \eta >$ will be created, and set the $prob = \eta$, for that there is no ICT yet.

whether to join the remote cluster: When the local node and the other node of the connection do not belong to the same cluster, we should figure it out whether this node should join to the other cluster. If $prob_{ij} >= \eta$,and by querying the local $\{CR\}$, the minim contact prob in the cluster of the local node. Assume that the $\{CR\}$ is as table IV-D, $\eta = 0.3$ and forcast time $t = 200s$.

In this case, the minim contact prob in $C1$ is 0.3, whereas, the mimim contact prob in $C2$ is also 0.3. So as local node,

TABLE I. THE CONTACT RECORDS OF NODE1,$cid = C1$

id	cid	encounters	$T_{disconnect}$	$\sum ICT$	prob
2	C1	1	1000	-	0.3
3	C2	2	3000	500	$1 - e^{-0.8} = 0.55$
4	C1	2	2000	1000	$1 - e^{-0.4} = 0.3297$

TABLE II. THE CONTACT RECORDS OF NODE 3,$cid = C2$

id	cid	encounters	$T_{disconnect}$	$\sum ICT$	prob
1	C1	2	3000	500	$1 - e^{-0.8} = 0.55$
5	C2	2	2000	1000	$1 - e^{-0.4} = 0.3297$
6	C2	1	2500	-	0.3

node1 will not join the cluster $C2$. Otherwise, the minum contact prob is larger than that of $C1$, node1 will join the $C2$ by assigning the $cid = C2$ and copy and replace the $\{GR\}$ from the remote node3.

Update GRs: Gateway is a host in the same cluster, where it performs higher probability to another cluster. From the $\{CR\}$ of node1, we can only figure out the probability of node1 to a node of another cluster $C1$(node3), so a $GR =< C2, 1, 0.55 >$ will be assigned to identify which node has the max prob to cluster $C2$. If $node1$ belongs to the same cluster of the remote node, the two GR will be merged. If two GRs conflict with each other, the one with latest $T_established$ will be selected.

Get valid messages: this process is an important step to implement the routing strategies. A node can carry many messages. When contacting with another node, routing strategies are implemented by deciding which message need to be delivered. Messages satisfied following rules will be delivered.

- The remote node is the destination node of the message.
- The remote node belongs to the same cluster of the destination node (searching the $\{CR\}$ to find out the cluster of destination node if $\exists CR =< destination, C_j, ... >\in \{CR\}$).
- The remote node belongs to a cluster which carries no copy of the message (searching the $\{Mlog\}$).

update MLogs and Msgs: If a message is delivered to the remote node, the mlog will be updated in case of sending the same message to another cluster twice or more. If the remote node is the destination, the status will be set $Success$. Message out of ttl or send to destination successfully will be dropped.

V. 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.

TABLE III. 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	$200s, 400s$

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 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, it aims 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 III, 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 3 and Figure 4 illustrate the change of the average scale of the cluster with time.

Figure 3 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 3 (left), the peak value of the scale is 13.6

TABLE IV. 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

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 3.

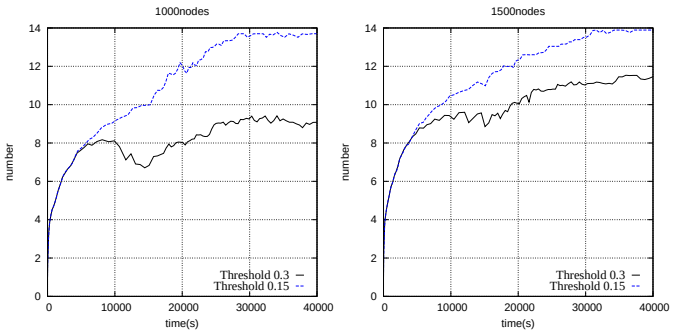


Fig. 3. Average scale of clusters for different η (left: 1,000 nodes; right: 1,500 nodes)

Figure 4 uses the same contact threshold ($\eta = 0.3$), but its forecast time t changes from $400s$ to $200s$. In Figure 4 (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 join clusters. Similar conclusion for the case with $1,500$ nodes (right subfigure in Figure 4).

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

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 IV.

Figure 5(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.

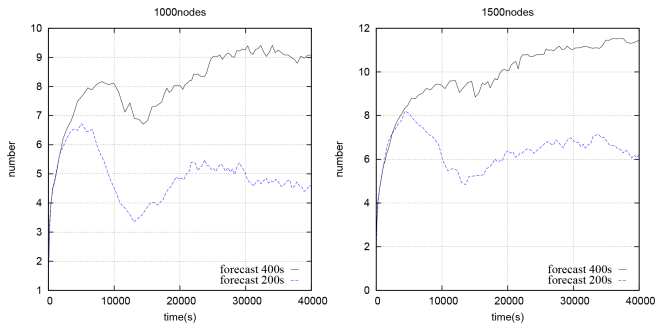
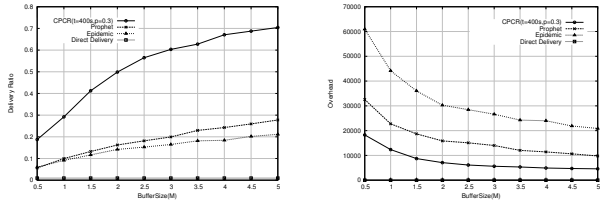


Fig. 4. Average scale of clusters for different t (left: 1,000 nodes; right: 1,500 nodes)



(a) (b)

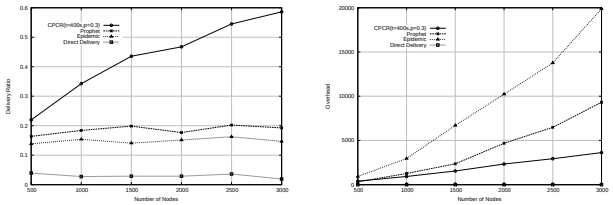
Fig. 5. Delivery ratio and overhead of all methods with different buffer sizes

Overheads are shown in Figure 5(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 IV.

Figure 6(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



(a) (b)

Fig. 6. Delivery ratio and overhead of all methods with different network scales

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 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 6(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 data set from Beijing. Distribution of ICTs 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 node pair-wise ICT distributions as the clustering metric. It can form the clusters in a decentralized way. The simulation results demonstrate that CPCR can build relative stable clusters and performs better than other routing algorithms in routing and scalability.