Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8374181, 11 pages
https://doi.org/10.1155/2022/8374181

Research Article

Dynamic Cooperative Cache Management Scheme Based on Social and Popular Data in Vehicular Named Data Network

M. Wasim Abbas Ashraf,[1] **Chuanhe Huang**,[1] **Khuhawar Arif Raza**,[2] **Shidong Huang**,[1] **Yabo Yin**,[1] **and Dong-Fang Wu**[1]

[1]*School of Computer Science, Wuhan University, Wuhan 430072, China*
[2]*School of Computer Science and Engineering, SUSTech, Shenzhen 518055, China*

Correspondence should be addressed to Chuanhe Huang; huangch@whu.edu.cn

Received 1 December 2021; Revised 27 February 2022; Accepted 3 March 2022; Published 22 March 2022

Academic Editor: Changqing Luo

Vehicular Named Data Network (VNDN) is considered a strong paradigm to deploy in vehicular applications. In VNDN, each node has its cache, but due to limited cache, it directly affects the performance in a highly dynamic environment, which requires massive and fast content delivery. To reduce these issues, the cooperative caching plays an efficient role in VNDN. Most studies regarding cooperative caching focus on content replacement and caching algorithms and implement these methods in a static environment rather than a dynamic environment. In addition, few existing approaches addressed the cache diversity and latency in VNDN. This paper proposes a Dynamic Cooperative Cache Management Scheme (DCCMS) based on social and popular data, which improves the cache efficiency and implements it in a dynamic environment. We designed a two-level dynamic caching scheme, in which we choose the right caching node that frequently communicates with other nodes, keep the copy of the most popular content, and distribute it with the requester's node when needed. The main intention of DCCMS is to improve the cache performance in terms of reducing latency, server load, cache hit ratio, average hop count, cache utilization, and diversity. The simulation results show that our proposed DCCMS scheme improves the cache performance than other state-of-the-art approaches.

1. Introduction

With the rapid development in the vehicular population [1], many applications and services are merged to provide desired data to the requester's vehicle. In this concern, the VANETs become in this race with the modifying version of Mobile Ad hoc Networks (MANETs) [2], which is aimed at improving the communication in moving vehicles and covering the large-scale traffic environment. Increasing wireless content traffic has a great burden on VENETs. As per the report of [3], 60% of content belongs to video or audio. Due to the growing size of data traffic and maintaining audio or video streaming, many studies have proposed solutions [4]. In these studies, content delivery is one of the bottlenecks in VANETs due to the dynamic environment and frequent disconnect connections between different nodes. The future paradigm Named Data Network (NDN) brings efficient services [5], which are more concerned about data. NDN works the terms "What" rather than "Where." Besides, in NDN, every node has cache and three data structures: (1) Content Store (CS), (2) Pending Information Table (PIT), and (3) Forward Information Table (FIB). Therefore, the VANETs turned into Vehicle Named Data Networks (VNDN) due to its incredible services. VNDN has been proposed as a key promising solution, especially for mobile network caching, and VNDN plays a vital role in monitoring traffic issues and supports road safety [6] applications.

The main focus of VNDN is content distribution between nodes in the network. In traditional vehicular networks (VNs), a specific IP address [7] is allocated to every node for information sharing to another node. Moreover, a secured channel must connect with the source and destination sides to transmit confidential content. So, traditional networks bring the mobility management issue to transfer the secured data between nodes. In particular VNDN environment, each node is self-detecting via content name and self-authorize. The consumer and their neighboring nodes probably travel in the same direction, and with the same speed, the link could be more stable and reliable. Moreover, the popular content may be fetched easily and may reduce the mobility issues in VNDN. Recently, the vehicular network caching manner is considered noncooperative in [8]. The caching decision is based on local demand rather than global demand, which may be difficult for the consumer to access desired data from the producer (nearby nodes) and raise the mobility and access delay issue.

Although VNDN has many advantages, the VNDN in dynamic traffic brings many issues such as network performance, cache miss, data delay, data loss, mobility, and resource managing. Figure 1 illustrates the following data access challenges in VANET scenario.

(i) Let us assume that at the time t_0 the $V1$ sends the request, and the forwarding distance between $V1$ and $V3$ is 120 m. The $V1$ is moving at 60 km/h while $V3$ is moving with a speed of 80 km/h. When the desired content is reached at $V3$ from $V4$, unfortunately, the $V1$ is moved out from the transmission range of $V3$. This happened due to a noncooperative method and the mobility speed of $V3$; as a result, $V1$cannot receive the desired corresponding data due to the large distance from V3.

(ii) Suppose the $V1$ transmission range is larger than $V3$. In this situation, the $V1$ may reach near $V3$, and $V3$ forwards the request from $V1$. In this case, the $V3$ again failed to deliver the desired corresponding data to $V1$

(iii) The network faced delays in data, data loss, and cache missing due to ineffective cache management strategy in VNDN

To enhance the performance of VNDN, the cooperative edge cache methods are introduced in [9] to improve the hit ratio of cache, network traffic management, and data access latency as per results with other baselines strategies.

A caching strategy and social interactions are considered to improve the better communication, secured, and entertainment in VANET's applications [10]. In this study, the cooperative caching methods are divided into two groups: (i) on-path cache and (ii) off-path cache based on where content is cached. In the on-path method, content is cached via some nodes using the reverse path technique from the consumer to the producer, while in the off-path method, some specific nodes are involved to cache the content for the consumer. According to the author, nodes with the same interests can make better connectivity links and efficient dissemination in vehicular networks. Many strategies are proposed to improve the cache performance in literature, but fewer cache methods are implemented in dynamic scenarios [11]. The existing strategies are based on the fixed probability that increases content redundancy and maximizes hop count distance and others based on content popularity which less popular content ignores, resulting in low diversity. In the light of the above discussion, an excellent caching management strategy is required to mitigate these issues. Therefore, introduce the Dynamic Cooperative Cache Management Scheme Based on Social and Popular Data in VNDN. The main focus of DCCMS is to improve the cache performance via popular data and social interaction in VNDN and build a reliable communication link among nodes based on master node. We design the master node that acts hierarchically and collaborates with neighboring nodes to keep the most popular contents in the cache. Our method uses maximum available cache resources to minimize the latency and improve the cache hit ratio performance. The master node is aimed at improving the overall network performance by keeping the popular content for a long time and sharing content with the requester as per demand. The existing strategies keep recent popular contents individually rather than the whole network. Moreover, social interaction between nodes is necessary to keep the complete history of popular contents in the whole network and evict it at a specific time slot. The main contributions of our paper are listed as follows:

(i) We present a novel two-level caching scheme that distributes popular content in VNDN networks

(ii) The master node is introduced to fulfill the future request from nearby nodes. The master node stores the content which cannot be accommodated from nearby nodes due to limited storage capacity. Moreover, the master node supports social interaction with other nodes to get the copies of the most popular contents (MPC) in the whole network rather than a single node

(iii) We design the eviction policy based on content popularity and time slot to cache contents' freshness to deal with less popular content

(iv) The comprehensive simulation shows that our scheme is good for dynamic traffic where the primary objectives are the highest cache hit, low latency, maximum cache utilization, and efficient network performance

The rest of this paper is organized as follows. We discuss the related work in Section 2. In Section 3, the system model and problem formulation are discussed. We presents DCCMS strategy in Section 4. The comprehensive evaluation and DCCMS performance are demonstrated in Section 5, and finally, we conclude the work in Section 6.

