A Cooperative Caching Scheme Based on Mobility Prediction in Vehicular Content Centric Networks

Lin Yao*†, Ailun Chen*†, Jing Deng‡, Jianbang Wang§, and Guowei Wu*†
*School of Software, Dalian University of Technology, China
†Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China
‡Department of Computer Science, University of North Carolina at Greensboro, U.S.A.
§Faculty of Engineering, National University of Singapore

***Abstract*—Vehicular Content Centric Networks (VCCNs) emerge as a strong candidate to be deployed in information-rich applications of vehicular communications. Due to vehicles' mobility, it becomes rather inefficient to establish end-to-end connections in VCCNs. Consequently, content packets are usually sent back to the requesting node via different paths in VCCNs. To improve network performance of VCCNs, node mobility should be exploited for vehicles to serve as relays and to carry data for delivery. In this work, we propose a scheme called Cooperative Caching based on Mobility Prediction (CCMP) for VCCNs. The main idea of CCMP is to cache popular contents at a set of mobile nodes that may visit the same hot spot areas repeatedly. In our CCMP scheme, we use Prediction based on Partial Matching (PPM) to predict mobile nodes' probability of reaching different hot spot regions based on their past trajectories. Vehicles with longer sojourn time in a hot region can provide more services and should be preferred as caching nodes. To solve the problem of limited buffer at each node, we design a cache replacement based on content popularity to guarantee only popular contents are cached. We evaluate CCMP through the ONE simulator for its salient features in success ratio and content access delay compared to other state-of-the-art schemes.**

***Index Terms*—Cooperative Caching; Mobility Prediction; VCCN**

I. Introduction

Vehicular Ad-hoc Networks (VANETs), a special type of mobile ad hoc network and preferred for intelligent transportation systems, usually contain regular users and road side units (RSUs). Regardless of safety-related or non-safety-related applications, vehicle nodes in VANETs usually depend on unique IDs to locate destinations and to establish/maintain end-to-end communications. However, node mobility makes it difficult to maintain the ongoing communication paths [1]. Interestingly, most communications are focused on contents but their actual carriers, giving rise to the so-called Vehicular Content Centric Networks (VCCNs) [2]. In comparison to regular networks, VCCNs allow nodes to communicate directly prior to the knowledge of any identification or IP address and asynchronous data exchanges are supported [2][3].

In VCCNs, a request/reply communication model is usually adopted. Contents are stored at the source as well as at a few other nodes. A requesting node broadcasts an Interest message for the content. The Interest message will be forwarded through the network toward the source node. The delivery of such messages is usually based on node mobility, i.e., nodes only forward messages toward other direct neighbors. However, if the requested content is found at an intermediate node, it will deliver the content to the requesting node instead of forwarding the Interest message further. With in-network caching, access latency and query overhead can be improved [4]. Depending on whether the content routers cooperate with each other, caching mechanisms can be classified into cooperative caching and non-cooperative caching [5]. In non-cooperative caching, content routers make their caching decisions independently, which may lead to problems of frequent cache updates, over-caching, etc. Cooperative caching benefits from the cooperation among caching nodes with higher hit ratio and lower access delay.

However, cooperative caching techniques for regular CCNs [6][7] cannot be directly employed in VCCNs because of the huge diversity in mobility patterns for different nodes. The lack of persistent network connectivity in VCCNs makes it difficult to establish end-to-end connections, or even to use the reversed path to deliver the requested content (see Fig. 1). Furthermore, while it may seem attractive to cache all contents, such a universal caching strategy can be too costly [8].

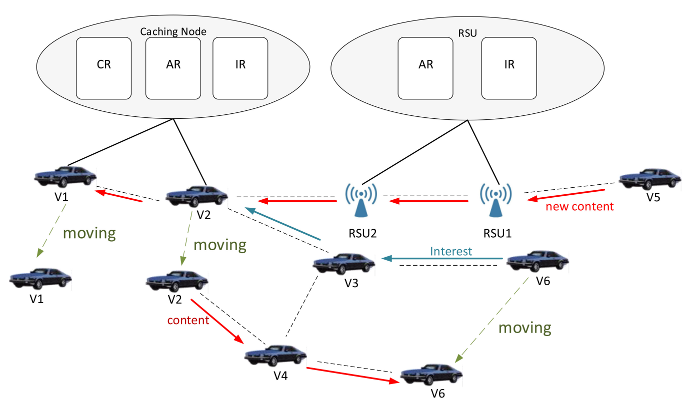


Fig. 1. VCCN network model

In this work, we propose a caching scheme called Cooperative Caching based on Mobility Patterns (CCMP) for VCCNs. The novel idea of CCMP is to take advantage of essential hot regions, where a set of caching nodes are likely to visit repeatedly and possibly stay for a long period of time. In fact, vehicles' trajectory often exhibits a high degree of repetition as a result of regular visits of certain places. These can be linked as the core or backbone of opportunistic content delivery. Our CCMP scheme takes advantage of such a repetition of visits to prioritize content caching. We adopt a technique call Prediction based on Partial Matching (PPM) to predict one vehicle's future trajectory based on the past traces. Nodes with higher chances of staying in hot regions and with longer sojourn times are chosen to cache contents. The main contributions of our paper are listed as follows:

(1) CCMP partitions urban areas into different hot regions based on users' mobility patterns and such regions can be adjusted based on dynamic vehicle density;
(2) PPM is adopted to predict the probabilities of different nodes' re-visiting different hot regions;
(3) We further propose a utility-based cache replacement technique that evaluates query history and content popularity, so that only popular contents are cached on those nodes visiting hot regions frequently;
(4) CCMP is compared to four other state-of-the-art caching techniques through extensive simulations and shown to be superior in success ratio and access delay.

The rest of this paper is organized as follows. Section II reviews related work. In Section III, network model, problem statement, and PPM are introduced. The details of our proposed CCMP scheme are presented in Section IV. We evaluate CCMP and compare it with other schemes in Section V. Finally, we conclude our work in Section VI.

II. Related Work

In this section, we discuss works that are related to our approach.

CCNs- Most of the existing caching schemes in CCNs can be classified into advertisement-based ones and non-advertisement-based ones depending on whether the cache state is advertised to other content routers [5]. Both categories belong to heterogeneous caching schemes, in which routers cache different contents due to different cache sizes and observation history [6].

Advertisement-based approaches aim to reduce access delay by advertising the cache state, with examples such as [9][10][11][12]. In [9], an inter-domain cooperative caching mechanism was proposed. A main content router, similar to the border area router, collects cache states of routers in its domain and then advertises them to its peer domains. CATT [10], an intra-domain cooperation scheme, requires a content router to advertise the availability information of the content to its k-hop neighbors. In [11][12], there exists a centralized manager that takes charge of maintaining the cache state of the network.