2. Related Work

In this section, we discussed some recent studies related to our scheme. The local cache is used to minimize the computational cost in the network. On-path cache (OPC) attracts the research community's attention due to its local uses; the term cooperative caching is very famous in vehicular networks to enhance the overall network performance. In

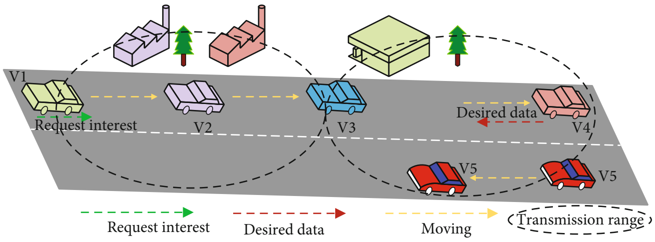


Figure 1: Simple-VNDN scenario.

[12], the authors proposed a cooperative cache to estimate the router capacity and cached the popular contents only. The collaborative caching model is based on the hierarchical method in [13] to fetch the contents from the nearest node in the network. In this proposed model, overhead occurred due to heavy network load in VNs. The authors proposed a caching strategy-based two-tier heterogeneous model to optimize the caching probability using two helpers for the edge server [14]. The IDCC [15] scheme is designed to improve cache performance. In this study, the static topology is used to get the desired results based on the probabilistic cache method. A content delivery scheme [16] is designed for social communities to park the vehicle. The vehicles are communicated using distributed content to minimize the cost of network and content latency. The content replacement strategy is discussed in [17, 18]; the content copy is moved down on each node to replace the content. Due to this method, the memory cost and redundancy will increase, and the root node cannot keep at least one copy of the content. In [19], a probability-based caching scheme is designed to get the cache capacity and root node distance. This cache decision is based on the hop count of the server and the content router size. The authors developed a framework for content distribution in [20] networking cache and across the domain. To investigate the cooperative cache scheme performance, the authors [21] introduced a fog-based social aware network and designed a content sharing scheme that allows the nodes' network to collaborate content cache locally. To improve the latency in VANET, the authors discussed the collaborative routing in [22] and, for data dissemination, choose the vehicle as edge node in [23]. In [24, 25], the authors introduced multitier cooperative cache architecture to deal with popular contents in different networks and improved the delayed hit ratio in caching. Similar approaches for heterogeneous networks are proposed in [26, 27, 28, 29], where caching nodes are introduced to reduce the content delivery ratio. Furthermore, in [30, 31], the study designed a caching scheme that incorporates the popularity of content depending on the cache decision. To determine the space and replacement content in the cache, the authors explore the idea in [32] that the proposed algorithm for popularity and replacement used static parameters. Moreover, in [33], cache strategy is introduced based on centrality, in which an intermediate node is chosen to cache the contents. The results show to improve the cache utilization and reduce content redundancy in the static network. The approach of cache consistency for ad hoc networks is introduced in [34], which used the Global Positioning System (GPS) and pull-based approach to enhance performance. The same work is discussed in [35], which used a hybrid approach to pull and push to select relay nodes between the cache and producer node. The cache nodes determine the cached data using a relay node. Many authors designed the algorithm for maintenance cache consistency and replacement content in [36, 37, 38, 39] to solve the cache consistency problems. In [40], the authors increase the performance of retrieval information from the cache in Mobile Ad hoc Networks (MANET) using location maintenance. Similar studies [41, 42] extract information based on user social interaction by base station (BS). The AlwaysCache strategies are discussed in NDN [43, 44, 45, 46], where incoming contents are always cached on all nodes without replacement. However, due to accepting every new content generates redundancy, and maximizing the memory cost in the network. The cache strategies are discussed in [47, 48] with fixed probability. According to its results, cache memory is reduced with less performance in diversity and content replication. However, the authors introduced DPC using a probabilistic strategy in [49]; this method increased the delay due to lack of collaboration with other nodes. The PCMP work is discussed in [50], where popular contents are cached at only RSU; due to this, the cache hit ratio decreased with low velocity and increased hop count with high velocity. The performance of these studies is lower than that of DCCMS when implemented in a dynamic environment to get the core objective in terms of the cache hit and latency. Therefore, we proposed DCCMS to mitigate these issues, enhance network performance regarding cache hit, and reduce latency.

3. System Model

3.1. Network Model. In this section, we discussed network model and problem formulation for VNDN.

In VNDN, the consumers can send the request to the content producer to get the desired content. The Server, RSUs, and neighboring vehicles could provide content to the consumer. Let us consider Figure 2 as a network scenario for VNDN by graph $G = (X, Y)$, where $X = \{V \cup C \cup C_{cn} \cup S\}$ indicates a set of network nodes and Y is the communication link between the nodes. A network consisting of vehicle $V = \{V_1, \cdots, V_n\}$ can request the contents $C = \{C_1, \cdots, C_n\}$ and get the desired data from the Caching nodes C_{pn} or server node s. We assume double-direction road d_r with mobility speed m_s for vehicle movement, and the transmission coverage of RSU is the one hop-based cluster, where $r_a (a \in \{1, 2, 3, \cdots, d_r\})$ and $|r_a|$ represents the clusters