In non-advertisement-based approaches, content routers do not rely on any additional advertisement mechanism to deliver their cache states. ProbCache [7], a probabilistic caching mechanism, was first proposed to approximate the capability of caching routers and then only popular contents are cached. In this scheme, contents are divided into small chunks for storage in different nodes in order to improve efficiency. WAVE [13] uses content popularity from the suggestion of upstream routers to decide downstream routers' chunk selection. In [14][15], each content router is associated with a label and only caches chunks whose number is equal to its label.

Caching in Mobile Networks- In recent years, caching for mobile networks has also been studied. Psephos [16] addressed caching with heterogeneity with a fully distributed technique, in which only items receiving the highest votes are stored in content routers. In [17][18], social attributes such as contact patterns and relationships are used to choose caching nodes. In [4][19][20], those nodes located at key positions are chosen as caching nodes to balance between data accessibility and caching overhead. Opportunistic approaches have also been proposed and investigated [21][22].

For instance, in [4], the Network Central Location Cooperative Caching (NCLCC) scheme identifies several NCLs to cache contents. These locations are expected to have a high chance of promptly replying users' queries and their selections are based on a probabilistic metric evaluating data transmission delay among nodes in the network. In Distributed Probabilistic Caching strategy [20], caching decisions are made by each node independently and three factions are considered: users' demands mined from the collected Interests, relative movement of the receiver and the sender, and the importance of vehicles based on the degree and betweenness centrality in the ego network which is composed of the current node and its one-hop neighbor with the current node as the center of the network. The Proper Node Cooperative Caching (PNCC) scheme [22] is essentially a greedy algorithm for choosing proper cache nodes, considering the contacts between pairwise nodes and the query situation for different data items. In Location Dependent Cooperative Caching (LDCC) [23], a state prediction model is applied to estimate the movement behavior of clients. The probability of the mobile client staying in a specific valid scope for any given time is then derived. Then, clients with higher probability will be chosen as caching nodes within a short period of time.

Furthermore, there are some other works by exploiting the public transportation to improve the network performance. In [24], Mobile Infrastructure Based VANET (MI-VANET) was proposed to make buses constitute a mobile backbone for data delivery and the ordinary cars run as the second tier (similar to a tiered-system in [25]). MI-VANET can improve both the network connectivity and the delivery ratio of packets. In [26], the bus-assisted transmission protocol was proposed by benefiting from the predictability and regularity of buses. To improve the reliability of transmitting packets in the VANET, packets are transmitted by switching between common vehicles and buses. In [27], an efficient service circulation and discovery scheme was proposed with the aid of public transportation systems. A virtual backbone is established by the buses that have the fixed route or routine bus ride in public transportation systems so that the required data can be effectively disseminated and discovered through the backbone. Wang et al. took advantage of buses fixed routes and predictable traces to alleviate the limited coverage issues of RSUs in [28]. Instead, we focus on the problem of the cooperation among regular users, private vehicles, and taxi, in addition to buses to cache popular contents for efficient delivery.

Summary- Though there have been some works taking advantage of mobile nodes to cache contents, none has taken a full advantage of trajectory records and prediction toward future trips. Instead, a successful caching scheme should solve two problems cohesively: the selection of caching nodes and the selection of caching contents. Our approach uses PPM to process past travel patterns in order to predict future trips before it selects the best caching nodes for the chosen cache contents.

III. Network Model and Problem Statement

In this section, we firstly describe our network model, then propose our problem statement.

We consider VCCNs consisting of a set of users and RSUs as shown in Fig. 1. Each user can communicate with other users or RSUs within wireless communication range. For vehicle-to-vehicle communication, each vehicle communicates and shares the messages with other vehicles over one-hop neighbors within its communication range. As most VANETs, RSUs are deployed throughout the network to relay messages as well as to facilitate Internet connections. We further assume that every vehicle is equipped with a GPS receiver for the location service.

Different from the push-based communication model in traditional VANETs, VCCNs usually take the pull-based approach. Interests are sent by users toward their direct neighbors. Such Interests are stored, carried, and forwarded until a Time-To-Live (TTL) timer expires or they reach the requested item cached at a node, which then uses similar mechanism to return the content. As contents are delivered inside the network, the intermediate nodes make their own decisions on whether to cache them or not based on a caching scheme (such as CCMP proposed in this work). Consequently, a given content may be available at multiple providers instead of being stored exclusively at a specific node (e.g. data producer) in VCCNs. All cooperative techniques in communication networks such as ad hoc networks depend on users willingness to help others in order to receive help. Unwillingness to help or misbehavior among different users can be interesting research topics but we believe they are out of the scope of this work.

The data flow model is as follows: Initially, after contents are produced, they are transmitted to the closest RSUs which broadcast them to other RSUs. As the contents are queried and delivered by regular users, they are cached by some vehicles. Our algorithm focuses on choosing nodes and contents to cache.

Due to limited buffer storage, it is obviously inefficient to cache all contents on all vehicles. Instead, only those nodes that are more likely to help others with contents should serve as the caching nodes. Furthermore, only popular contents, with a large number of requests, should be cached; less popular contents should be replaced by popular ones. With the dynamic feature of VCCNs, the selection of caching nodes and caching contents can intriguing. Therefore, we have the following problem statement:

Problem Statement (Selection of Caching Nodes and Caching Contents in VCCNs): *In VCCNs, the goal is to allow contents to be queried and delivered efficiently among different users and RSUs. Different nodes can be chosen to cache contents that are of interest to users. How to choose these nodes to improve success ratio as well as to reduce access delay of content requests? Furthermore, after the caching nodes are selected, how to choose different contents to cache so that they can be accessed with the best efficiency?*

IV. The Cooperative Caching based on Mobility Prediction Scheme

In this section, we present the details of the CCMP scheme. First, we list frequently used notations in Table I.