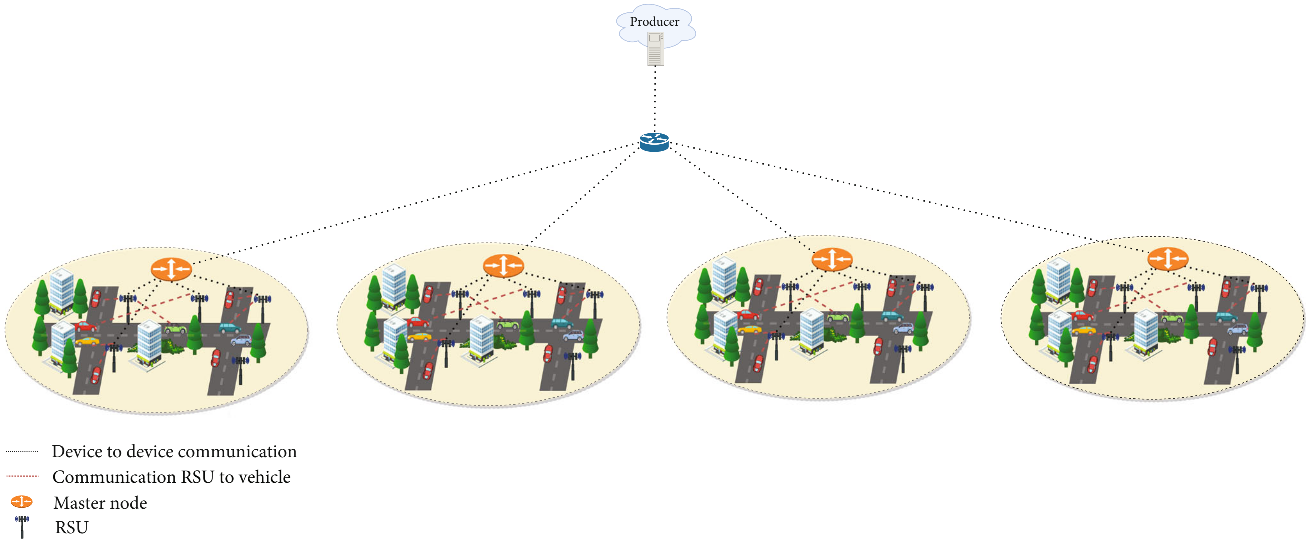


Figure 2: Network topology for DCCMS.

vehicle at $L1$. The definitions of the main symbols employed in this paper are summarized in Table 1.

3.2. Packet-Flow Structure. We designed the packet-flow structure integrated with the default NDN Type Length Values (TLVs) packet structure for unique content to perform the desired action according to our proposed scheme. Figure 3(a) shows the interest packet structure, and in Figure 3(b), Acknowledgment (Ack) attribute is added to get the information about content availability and data-chunks size.

3.3. Cache Model. We consider that all contents are defined by the Poisson process with content file $|C| = F$ files in which $C = \{C_1, C_2, \cdots, C_f\}$. Each node requests the C using the Zipf distribution for content probability. Let us assume that the $V3$ requests the random content at time t_{s_1} with the content size z. If the desired request is available in the local cache associated with $V3$, then the request will be fulfilled, and no need to send other nodes. Otherwise, the $V3$ communicates with neighboring nodes (vehicles and RSUs). If the requested content is found, the content provider's node is immediately sent to the requester's node. If the neighboring nodes do not contain them, then the content server will respond. So, fetching the content from the server affects the network performance, such as minimizing cache hit ratio, increasing latency, and minimizing cache utilization. To overcome these issues, we designed DCCMS; let us assume a binary matrix M to check whether the content is cached from the $L2$ or $L1$, where $C_{i|r_{(a+1)}|} = 1$ means that the content cached at $L1$ and 0Cache hit is the probability of cacheotherwise, it means the content is missed at $L1$. Moreover, each element takes value 0 or 1 in binary cache decision matrix, such as

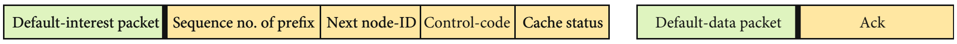
$$C_i \left| r_{(a+1)} \right| \in [0, 1]. \tag{1}$$

Table 1: Key notations.

Notation	Description
C	Numbers of content
z	Size of content
$L1$	RSU
$L2$	Master node
r_a	Numbers of road side units
S	Server node
F	Content file
F_l	Content files packet at caching node
C_p	Content popularity
T_s	Time slot
E_n	Neighboring node
ω	Cache size L_1
Q	Cache size L_2
$M_l(C_i)$	Matrix with binary value to check cached level
l	$l \in L1$ and $L2$
d_r	Double direction road
$V_c^{L1,L2}$	Vehicle cached content from RSU or master node

According to the above discussion, if the content is missed at $L1$ due to limited capacity, then the content must be cached at the $L2$ to enhance the cache performance. For more details, see Section 4.

3.4. Problem Formulation. In this section, the DCCMS is aimed at reducing the latency to get the maximum cache hit ratio. We assume that each node assumes equal contents, and the content size is normalized to 1; for example, each node in the network can cache C contents. The content popularity C_p is changed with respect to the given time slot T_s



(a) Interest packet structure

(b) Data packet structure

Figure 3: Packet-flow structure.

$= \{t_{s_1}, \cdots, T_{s_n}\}$. So, content with the higher popularity value must be detected first and cached at the neighboring nodes E_n to reduce the latency, and the other low popularity value's contents will be evicted to enhance the locality cache hit rate. The problems can be formulated as

$$\max \quad \sum_{c_i \in C} C_p(c_i) \sum_{n \in E_n} \sum_{l \in L_1 \cup L_2} \mathrm{CH}_r(c_i) M_l(c_i) \tag{2}$$

$$\text{s.t.} \quad \sum_{c_i \in C} z(c_i) M_l(c_i) \le F_l, \quad \forall l \in L_1, \tag{3}$$

$$\sum_{l \in L_1 \cup L_2} M_l(c_i) = 1, \quad \forall n \in E_n, \quad \forall c_i \in C, \tag{4}$$

$$M_l(c_i) = \{0, 1\} \quad ,\forall l \in L_1 \cup L_2, \forall c_i \in C. \tag{5}$$

Constraint in Equation (3) describes neighboring nodes' content placement according to storage limits. The desired content is cached from one node rather than multiple nodes by constraint Equation (4); if the current state $M_s(C_1) = 1$, the content will be cached at the server node. The cache decision attribute is restricted in Equation (5). According to the above discussion, we introduced the DCCMS (see in Section 4) that keeps the most popular content during T_s and distribute it with other nodes to maximize the cache hit ratio and improve the overall network performance.