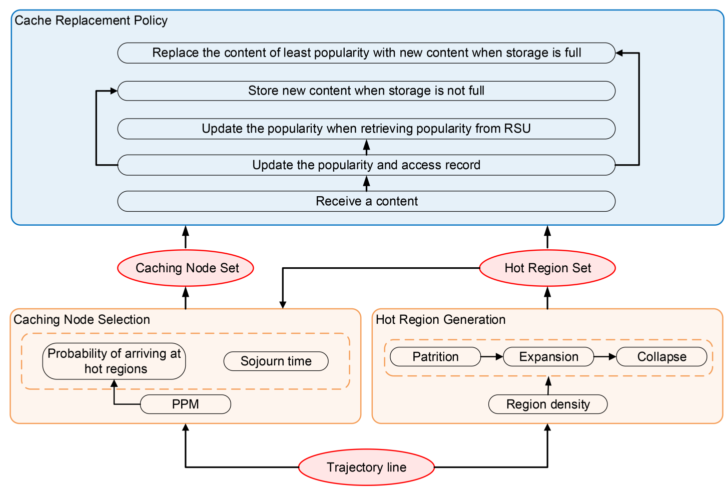


Fig. 2. System model

TABLE I
FREQUENT NOTATIONS

Notation	Description
$\mathcal{L}$	The trajectory point set of one node
ℓ_i	The *i-th* trajectory point
(x_i, y_i)	The location coordinate of the ℓ_i
$\mathcal{R}$	The region set
r_i	The *i-th* region
$\rho(r_i)$	The density of r_i during the training phase
$n(r_i, \Delta t)$	The number of trajectory points in r_i during the period Δt.
$S(r_i)$	The area size of region r_i
δ	The density threshold for choosing hot regions
R	The communication range of RSU
Δr	The radius increment for region expansion
τ	The length of time slice
H	The hot region set
h_i	The *i-th* hot region
P	The probability of prediction
ST	The sojourn time of vehicular node
ψ	The requested content
η_ψ	The popularity of content ψ
$\eta_\psi^{(R)}$	The popularity of content ψ maintained by RSU
RSU	The Road Side Unit

A. Overview

Our CCMP scheme aims to predict a vehicle's probability of reaching hot regions based on its past trajectories by adopting PPM. A vehicle with longer sojourn time in a hot region is expected to provide more services and should be preferred as a caching node to cache popular data. The CCMP scheme mainly contains three modules: hot region generation, caching node selection, and cache replacement policy as illustrated in Fig. 2. Hot region generation and caching node selection are both completed in the training stage. The regions with the higher density of trajectory points are regarded as hot regions (i.e, they are most frequently visited in the network). Our hot region generation includes three phases, partition, expansion, and collapse. After hot regions are identified, we perform caching node selection to choose the best nodes that have the highest chance of visiting the hot regions and staying there for a longer time. The probability of each node arriving at hot regions is obtained by PPM prediction. An overview of how CCMP works follows:

1) Nodes record and upload their trajectory records to RSUs in the training phase as well as the actual operation phase. Such records are synchronized among all RSUs. After the training phase, those regions are chosen as the hot regions with higher density of trajectory points.
2) We use PPM to predict the future trajectory of each node based on its past traces. Only those nodes with longer sojourn time in any hot region will be chosen as the cache nodes.
3) After receiving new contents, RSUs in hot regions push them to caching nodes within the communication range.
4) Interests to contents are generated periodically and sent to neighbors. Such Interests are stored, moved, and forwarded as nodes move.
5) Upon receiving an Interest, a node searches its cache looking for a hit. If so, the content is replied toward the requesting node in a similar fashion. Otherwise, the Interest will be forwarded to the neighbors and then it will be stored, moved and forwarded until its TTL expires.
6) As Interests and contents are stored, moved, and forwarded in the network, intermediate nodes update their data structure as shown in Table II. Table II(a) keeps a record of contents. Table II(c) records information about all of those contents that are not cached locally, such as name, source ID, and expiration time. Interests are stored in Table II(b) with content name, time of last access, and popularity. RSUs do not cache contents, so they only maintain Table II(a) and Table II(b).

TABLE II
DATA STRUCTURES MAINTAINED AT EACH NODE

(a) Content Record

Notation	Description
$name$	content name
$content$	cached content

(b) Interest Record

Notation	Description
$name$	content name
$time$	time of the last request
$popularity$	content popularity

(c) Access Record

Notation	Description
$name$	content name
$source$	source ID
$time$	expiration time

B. Identifying Hot Regions

The entire network region can be divided into many different functional regions such as residential areas, business districts, educational areas, health centers, etc. Intuitively, the number of visits implicitly reflects the popularity of a certain functional region [29]. In other words, people's mobility patterns imply the visit intensity. In the CCMP, the hot regions are first identified according to vehicles' trajectory.

1) Terminologies:

Definition 1 (Mobility Trajectory): The mobility trajectory of one vehicle can be denoted by a sequence of points $\mathcal{L} = \{\ell_1, \ell_2, ..., \ell_i, ...\}$. Each trajectory point ℓ_i is composed by a triple (x_i, y_i, t_i), representing one vehicle locates at (x_i, y_i) at time t_i.

Definition 2 (Region Density): During the time interval Δt, the density of region r_i is denoted as

$$\rho(r_i) = n(r_i, \Delta t)/S(r_i), \quad (1)$$

where $n(r_i, \Delta t)$ is the number of trajectory points of all vehicles in r_i during Δt, and $S(r_i)$ is r_i's area.

Definition 3 (Hot Region): Region r_i is an independent hot region if and only if the following conditions are satisfied:

(1) $\rho(r_i) \geq \delta$.
(2) $r_i > R$.
(3) Region r_i does not overlap with any other hot regions.

2) Generating Hot Regions:

Algorithm 1 describes the function of generating hot regions, with three sub-functions:

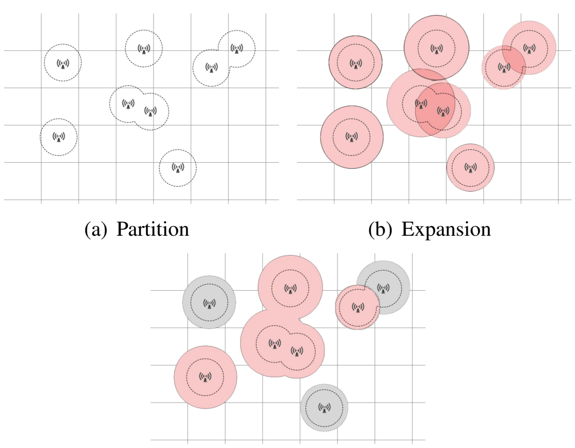
a. **Partition:** As shown in Fig. 3(a), the whole area is initially divided into squares with the side length of $2R$ on lines 2-6.
b. **Expansion:** On lines 9-19, every region whose center is an RSU keeps increasing by the increment of Δr until density $\rho(r_i)$ stops growing (see Fig. 3(b)).
c. **Collapse:** On lines 20-28, the pink region in Fig. 3(c) whose density exceeds δ will be chosen as a hot region. Fig. 3(c) also shows that any two overlapped hot regions will collapse into a bigger one if they have any intersection area.

C. Selecting Caching Nodes

In order to select caching nodes, we PPM to predict each node's probability of reaching hot regions. Before predicting a

Algorithm 1 Hot Region Selection

Input: $\mathcal{R}$
Output: H ▷ H: the hot region set
$H \leftarrow \phi$ ▷ Partition
for all $r_i \in \mathcal{R}$ **do**
  $r(r_i) \leftarrow R$
  $compute \quad n(r_i)$
  $\rho(r_i) \leftarrow n(r_i)/S(r_i)$
end for
for all $r_i \in \mathcal{R}$ **do**
  $\rho \leftarrow \rho(r_i)$
  while $\rho(r_i) \geq \rho$ **do** ▷ Expansion
    $\rho \leftarrow \rho(r_i)$
    $R(r_i) \leftarrow R(r_i) + \Delta r$
    ▷ $R(r_i)$: the radius of region r_i
    $n(r_i) \leftarrow n(r_i) + \Delta n$
    ▷ Δn: the trajectory number in extended area
    $\rho(r_i) \leftarrow n(r_i)/S(r_i)$
  end while
  if $R(r_i) > R$ **then**
    $R(r_i) \leftarrow R(r_i) - \Delta r$
  end if
  if $\rho(r_j) \geq \delta$ **then** ▷ Collapse
    for all $r_j \in H$ **do**
      if r_i overlaps with r_j **then**
        $r_i \leftarrow r_i \bigcup r_j$
        $H \leftarrow H - r_j$
      end if
    end for
    $H \leftarrow H \bigcup r_i$
  end if
end for



(a) Partition (b) Expansion

(c) Collapse

Fig. 3. An example of hot region selection

vehicle's future trajectory, a trip sequence should be collected periodically in our CCMP: each vehicle records its trajectory point at each sample time. When a vehicle encounters an RSU, it sends its trajectory record. In our simulation, the initial trajectory of each vehicle is collected during the training period.

1) Prediction by PPM:

PPM is one of the important techniques in variable order Markov models extending from the well-known Markov chain models. PPM has been demonstrated to outperform other techniques in sequence predictions [30]. In [31], PPM has also shown its advantages in the prediction of future location. During the training stage, the time each vehicle arrives at or leave places and its current location are recorded. The experiments have proved that PPM can provide a higher accuracy of predicting the user's future location. Consequently, we adopt PPM to predict a vehicle's future location in the CCMP.

In general, there is a learning phase in PPM to process known sequences. The learning phase has two stages, trie construction and escape mechanism.

Trie construction starts with a root node corresponding to an empty sequence. The training sequence is then processed one symbol at a time. Each node except the root in the trie represents a symbol and a counter, denoted by $(x, M(x))$, where $M(x)$ represents the number of times that x appears in all D-sized contexts. Each parsed element x and its D-sized context s form a path in the trie, T. If the path has not appeared before, it will be added into T. Otherwise, all the counters along the path will be incremented. At last, the number of times that every node appears along this path constitutes $M(sx)$.

An example is shown in Fig. 4, suppose a sequence $\mathcal{L} = \{\ell_1, \ell_2, \ell_3, \ell_1, \ell_4, \ell_1, \ell_5, \ell_1, \ell_2, \ell_3, \ell_1\}$ is trained to construct a trie. The first symbol ℓ_1 and its 2-sized context ℓ_2, ℓ_3 form the first path in T. Because $\mathcal{L}$ also contains subsequences ℓ_1, ℓ_4, ℓ_1 and ℓ_1, ℓ_5, ℓ_1, two other branches are added below ℓ_1. Other subsequences starting with symbols $\ell_2, \cdots, \ell_5$ are added in a similar way. Overall, ℓ_1 appears 4 times as the leading symbol.

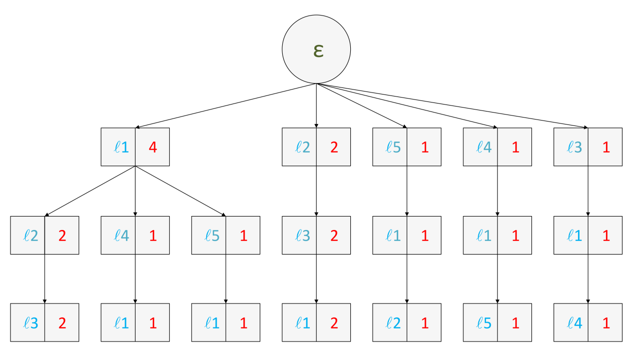


Fig. 4. Illustration of trie construction based on a sequence $\mathcal{L} = \{\ell_1, \ell_2, \ell_3, \ell_1, \ell_4, \ell_1, \ell_5, \ell_1, \ell_2, \ell_3, \ell_1\}$.

The **escape mechanism** is then used to compute probability $P_k(\xi|s)$, denoting the probability that the characters do not appear after context s of length k ($k \leqslant D$). The probability that other characters do appear after s is then $1 - P_k(\xi|s)$. Prediction on the chance of symbol x appearing after s can be computed as:

$$P(x|s) = \begin{cases} P_D(x|s) & x \in \sum_s \\ P_D(\xi|s) \cdot P(x|s') & \text{otherwise} \end{cases}, \qquad (2)$$

where $\sum_s$ is the set of characters appearing after context s in the training sequence and s' is the suffix of s [32]. For the empty context ϵ, $P(x|\epsilon) = 1/|\sum|$ holds, where $\sum$ represents the set of all the characters appearing in the training sequence. For each sequence s and character x, let $M(sx)$ represent the number of times of sx appearing in the training sequence. We can obtain the following equations,

$$P_k(\xi|s) = \frac{|\sum_s|}{|\sum_s| + \sum\limits_{q\in\sum_s} M(sq)} \tag{3}$$

and

$$P_k(x|s) = \frac{M(sx)}{|\sum_s| + \sum\limits_{q\in\sum_s} M(sq)}, \tag{4}$$

where $k = |s|$ holds. Then, we can compute probability $P(x|s)$ by applying Eq. (3) and Eq. (4) into Eq. (2). Then, Markov chain will be used for prediction based on Eq. (2).