4. Dynamic Cooperative Cache Strategy

4.1. Cache Management. We design the two-level cache strategy to get the desired content in VNDN. Therefore, we define the chunk-layered weight function for a cache as

$$\omega_{c_l} = e^{-c_l + \alpha / c_m}, \tag{6}$$

where c_l represents the chunks layered, c_m indicates highest value of content, and α illustrates the control variable for each packet layer. Let assume that $c_l = \{0, 1, 2, \cdots, n\}$, where 0 represents the r_a-layer packet and n^{th} is highest layer packet for C. In the VNDN network, the requester node can get the desired contents from the neighboring node ($L1$ or $L2_n$) as discussed in Algorithm 1. If the desired content is arrived at t_{s1} and t_{s2} is the slate time, then the content capacity is $z > \omega$; then, $V_c^{L1,L2} = \{0, 1\}$. The content is moved to $L2$ node either cache time expired $t_{s1} = t_{s2}$ or $L1$ has no more content capacity for new content.

$$L2_c \longleftarrow L1_{ac}, \quad \forall t_1 = t_2, \mu = n. \tag{7}$$

Moreover, the influence node considers the key node of the network that can enhance the performance, but the selection of cache nodes in dynamic traffic is very complex. However, we choose the $(L2)$ as the influence node, the details of getting the cache node are discussed in Section 4.2.

```
    Initialize:
    $C_i$ : Requested content
    $L1$ : RSU
    $L2$ : Master node
    $T_s$ : Time slot
    $Q$ : L2 cache size
    $z$ : Requested content size
    $\omega$ : L1 cache size
Output: Content cached at $l \in L1$, or $L2$
1 if $C_i \in C$ is available at L1 then
2    $C_i -> C_h = 1$ ;
       $(++\theta)$ ;
       Set $T_s$
3 else
4    (forward the request to $L2$) ;
5 if $z > \omega$ then
6    Initiate Algorithm $-$ (2) ;
7 else
8    Go to step 2 ;
9 if $C_i \in C$ is not cached at $L2$ then
10   Request forward to $S$ node
11 if Q is full then
12   Initiate Algorithm-(2)
13 End Procedure
```

Algorithm 1: DCCMS caching scheme for incoming content.

4.2. Caching Node. We choose the cache node based on node centrality methods [46] and social characteristics based on Jaccard similarity indexing to measure the most relative influence node in the network [51]. Let M be the adjacent matrix of the network. The contents (C_{i_j}) of M take 1 if the n_i connected to the n_j, otherwise 0. Moreover, the exponential sign of M shows the link between two nodes via the intermediate node. For example, in adjacent matrix M^4 if the contents $(C_{4,10}) = 1$, it shows that the n_4 and n_{10} are linked with the distance length 4. Thus, the C_{n_i} is measured as

$$C_{n_i} = \sum_{\mathrm{dc}=1}^{n} \sum_{j=1}^{n} \beta^{\mathrm{dc}} \left(M^{\mathrm{dc}}\right)_{ij}, \tag{8}$$

where C_{n_i} is the centrality of n_i, at (i, j) location of matrix M, dc indicates the numbers of degree connection between above two nodes, and β represents the weight factor value. We can also calculate the dynamic communication between the nodes as

$$\rho = \left(I_m - \beta M^{[tm0]}\right)^{-} 1, \cdots, \left(I_m - \beta M^{[tmn]}\right)^{-1}, \tag{9}$$

where I_m is the identity matrix and $0, -1$ is the time points series, $(t_{m0} < t_{m0} < 1t_{m0} < 2, \cdots, t_{mn})$. Finally, by using Equations (6) and (7), we can calculate the centrality effect of n regarding C_n^b and C_n^{rc} request dynamically in the whole network.

$$
\begin{aligned}
C_n^b &= \sum_{dc=1}^{N} \rho n \cdot \mathrm{dc}, \\
C_n^{rc} &= \sum_{dc=1}^{N} \rho n \cdot \mathrm{dc},
\end{aligned}
\tag{10}
$$

where C_n^b indicates the broadcasting and C_n^{rc} illustrates the receiving requests. Furthermore, ρ is the dynamic communication weight count at length l_n from n_i to n_j.

4.3. Content Replacement. We design Algorithm 2 for the replacement content based on content's popularity which concerns T_s. We divided the cache into two queues: (1) the most popular content (MPC) and (2) the lowest popular content (LPC). First, the consumer node checked its own CS-table. If the content is available, then cache the content and update the CS table. If the desired content is unavailable, it forwards the request to nearby nodes or caching nodes, selected based on centrality (see in Section 4.2). After receiving the desired content, if the CS is full, then the following method is used to evict the content from CS.

$$
C_{pi} = \frac{r_{c(ti)}}{t_{c(ti)}} + \theta, \tag{11}
$$

where $r_{c(ti)}$ indicates the requested content within time slot (ti) and $t_{c(ti)}$ represent the total numbers of request within giving time slot and θ is the popularity increase constant.

```
    Initialize:
    $C_i$ : Requested content
    $L1$ : RSU
    $L2$ : Master node
    $T_s$ : Time slot
    $Q$ : L2 cache size
    $z$ : Requested content size
    $\omega$ : L1 cache size
Output: Content Replaced
if $\omega$ is full then
    share copy of MPC into L2; // Using Eq.(6)
      Evict LPC;
if Q is full then
    Calculate $C_p$;
      Evicted LPC content;
      Allocate space to new content;
End Procedure
```

Algorithm 2: Cache replacement.

5. Simulation and Performance Evaluation

5.1. Simulation Scenario. We consider the bidirection road on two lanes with the 15 km pathway. The network size varies from 30 to 90 vehicles, and the velocity is set from 5 to 20 meters per second (m/s) with 20 RSUs. The simulation time is set from 100 to 1000 seconds and run 100 times to validate the results. Moreover, the simulation parameters are mentioned in Table 2.

5.2. Performance Evaluation. We evaluate the DCCMS performance using the ndnSIM [52] based on NS-3 [53]. We compare our work with the AlwaysCache [45], Prob(0.5) [48], DPC [49], and PCMP [50]. The overall results show that our proposed method DCCMS is good compared to other methods. We improve the results with the master node that collaborates with other nodes to get MPC in the whole network and distribute it when needed.

Table 2: Simulation Parameters.

Parameter description	Value
Numbers of vehicles	30-90
Vehicle speed	5-20 (m/s)
Communication range	250 m
Road propagation model	Nakagami
MAC protocol	802.11p
Routing scheme	Broadcasting
Cache replacement	DCCMS
C_p-increment	θ
Content size	10 KB
Numbers of consumer	2
Simulation time	100-1000 (sec)

5.3. Simulation Results. Cache Hit: the probability of cache hit C_h from a cache node rather than server node, which is measured as

$$
\mathrm{CH}_r = \left[\frac{\sum_{h=1}^{n} C_h}{\sum_{m=1}^{n} C_m + \sum_{h=1}^{n} C_h} \right], \tag{12}
$$

where CH_r represents the cache hit ratio, C_h indicates cache hit, and C_m shows cache miss ratio, whereas, n and m count the numbers of cache hit and cache miss, respectively. Figure 4 shows the results of the cache hit ratio, in which the study DPC claims that, initially, the cache hit ratio of DPC is lower than those PCMP, AlwaysCache, and Prob(0.5), but with the increasing time, the cache hit performance is improved. We analyze that our proposed work achieved performance cache hit an average of 68% from Prob(0.5), 65.5% from DPC, 55.8% from AlwaysCache, and 21.6% from PCMP. We obtained this result because our cache strategy

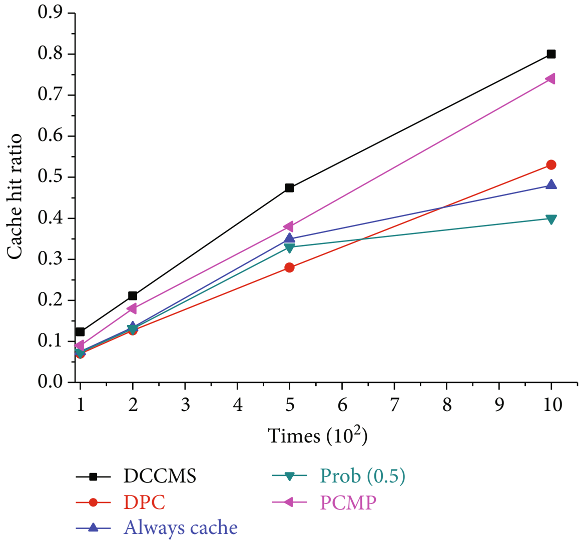


Figure 4: Cache hit (No. of 50 vehicles and times/sec).

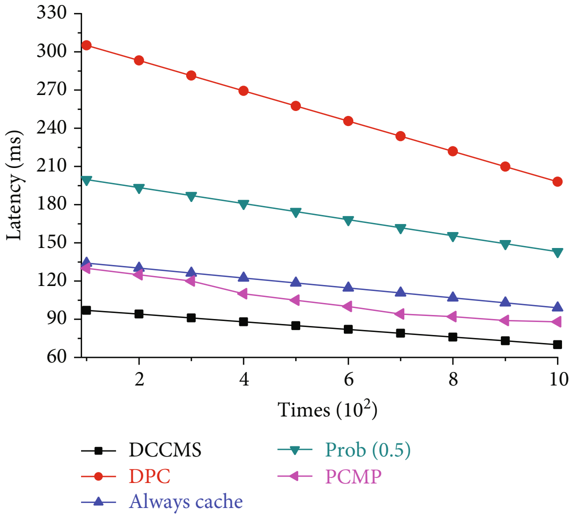


Figure 5: Latency (No. of 50 vehicles and times/sec).

collaborates with other nodes, and all requested contents are cached at nearby nodes in the network.

Latency is the time delay to obtaining the successful response, which can be measured as

$$L_t = C_{\mathrm{rt}} - \mathrm{Pr}_t, \tag{13}$$

where L_t indicates the latency, C_{rt} is the consumer request time, and Pr_t represent the producer response time for time for specific content. As per Figure 5, the DCCMS achieves the performance regarding the shortest latency as compared to mentioned other schemes. DPC shows higher latency than AlwaysCache, Prob(0.5), and PCMP. We minimized the latency rate overall on average 42% from other state-of-the-art schemes, with 67% from DPC, 51% Prob(0.5), 28% from AlwaysCache, and 21% from PCMP.

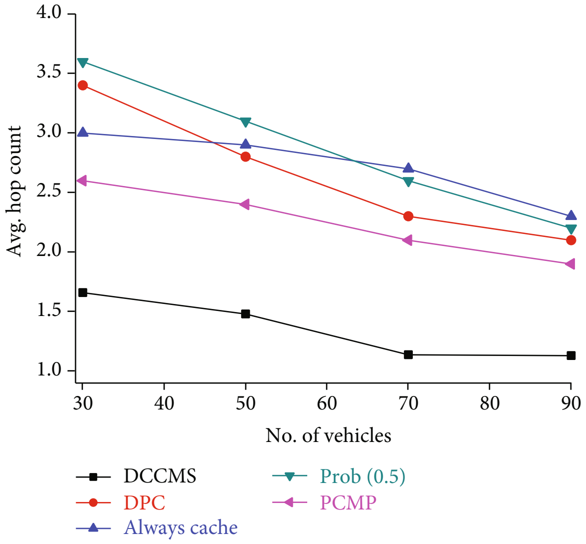


Figure 6: Average hop count.

Average hop is the distance between node$n_1(x_{n1} - y_{n1})$ and another node$n_2(x_{n2} - y_{n2})$, which is calculated by

$$H_c = \frac{\sum_{n_1 \neq n_2} \sqrt{(x_{n1} - x_{n2})^2 + (y_{n1} - y_{n2})^2}}{\sum_{hn1}}. \tag{14}$$

In Figure 6, the results show that the proposed scheme DCCMS outperforms other strategies. We decreased the hop count with the master node that cooperated with other nodes using most popular content. However, our proposed method decreased the average hop count by 53%, 50%, 49%, and 40% from Prob(0.5), AlwaysCache, DPC, and PCMP, respectively.

Content cache utilization is obtained by our proposed work, which indicates how often the content is used after being cached. We can obtain it as

$$C_{\mathrm{ut}} = \left[\frac{\sum_{h=1}^{n} C_h}{\sum_{j=1}^{n} \mathrm{tt}} \right], \tag{15}$$

where C_h indicates the cache hit probability and tt represents the caching event's total time. Cache utilization is an important factor in getting better cache hit results and improving network performance. Figure 7 shows that our scheme is slightly low compared to DPC with 30 vehicles but better than AlwaysCache, PCMP, and Prob(0.5). Moreover, with the increase in network size, such as 50, 70, and 90 vehicles, our proposed work utilized the maximum cache averagely of 55% overall, compared to other strategies DPC, AlwaysCache, Prob(0.5), and PCMP.

In diversity, the unique content ratio is stored on all nodes in the entire network. We can obtain it by the average number of networks' nodes nc that failed to cache the content each time it is divided by the total nodes tn of the whole network.