Given a segment of mobility trajectory $\ell_3\ell_1$ and the subsequent trajectory ℓ_5 in Fig. 4, we will compute the probability $P(\ell_5|\ell_3\ell_1)$ with Eq. (2). In the case of no training set $\{\ell_3\ell_1\ell_5\}$, $P(\ell_5|s) = P(\xi|s) \cdot P(\ell_5|s')$ holds, where s is $\ell_3\ell_1$. Since s' is the suffix of s and s' is ℓ_1, we get $P(\ell_5|\ell_3\ell_1)$, the probability of one vehicle's arriving at ℓ_5 from Eq. (3) and Eq. (4).

$$P(\xi|\ell_3\ell_1) = \frac{|\sum_{\ell_3\ell_1}|}{|\sum_{\ell_3\ell_1}| + \sum\limits_{q\in\sum_{\ell_3\ell_1}} M(\ell_3\ell_1 q)} = 0.5,$$

$$P(\ell_5|\ell_1) = \frac{M(\ell_1\ell_5)}{|\sum_{\ell_1}| + \sum\limits_{q\in\sum_{\ell_1}} M(\ell_1 q)} = 1/7,$$

$$P(\ell_5|\ell_3\ell_1) = P(\xi|\ell_3\ell_1) \cdot P(\ell_5|\ell_1) \approx 0.07.$$

Algorithm 2 lists the process of predicting the future trajectory. Lines 2-6 list the learning phase of PPM. PPM constructs a trie data structure from the extracted trip sequence. Lines 8-16 list the escape mechanism of PPM to compute the probability of arriving at one region.

Algorithm 2 comp_prob Function

Input: $T(v_i), t_{k+1}, h_j$
Output: P ▷ P: probability of v's reaching hot region h
1: $P \leftarrow 0$
2: $\ell_a \leftarrow \ell_{k-1}(t_{k-1})$, $\ell_b \leftarrow \ell_k(t_k)$
3: **for all** $\ell_n \in h$ **do** ▷ at time slice t_{k+1}
4: $P_n \leftarrow 0$
5: **if** $\ell_a\ell_b\ell_n$ exists in T **then**
6: $P_n \leftarrow P(\ell_n|\ell_a\ell_b)$ ▷ applying Eq. (4)
7: **else if** $\ell_a\ell_b$ exists in T and $\ell_b\ell_n$ exists in T **then**
8: $P_n \leftarrow P(\xi|\ell_a\ell_b) * P(\ell_n|\ell_b)$ ▷ applying Eq. (3)(4)
9: **end if**
10: $P \leftarrow P + P_n$
11: **end for**

2) Caching nodes selection:

Considering that one vehicle with longer sojourn time in a hot region is expected to be able to provide more services for other vehicles, we use sojourn time to choose caching nodes. The expected sojourn time $\overline{ST_i}$ of vehicle v_i is computed with $\overline{ST_i} = \sum_{h_j\in H} P_j \cdot t_j$, which is listed on lines 2-15 of Algorithm 3, where h_j denotes the hot region, P_j is the predicted probability of reaching h_j, and t_j represents the sojourn time in h_j. The vehicles are sorted in descending order of the sojourn time. Only vehicles with longer sojourn times are preferred to serve as caching nodes.

Algorithm 3 shows the procedure of selecting caching nodes. The sojourn time of each vehicle is calculated on lines 2-15, while line 6 the comp_prob function as listed in Algorithm 2 is called to compute the probability of each vehicle reaching a specific region. And lines 16-19 show that only vehicles with longer sojourn time will be chosen as caching nodes.

Algorithm 3 Caching Node Selection

Input: V, $\mathcal{L}$, H ▷ V: the vehicle set
Output: C ▷ C: the caching node set
1: $C \leftarrow \phi$
2: **for all** $v_i \in V$ **do**
3: $ST_i \leftarrow 0$
4: $P_{max} \leftarrow 0$
5: $T(v_i) \leftarrow \phi$ ▷ Trie Construction
6: **for all** t_k in TrainingStage **do**
7: $T(v_i) \leftarrow T(v_i) \bigcup \mathcal{L}[v_i][t_k]$
8: **end for**
9: $T(v_i) \leftarrow InitTrieTree(T(v_i))$
10: **while** t_{k+1} **do** ▷ prediction
11: **for all** $h_j \in H$ **do**
12: $P_j \leftarrow comp\_prob(T(v_i), t_{k+1}, h_j)$
13: ▷ escape mechanism in Algorithm 2
14: **if** $P_j > P_{max}$ **then**
15: $P_{max} \leftarrow P_j$
16: $r_k \leftarrow h_j$
17: **end if**
18: **end for**
19: ▷ compute sojourn time
20: **if** $k > 1$ and $r_{k-1} == h_j$ **then**
21: ▷ v_i remains in the same region
22: $ST_i \leftarrow ST_i + P_j * \tau$
23: **else**
24: ▷ v_i moves to another region
25: $ST_i \leftarrow ST_i + P_j * (\tau/2)$
26: **end if**
27: **end while**
28: **end for**
29: ▷ select caching node
30: $V \leftarrow Sort(V)$
31: **for** $i = 0 \rightarrow \theta * Size(V)$ **do** ▷ θ: the caching node ratio
32: $C \leftarrow C \bigcup V[i]$
33: **end for**

D. Cache Replacement

Since caching storage can only fit some contents but not all, we discuss cache selection and replacement in this subsection. Our cache replacement is generally based on content popularity. Each node observes all the Interests and deliveries passing through itself and records such information. For each Interest/delivery, the source node's ID, content name, and request time are recorded. When a new Interest/delivery is observed, a content popularity, η_ψ, is updated for each named content:

$$\eta_\psi \Leftarrow \eta_\psi \cdot e^{-\lambda \Delta t} + \beta \ , \tag{5}$$

where λ is an exponential decay constant and β is the popularity increase constant. When a node receives a request for the content ψ, it will compute the time interval Δt between the current time and the last time of receiving the same request on ψ. Popularity should decrease exponentially with time δt. Hence, the popularity is defined as $e^{-\lambda \Delta t}$. This equation considers both frequency and freshness of the request.

RSU nodes have better chances to observe content Interests and deliveries. Therefore, we design RSU nodes to maintain a similar content Interest, $\eta_\psi^{(R)}$, which is updated among all RSU nodes through Internet connections. Essentially, $\eta_\psi^{(R)}$ is updated in a similar fashion, but it has a different impact on the vehicles:

When a vehicle travels close to a RSU node, it will retrieve $\eta_\psi^{(R)}$ from the RSU and update its own content popularity with:

$$\eta_\psi \Leftarrow \alpha \cdot \eta_\psi + (1-\alpha) \cdot \eta_\psi^{(R)} \ , \tag{6}$$

where α is the weight ratio. Compared with each vehicle, the RSU can have a more complete observation on different Interests/deliveries. Consequently, we set $0 < \alpha < 0.5$ to give $\eta_\psi^{(R)}$ a higher weight.

Algorithm 4 Cache Replacement Scheme

Input: ψ ▷ content requested
v_i receives a packet that contains content ψ
v_i updates AR
if ψ exists in IR **then**
  $\eta_\psi \leftarrow \eta_\psi \cdot e^{-\lambda \Delta t} + \beta$
else
  $\eta_\psi \leftarrow \beta$
end if
if ψ is not in CR **then**
  if CR is not full **then**
    v_i stores ψ in CR
  else if CR is full and $\eta_\psi > \eta_{min}$ in CR **then**
    v_i removes the content with η_{min} from CR
    v_i sotres ψ in CR
  end if
end if
if v_i meets a RSU **then**
  v_i retrieves $\eta_\psi^{(R)}$ from RSU
  $\eta_\psi \leftarrow \alpha \cdot \eta_\psi + (1-\alpha) \cdot \eta_\psi^{(R)}$
end if

V. Performance Evaluation

In order to evaluate the performance of our caching scheme, we conducted our experiment based on the Opportunistic Network Environment (ONE) simulator [33], which is specially implemented for opportunistic networks. In our design, there are 300 nodes, including 30 RSUs and 270 users (50 people, 100 buses, and 120 taxis), distributed in the map. All people follow Working-Day-Movement Model [34] with daily routines which mainly consist of staying at home, working in office, and shopping in markets, etc. In addition, we assume that the chance for a person owning a car is p_v, or s/he must take the bus to reach different destinations. Buses follow a few fixed driving routes throughout the map, and taxis drive as the Random-waypoint Model. All the nodes are with the same caching buffer size, movement speed range, and transmission range and data rate. First of all, we collected the location and communication records of vehicles in 5 working weeks in the training stage and applied them to different caching schemes for training. In our training stage, we start collecting trajectory records after the first 36 hours. Then, we use the three latest records that have just been collected and all of the trajectory records collected through the training stage to predict its trajectory. Next, in the actual evaluation, every person periodically generated queries following the Zipf's law distribution [35], $f(k;s,N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)}$, where k is the rank of elements, s is the exponent parameter, N is the number of elements (number of contents in our paper). We list important simulation parameters in Table III.

TABLE III
Simulation parameters

Parameter description	Value
Caching Buffer	1000MB
Query Interval	[50minutes, 100minutes]
Message TTL	10 minutes
Network Area	10km*7.5km
Simulation Time	15weeks
Vehicle Speed	[7m/s, 10m/s]
RSU Transmission Range	500m
Vehicle Transmission Range	50m
RSU Transmission Speed	10Mbps
Vehicle Transmission Speed	2Mbps
Vehicle ratio (p_v)	0.4
Popularity Weight (α)	0.3
Popularity Increment (β)	0.1
Exponential Decay (λ)	0.0005

Specifically, caching node ratio is set to 0.4, content size is set to 20MB unless specified otherwise.

A. Performance Metrics

Our main comparisons were made between the proposed CCMP scheme, LDCC [23], NCLCC [4], PNCC [22] and DPC [20]. Our CCMP scheme predicts a vehicle's future locations using PPM and selects caching nodes based on their probability of arriving at the hot regions and their sojourn time. We thus choose a prediction-based scheme LDCC as a reference. LDCC prefers nodes with highest probability staying in a valid scope. NCLCC is to intentionally cache data at a set of chosen Network Central Locations (NCLs), serving as the best central locations to cache popular contents. PNCC computes opportunistic path weight and centrality of nodes as parts of caching node selection elements. PNCC considers the rate of nodes querying data items to select the cache nodes. In DPC, the caching decisions are taken by each node separately and independently, which considers users' demands mined from collected interest entries, and the importance of vehicles in the ego network.

Performance metrics are listed below:

- **Success Ratio**, the ratio of successful queries that generate requested contents among all queries.
- **Average Access Delay**, the average delay of obtaining responses in successful queries.
- **Standard Deviation of Storage Usage** σ, the standard deviation of storage usage among all nodes in the network.
- **Average Storage Usage,** Γ_B**,** Γ_T**,** Γ_C**,** Γ_O, the average storage usage of different nodes: Bus, Taxi, users with Cars, and users withOut cars, respectively, in the network.

B. Performance Comparison

In Fig. 5, we compare success ratio, average access delay, and standard deviation of storage usage among CCMP, DPC, PNCC, LDCC, and NCLCC with different caching node ratios. As caching node ratio increases, success ratios of all five schemes improve. CCMP enjoys the highest success ratios among all schemes. Meanwhile, Fig. 5(b) shows that, with more caching nodes, the average access delays of all schemes decline, as a result of easier access to caching contents. In Fig. 5(c), the interesting convex shape of standard deviation might have suggested an "optimum" caching node ratio, in which success ratio is close to the best and a collection of "core" nodes are used to cache popular contents. Compared with other four schemes, our CCMP scheme achieves better success ratio (up to 28% gain), shorter access delay (about 21% lower), and slightly more diversed storage usage.

In Fig. 6, we compare success ratio and average access delay of these schemes with different content sizes. As the content size increases, each node caches fewer contents, lowering success ratios and increasing access delays. The superior performance of CCMP is clearly shown.

Storage usage of different types of nodes in the network is shown in Fig. 7. We only compare CCMP, NCLCC, and LDCC because NPCC and DPC have been shown to be much inferior than other three in Fig. 5 and Fig. 6. Note that the results in Fig. 7 have been multiplied with the ratios of these types of nodes in networks, so the overall height of each column represents the average cache storage usage. Our first observation is that more storage is consumed as the content size increases. Secondly, LDCC relies heavily on taxi (Γ_T). This should have been caused by the predictive method of identifying caching nodes with better chances to visit different regions. Furthermore, CCMP consumes slightly more storage space than NCLCC and LDCC, some of which are caused by the interesting usage of users without cars (Γ_O). With such a slight increase in cache storage usage, CCMP is able to achieve about 15% gains in success ratio (see Fig. 6(a)).

We evaluate CCMP's success ratio with different p_v in Fig. 8. As p_v increases, more users have personal vehicles and, thus, their mobility patterns are less predictable, lowering success ratios for CCMP. A network scenario with different number of people (75-people: a total of 325 nodes, including 30 RSUs and 295 users: 75 people, 100 buses, and 120 taxis) is also shown in Fig. 8. With an increased number of people in the network, CCMP's success ratio actually increases, underlying its capability of taking advantages of different nodes' mobility patterns.

TABLE IV
IMPACT OF CONTENT NUMBER

Content Number	Success Ratio
100	0.241
1,000	0.167
10,000	0.134
100,000	0.113

On Table IV, we show the success ratio of different content numbers. It can be seen that success ratio decreases as the number of contents increases. This can be explained with the increasing difficulties of finding hits with more contents. However, even as the number of contents increases by a 100-fold, the successful ratio of our CCMP scheme is only reduced to about a half, underlining its robustness.