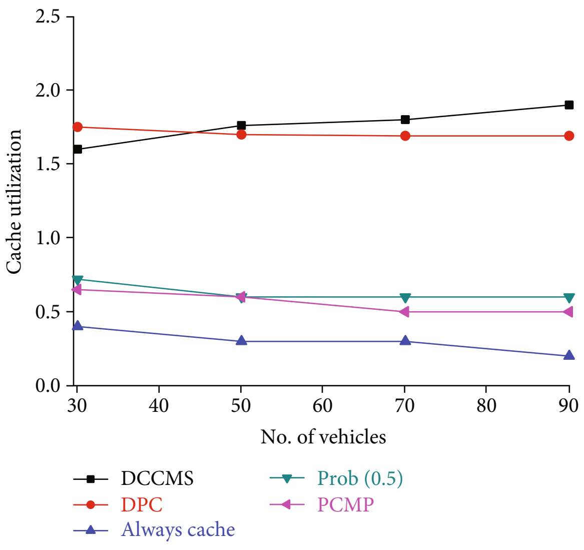


Figure 7: Cache content utilization ratio.

$$D=\left[\frac{\sum_{nc=1}^{n}\mathrm{nc}}{\sum_{tn=1}^{n}\mathrm{tn}}\right]. \quad (16)$$

Figure 8 illustrates that Prob(0.5) diversity is very low compared to other schemes. This happened due to higher content replication. PCMP and DPC perform much better than AlwaysCache and Prob(0.5). However, the DCCMC improves overall diversity performance on average 39% rather than baseline strategies. We achieved the diversity ratio with minimum content replicas. In addition, the other strategies generate too much content redundancy that causes the low diversity in the network.

Server load ratio indicates that the content is satisfied at server node rather than other nodes in the whole network. The server load can be calculated as

$$\mathrm{CH}_r=1-\left[\frac{\sum_{h=1}^{n}C_h}{\sum_{m=1}^{n}C_m+\sum_{h=1}^{n}C_h}\right]. \quad (17)$$

Figure 9 indicates that our scheme is better in terms of cache missing and decreases the server load by 30%, 28%, 27%, and 18% from Prob(0.5), DPC, AlwaysCahe, and PCMP, respectively, when time is increased. We achieved the performance due to catching the consumer's content at nearby nodes.

5.3.1. Effect of Velocity. We analyze the effect of velocity on the network performance with different vehicle velocities, using 50 vehicles, and set the 100 sec simulation time. Figures 9–11 indicate the cache hit ratio, latency, and average hop count performance, respectively, with different velocities from 5 m/s to 20 m/s. Furthermore, we observe that the overall performance of our DCCMC scheme is best than those of other methods due to the master node, which mitigates the impact of velocity on the network and cooperates with the neighboring node to cache the content.

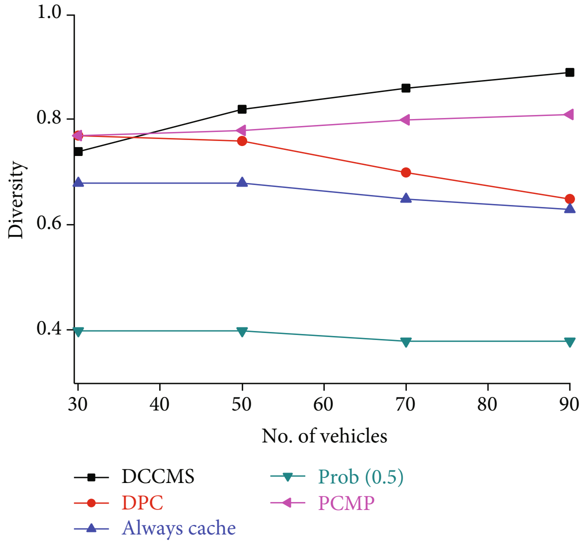


Figure 8: Diversity ratio (store unique contents).

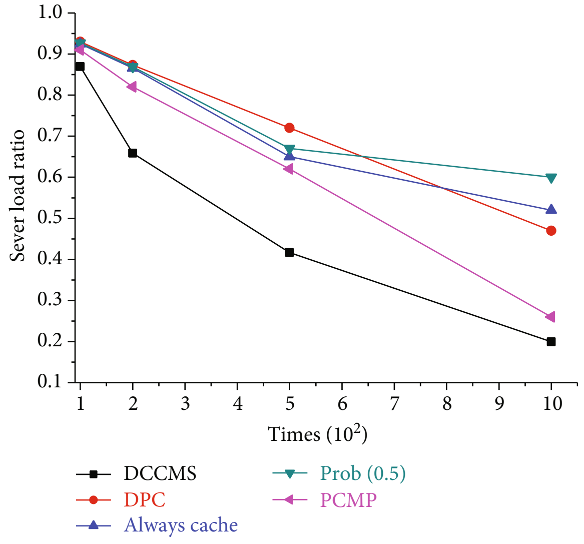


Figure 9: Server load (using 50 vehicles and simulation time: 100 sec).

In Figure 10, the results show that the cache hit ratio of all strategies decreases when vehicle speed increases. However, we observed that the performance of DCCMC is better and less affected with the increase of velocity. We achieved performance of CH_r 47% from DPC, 24% from Prob(0.5), 23% from AlwaysCache, and 21% from PCMP.

Figure 11 shows that the performance of DCCMC is best than those of other strategies. We decrease latency by 69%, 42%, 63%, and 10% from DPC, Prob(0.5), AlwaysCache, and PCMP, respectively.

Compared with the other methods in terms of average hop count, we notice that the results shown in Figure 12 illustrate that our DCCMC scheme reduced the hop count on average, 45% from Prob(0.5), 44% from AlwaysCache, 43% from DPC, and 35% from PCMP.

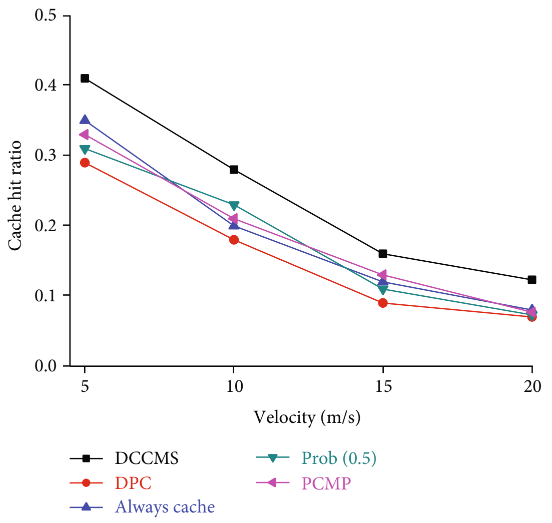


Figure 10: Cache hit ratio (simulation time: 100 sec).