VI. CONCLUSION

In this work, we have proposed a novel scheme, termed CCMP, to support the content caching, request, and delivery in VCCN. Utilizing the trajectory history records of different vehicles, CCMP is able to predict the probability of their next visits to different hot regions in the area. Caching nodes are chosen based on the sojourn time and caching contents are decided from their popularity. We have presented our extensive simulations on the CCMP scheme and compared it with several other state-of-the-art schemes in success ratio, average access delay, and storage usage. It has been shown that CCMP enjoys a higher success ratio and lower access delay and can be a strong candidate for VCCNs. In our future work, we plan to investigate more accurate predictions of mobility patterns and use such predictions to enhance caching decisions.

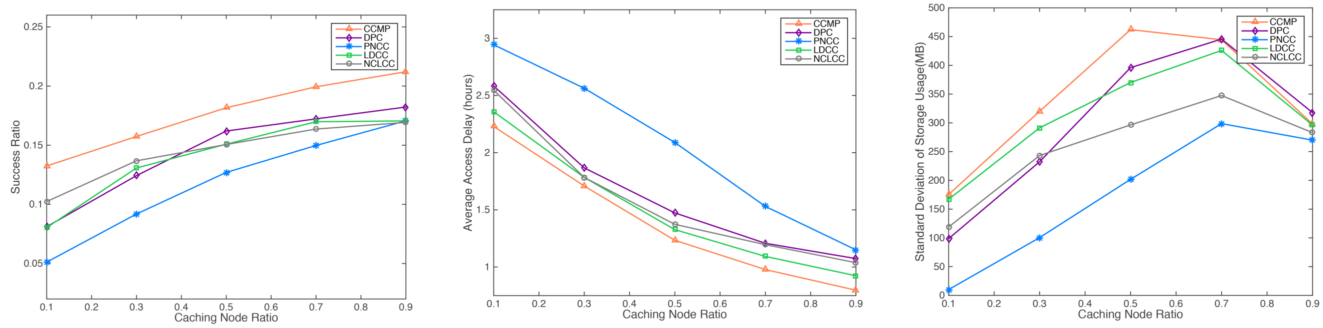
VII. ACKNOWLEDGMENT

This research is sponsored in part by the National Natural Science Foundation of China (contract/grant number: No.61672129, No.61772113). This research is also sponsored in part supported by the Fundamental Research Funds for the Central Universities (DUT16QY41). Deng's research was supported in part by NSF grant CCF-1320428.

REFERENCES

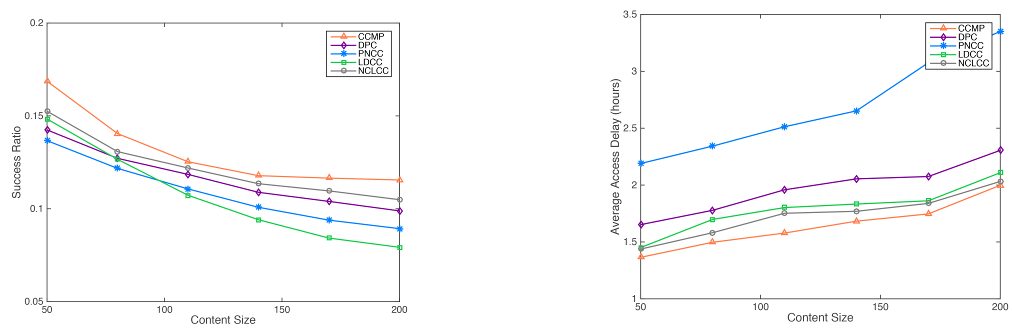
[1] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, "Content-centric wireless networking: A survey," *Computer Networks*, vol. 72, no. 7, pp. 1–13, 2014.

[2] S. H. Ahmed, S. H. Bouk, M. A. Yaqub, and D. Kim, "CONET: Controlled data packets propagation in vehicular named data networks," in *IEEE Consumer Communications and NETWORKING Conference*, 2016.



(a) Success Ratio (b) Average Access Delay (c) Standard Deviation of Storage Usage

Fig. 5. Impact of caching node ratio on different schemes



(a) Success Ratio (b) Average Access Delay

Fig. 6. Impact of content size on different schemes

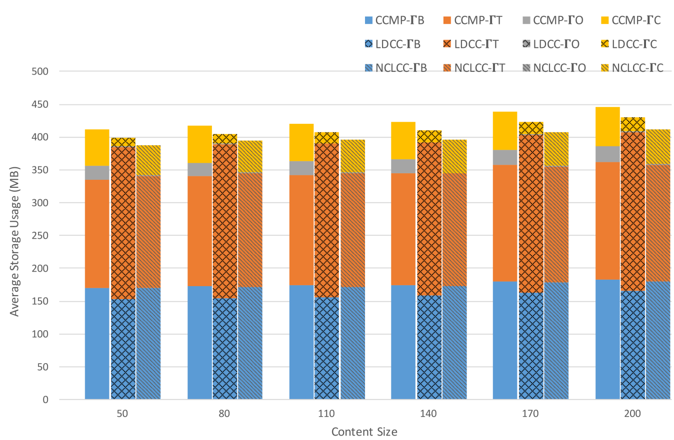


Fig. 7. Storage usage of different types of nodes in CCMP, LDCC, and NCLCC

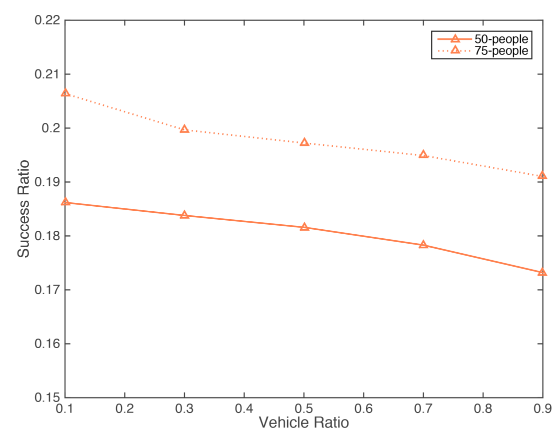


Fig. 8. Impact of vehicle ratio p_v

[3] T. H. Luan, S. Shen, and F. Bai, *Enabling Content Distribution in Vehicular Ad Hoc Networks*. Springer, 2014.

[4] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, "Cooperative caching for efficient data access in disruption tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 611–625, 2014.

[5] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1–1, 2015.

[6] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-AS cooperative caching for content-centric networks," in *ACM SIGCOMM Workshop on Information-Centric NETWORKING*, 2013, pp. 61–66.

[7] I. Psaras, K. C. Wei, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 11, pp. 2920–2931, 2014.

[8] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks (extended version)," *Computer Communications*, vol. 36, no. 7, pp. 758–770, 2013.

[9] S. Saha, A. Lukyanenko, and A. Ylä-Jääski, "Cooperative caching through routing control in information-centric networks," *Proceedings - IEEE INFOCOM*, vol. 12, no. 11, pp. 100–104, 2013.

[10] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga, "CATT: potential based routing with content caching for ICN," in *Edition of the Icn Workshop on Information-Centric NETWORKING*, 2012, pp. 81–86.

[11] D. Rossi and G. Rossini, "On sizing CCN content stores by exploiting topological information," *Proceedings - IEEE INFOCOM*, pp. 280 – 285, 2012.

[12] V. Sourlas, L. Gkatzikis, P. Flegkas, and L. Tassiulas, “Distributed cache management in information-centric networks,” *IEEE Transactions on Network & Service Management*, vol. 10, no. 10, pp. 286–299, 2013.

[13] M. Detlefsen and M. Luker, “WAVE: Popularity-based and collaborative in-network caching for content-oriented networks,” *Proceedings - IEEE INFOCOM*, vol. 11b, no. 12, pp. 316–321, 2012.

[14] S. Wang, J. Bi, and J. Wu, “Collaborative caching based on hash-routing for information-centric networking,” *ACM Sigcomm Computer Communication Review*, vol. 43, no. 4, pp. 535–536, 2013.

[15] Y. Li, T. Lin, H. Tang, and P. Sun, “A chunk caching location and searching scheme in content centric networking,” in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 2655 – 2659.

[16] S. Ioannidis, L. Massoulie, and A. Chaintreau, “Distributed caching over heterogeneous mobile networks,” *Queueing Systems*, vol. 72, no. 3-4, pp. 311–322, 2010.

[17] X. Zhuo, Q. Li, G. Cao, Y. Dai, B. Szymanski, and T. L. Porta, “Social-based cooperative caching in dtns: A contact duration aware approach,” in *IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011, pp. 92–101.

[18] T. Le, Y. Lu, and M. Gerla, “Social caching and content retrieval in disruption tolerant networks (DTNs),” in *International Conference on Computing, NETWORKING and Communications*, 2015, pp. 905–910.

[19] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, “Supporting cooperative caching in disruption tolerant networks,” in *2011 31st International Conference on Distributed Computing Systems*, 2011, pp. 151–161.

[20] D. Gang, W. Liwei, L. Fengchao, and L. Rere, “Distributed probabilistic caching strategy in VANETs through named data networking,” in *IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2016, San Francisco, CA, USA, April 10-14, 2016*, 2016, pp. 314–319.

[21] I. Psaras, W. K. Chai, and G. Pavlou, “Probabilistic in-network caching for information-centric networks,” in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 55–60.

[22] F. Tian, B. Liu, J. Xiong, and L. Gui, “Efficient caching scheme for data access in disruption tolerant networks,” in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2015, pp. 1–5.

[23] E. Chan, Y. Wang, W. Li, and S. Lu, “Movement prediction based cooperative caching for location dependent information service in mobile Ad hoc networks,” *Journal of Supercomputing*, vol. 59, no. 1, pp. 297–322, 2012.

[24] J. Luo, X. Gu, T. Zhao, and W. Yan, “Mi-vanet: A new mobile infrastructure based vanet architecture for urban environment,” in *Vehicular Technology Conference Fall*, 2010, pp. 1–5.

[25] X. Jiang and D. H. C. Du, “Bus-vanet: A bus vehicular network integrated with traffic infrastructure,” *Intelligent Transportation Systems Magazine IEEE*, vol. 7, no. 2, pp. 47–57, 2015.

[26] W. K. Lai, K. T. Yang, and M. C. Li, “Bus assisted connectionless routing protocol for metropolitan vanet,” in *Fifth International Conference on Genetic and Evolutionary Computing*, 2011, pp. 57–60.

[27] C. R. Dow, P. Hsuan, Y. H. Lee, Y. T. Lee, and C. Y. Huang, “An efficient data circulation and discovery scheme in vanets using public transportation systems,” in *International Conference on Network and Service Management*, 2011, pp. 286–289.

[28] J. Wang, K. Liu, P. Dai, H. M. Sha, L. Feng, C. Chen, and C. Xiao, “Co-operative information services based on predictable trajectories in bus-vanets,” in *IEEE International Conference on High PERFORMANCE Computing and Communications; IEEE International Conference on Smart City; IEEE International Conference on Data Science and Systems*, 2017, pp. 951–958.

[29] J. Yuan, Y. Zheng, and X. Xie, “Discovering regions of different functions in a city using human mobility and pois,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 186–194.

[30] R. Begleiter, R. El-Yaniv, and G. Yona, “On prediction using variable order markov models,” *arXiv preprint arXiv:1107.0051*, 2011.

[31] F. D. N. Neto, “Predicting routes and destinations of urban trips using PPM method,” in *Simpsio Brasileiro de Computao Ubqua e Pervasiva*, 2015.

[32] J. Yang, J. Xu, M. Xu, N. Zheng, and Y. Chen, “Predicting next location using a variable order markov model,” in *ACM Sigspatial International Workshop on Geostreaming*, 2014, pp. 37–42.

[33] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in *International Conference on Simulation TOOLS and Techniques for Communications, Networks and Systems, Simutools 2009, Rome, Italy, March*, 2009.

[34] F. Ekman, A. Keränen, J. Karvo, and J. Ott, “Working day movement model,” *Acm Wmm*, pp. 33–40, 2008.

[35] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: evidence and implications,” *Proceedings - IEEE INFOCOM*, vol. 1, pp. 126 – 134, 1999.

Lin Yao is an associate professor in School of Software, Dalian University of Technology (DUT), China. Her research interests include security and privacy of VANETs, CCN, and WSN.

Ailun Chen is an M.E. candidate in School of Software, Dalian University of Technology. Her research interests include security and privacy in mobile social network.

Jing Deng (S’98-M’02-SM’13-F’17) received B.E. and Ph.D. degrees from Tsinghua University, China, in 1994 and 2002, respectively. Dr. Jing Deng is an associate professor in the Department of Computer Science (CS) at the University of North Carolina at Greensboro (UNCG). Dr. Deng is an editor of IEEE Transactions on Vehicular Technology (TVT). His research interests include wireless network security, information assurance, mobile ad hoc networks, and online social networks.

Jangbang Wang is a M.Sc. candidate in Faculty of Engineering, National University of Singapore. His research interests include blockchain and security in vehicular ad-hoc network.

Guowei Wu received his Ph.D. degree from Harbin Engineering University, PR China. He is now a professor at the School of Software, Dalian University of Technology (DUT). His research interests include embedded real-time system, cyber-physical systems (CPS), and wireless sensor networks.