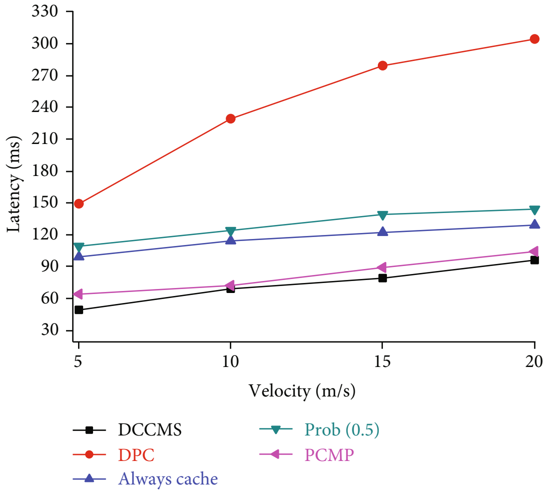


Figure 11: Latency (simulation time: 100 sec).

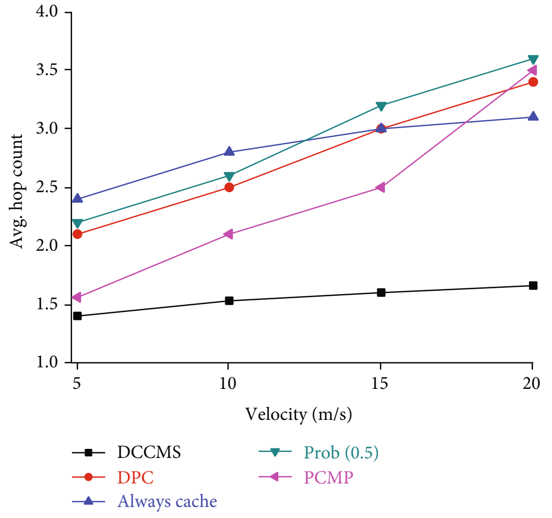


Figure 12: Average hop count (simulation time 100-sec).

6. Conclusion

VNDN plays a key role in content delivery in vehicular networks. This study proposed a novel two-level cache strategy to overcome the popular content delivery in high mobility situations. We introduced the master node that cooperates with other networks node based on content popularity in the dynamic environment. The caching node is chosen based on the centrality and social interaction with other nodes. The simulation results show that our DCCMS scheme is efficient rather than other state-of-the-art strategies. DCCMS is better when the higher cache hit, less hop, less redundancy, more cache utilization, and less latency are the genuine concern in the networks. In the future, we plan to investigate forwarding schemes and the replacement content in both static and dynamic environments to improve the network performance based on the most frequent nodes in the network. Moreover, we will design caching strategy for the Internet of things (IoT) to optimize cache and improve our scheme according to the problem statements.

Data Availability

The simulation data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research study is funded by the National Natural Science Foundation of China under Grant 61772385. The authors acknowledge this financial support.

References

[1] J. Ren, Y. Chen, L. Xin, J. Shi, and H. Mahama, “Detecting and locating of traffic incidents in a road segment based on lane-changing characteristics,” *Transportmetrica A: transport Science*, vol. 13, no. 10, pp. 853–873, 2017.

[2] Z. Wang, P. Dong, T. Zheng, and H. Zhang, “DDGS: a network coding scheme for dynamic adaptation to heterogeneous vehicular networks,” *Wireless Communications and Mobile Computing*, vol. 2021, 12 pages, 2021.

[3] T. Barnett, S. Jain, U. Andra, T. Khurana, and Cisco visual networking index, “Global mobile data traffic forecast updata,” 2017, https://www.cisco.com/.

[4] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, “A survey of green information-centric networking: Research issues and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1455–1472, 2015.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” *Communications of the ACM*, vol. 55, no. 1, pp. 117–124, 2012.

[6] M. A. Hail, “IoT-NDN: an IoT architecture via named data netwoking NDN,” in *International Conference on Industry 4.0 Artificial Intelligence and Communications Technology*, pp. 74–80, Bali, Indonesia, August 2019.

[7] C. Campolo, A. Molinaro, and R. Scopigno, “From today's VANETs to tomorrow's planning and the bets for the day after,” *Vehicular Communications*, vol. 2, no. 3, pp. 158–171, 2015.

[8] L. Wan, L. Sun, K. Liu, X. Wang, Q. Lin, and T. Zhu, “Autonomous vehicle source enumeration exploiting non-cooperative UAV in software defined internet of vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3603–3615, 2021.

[9] J. Chen, H. Wu, P. Yang, F. Lyu, and X. Shen, “Cooperative edge caching with location-based and popular contents for vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10291–10305, 2020.

[10] L. Yao, Y. Wang, X. Wang, and W. U. Guowei, “Cooperative caching in vehicular content centric network based on social attributes and mobility,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 391–402, 2021.

[11] H. Khelifi, S. Luo, B. Nour et al., “Named data networking in vehicular ad hoc networks: state-of-the-art and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 320–351, 2020.

[12] S. Saha, A. Lukyanenko, and A. Ylä-Jääski, “Cooperative caching through routing control in information-centric networks,” *Proceedings IEEE INFOCOM*, vol. 2013, pp. 100–104, 2013.

[13] J. M. Wang, J. Zhang, and B. Bensaou, *Intra-AS cooperative caching for content-centric networks*, Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, 2013.

[14] Y. Liu, D. Zhu, and W. Ma, “A novel cooperative caching scheme for content centric mobile ad hoc networks,” *IEEE Symposium on Computers and Communication ISCC*, vol. 2016, pp. 824–829, 2016.

[15] J. Ji, M. Xu, and Y. Yang, “Content-hierarchical intra-domain cooperative caching for information-centric networks,” in *Proceedings of The Ninth International Conference on Future Internet Technologies*, pp. 1–6, Tokyo Japan, June 2014.

[16] Z. Su, Y. Hui, and S. Guo, “D2D-based content delivery with parked vehicles in vehicular social networks,” *IEEE Wireless Communications*, vol. 23, no. 4, pp. 90–95, 2016.

[17] S. Arif, S. Hassan, and I. Abdullahi, “Cache replacement positions in information-centric network,” in *4th International Conference on Internet Applications, Protocols and Services*, pp. 54–58, Putrajaya, Malaysia., December 2015.

[18] N. Laoutaris, H. Che, and I. Stavrakakis, “The LCD interconnection of LRU caches and its analysis,” *Performance Evaluation*, vol. 63, no. 7, pp. 609–634, 2006.

[19] D. Rossi and G. Rossini, “On sizing CCN content stores by exploiting topological information,” *Proceedings IEEE INFOCOM Workshops*, vol. 2012, pp. 280–285, 2012.

[20] S. Zhang, W. Sun, and J. Liu, “Spatially cooperative caching and optimization for heterogeneous network,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11260–11270, 2019.

[21] J. A. Khan, C. Westphal, and Y. Ghamri-Doudane, “Information-centric fog network for incentivized collaborative caching in the Internet of everything,” *IEEE Communications Magazine*, vol. 57, no. 7, pp. 27–33, 2019.

[22] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, “Collaborative learning of communication routes in edge-enabled multi-access vehicular environment,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1155–1165, 2020.

[23] C. Wu, Z. Liu, T. Yoshinaga, Y. Ji, and D. Zhang, “Spatial intelligence toward trustworthy vehicular IoT,” *IEEE Communications Magazine*, vol. 56, no. 10, pp. 22–27, 2018.

[24] X. Li, X. Wang, K. Li, Z. Han, and V. C. M. Leung, “Collaborative multi-tier caching in heterogeneous networks: modeling analysis and design,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6926–6939, 2017.

[25] C. Yang, Y. Yao, Z. Chen, and B. Xia, “Analysis on cache-enabled wireless heterogeneous networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 131–145, 2016.

[26] 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, 2012.

[27] H. Xing and W. Song, “Collaborative content distribution in 5G mobile networks with edge caching,” in *IEEE International Conference on Communications*, Shanghai, China, May 2019.

[28] J. Song, P. Lin, Q. Song, and L. Guo, “Cost-oriented cooperative caching scheme in energy-harvesting-powered ultra-dense networks,” *IEEE/CIC International Conference on Communications in China (ICCC)*, vol. 2020, pp. 729–734, 2020.

[29] P. Lin, Q. Song, Y. Yu, and A. Jamalipour, “Extensive cooperative caching in D2D integrated cellular networks,” *IEEE Communications Letters*, vol. 21, no. 9, pp. 2101–2104, 2017.

[30] F. Qazi, O. Khalid, R. N. B. Rais, I. A. Khan, and A. . R. Khan, “Optimal content caching in content-centric networks,” *Wireless Communications and Mobile Computing*, vol. 2019, 15 pages, 2019.

[31] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, “WAVE: popularity-based and collaborative in-network caching for content-oriented networks,” *Proc. IEEE INFOCOM Workshops*, pp. 316–321, 2012.

[32] Z. Ming, M. Xu, and D. Wang, “Age-based cooperative caching in information-centric networks,” *Proc. IEEE INFOCOM Workshops*, pp. 268–273, 2012.

[33] 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.

[34] S. Lim, W. Lee, G. Cao, and C. R. Das, “Performance comparison of cache invalidation strategies for Internet-based mobile ad hoc networks,” in *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 104–113, Fort Lauderdale FL US, October 2004.

[35] Y. Zhang, L. Xie, and G. Cao, “Data consistency for cooperative caching in mobile environments,” *Computer*, vol. 40, no. 4, pp. 60–66, 2007.

[36] D. Barbara and T. Imielinski, “Sleepers and workaholics: caching strategies for mobile environments,” in *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 1–12, ACM Press, USA, 1994.

[37] G. Cao, “A scalable low-latency cache invalidation strategy for mobile environments,” *IEEE Trans. Knowledge and Data Eng*, vol. 15, no. 5, pp. 1251–1265, 2003.

[38] S. Soleimani and X. Tao, “Caching and placement for in-network caching in device-to-device communications,” *Wireless Communications and Mobile Computing*, vol. 2018, 9 pages, 2018.

[39] C. Bernardini, T. Silverston, and O. Festor, “A comparison of caching strategies for content centric networking,” *IEEE Global Communications Conference GLOBECOM*, vol. 2015, pp. 1–6, 2015.

[40] F. J. Gonzalez-Canete and E. Casilari, “A Trivi no-Cabrera A cross layer interception and redirection cooperative caching scheme for MANETs,” *Journal on Wireless Communications and Networking*, vol. 2012, no. 1, 2012.

[41] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: the role of proactive caching in 5G wireless networks,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[42] E. Baştuğ, M. Bennis, and M. Debbah, “A transfer learning approach for cache-enabled wireless networks,” *13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pp. 161–166, 2015.

[43] D. Rossi and G. Rossini, *Caching performance of content centric networks under multi-path routing and more*, Telecom ParisTech, Relatório técnico, 2011.

[44] W. Huang, T. Song, Y. Yang, and Y. Zhang, “Cluster-based cooperative caching with mobility prediction in vehicular named data networking,” *IEEE Access*, vol. 7, pp. 23442–23458, 2019.

[45] M. A. M. Hail, M. Amadeo, A. Molinaro, and S. Fischer, *On the performance of caching and forwarding in information-centric networking for the IoT*, In International conference on wired/wireless internet communication, Cham, 2015.

[46] K. A. Raza, A. Asheralieva, M. M. Karim, K. Sharif, M. Gheisari, and S. Khan, “A novel forwarding and caching scheme for information-centric software-defined Nnetworks,” *International Symposium on Networks, Computers and Communications*, vol. 2021, pp. 1–8, 2021.

[47] S. Arianfar, P. Nikander, and J. Ott, “On content-centric router design and implications,” *Proceedings of the Re-Architecting the Internet Workshop*, pp. 1–6, 2010.

[48] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji, *Performance of probabilistic caching and cache replacement policies for content-centric networks*, 39th Annual IEEE Conference on Local Computer Networks, 2014.

[49] G. Deng, L. Wang, F. Li, and R. Li, “Distributed probabilistic caching strategy in VANETs through named data networking,” *IEEE INFOCOM Workshops*, pp. 314–319, 2016.

[50] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, and F. Naït-Abdesselam, “An optimized proactive caching scheme based on mobility prediction for vehicular networks,” *IEEE Global Communications Conference GLOBECOM*, vol. 2018, pp. 1–6, 2018.

[51] P. Shunmugapriya, S. Kanmani, P. J. Fredieric, U. Vignesh, J. R. Justin, and K. Vivek, “Effects of introducing similarity measures into artificial bee colony approach for optimization of vehicle routing problem,” *Int. J. Comput. Inf. Eng*, vol. 10, no. 3, pp. 651–658, 2016.

[52] A. Afanasyev, I. Moiseenko, and L. Zhang, *ndnSIM: NDN simulator for NS-3*, NDN, Technical Report NDN-0005, 2012, http://named-data.net/techreports.html.

[53] G. F. Riley and T. R. Henderson, *The Ns-3 Network Simulator*, Berlin, Heidelberg, Modeling and tools for network simulation. Springer, 2010.