

Research Article

Node Value and Content Popularity-Based Caching Strategy for Massive VANETs

Jinyuan Gu ^{1,2}, Yancheng Ji,¹ Wei Duan ¹ and Guoan Zhang ¹

¹School of Information Science and Technology, Nantong University, Nantong 226000, China

²Kangda College of Nanjing Medical University, Lianyungang, China

Correspondence should be addressed to Guoan Zhang; gzhang@ntu.edu.cn

Received 28 January 2021; Revised 5 March 2021; Accepted 24 March 2021; Published 5 April 2021

Academic Editor: Zhaolong Ning

Copyright © 2021 Jinyuan Gu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The high-speed dynamic environment and massive information transmitted via wireless communications in the vehicular ad hoc networks (VANETs) pose a great challenge to privacy and security. To overcome this issue, use of the content-centric networking (CCN) provides a potential and practical solution. In-network caching is a main feature for future smart cities, in which the content is mainly placed in network nodes. Therefore, how to effectively select the cache locality and cache content is essential to improve the overall network performance, which is an inevitable trend. With these observations, this article proposes a caching strategy based on the node value and content popularity (NVCP) for the massive VANET scenario. In the proposed NVCP scheme, different from the traditional caching strategies, we evaluate the node value from three aspects: the connectivity, intermediary, and eigenvector centralities, synthetically, since the content with different types of popularity is placed in nodes with different values, resulting in the redundancy deterioration and diversity improvement for the content. The proposed caching strategy is evaluated by the stochastic network topology with multifactors, which provides different impacts on the system performance. Simulation results show that the NVCP outperforms the traditional cache strategies for 6G-CCN in terms of the cache hit ratio, average hop count, and transmission latency. Moreover, placing the content in the neighbor nodes is also introduced to further improve the utilization of the cache space and achieve better cache performance.

1. Introduction

The sixth generation (6G) wireless communication will build an integrated air-space, sea/ground network with distributed components [1]. In 6G, the “ubiquitous connectivity” will become one of the main features with the deployment in the desert, deep ocean, high mountain, and other extreme environments to achieve seamless coverage. As one of the infrastructures for the future smart city, 6G will adopt a unified network architecture, introduce new business scenarios, and build a more efficient and complete network, i.e., content-centric networking (CCN) and software-defined network (SDN). In the future, the 6G network can be jointly invested by many operators, and the physical network and logical network can be separated by network virtualization, SDN, and network slicing technologies [2].

Moreover, with the rapid development of Internet of Vehicles (IoV) for the future smart city, emerging exten-

sive networking applications and the content-oriented and personalized information services have become the mainstream trend of the networking development [3, 4]. Since the existing transmission control protocol/internet protocol (TCP/IP) architecture is based on the endpoint addressing, the frequent session connections cannot effectively guarantee the reliable exchange of the information, which provides urgent problems such as the waste of network resource, data redundancy, and transmission congestion. The current typical solution to solve the above problems is to build an overlay network at the application layer, i.e., the content distribution network (CDN) [5, 6] and peer-to-peer (P2P) schemes. Even though these solutions can alleviate the problem of content distribution and sharing to a certain extent, they are hard to adapt to the various network layer functions and meet the high-speed transmission requirements.

Moreover, since vehicular ad hoc networks (VANETs) provide short-distance, low latency connections, it is hard

for the TCP/IP networking approach to satisfy these requirements, and its drawbacks can be overcome by adopting CCN, especially in 6G networks [7]. The information-centric network is centered on the content, breaking the traditional “host-to-host” communication mode. Specifically, the “end-to-end” communication driven by the information provider is transformed into content retrieval driven by the receiver. As a typical representative of the new information-centric networking, CCN carries the content as the focus and basic unit of transmission and replaces the IP address as the waist of the hourglass structure, which receives great attention and becomes a research hot spot for the next generation internet architecture [8]. Different from the Web, CDN, and P2P, in-network caching is the main feature of CCN. The current research [9], i.e., RFC 8569, on CCN mainly includes the content of the allocation and sharing, placement strategy, replacement strategy, and utilization, etc., in which the content placement is used to determine whether a node places the content; also, the content replacement strategy is used to replace the old content with new content when the cache space of a node is saturated. In CCN, when a request sent by a consumer finds the corresponding content at the content router, the content will be directly sent to the consumer without forwarding the request to the source server. In this manner, the transmission latency and pressure will be significantly reduced with an improvement of the consumer’s experience. Therefore, the formulation of the caching strategy has a significant impact on the performance of CCN, especially for the selection of cache locality and cache content. The reasonable selections of the cache locality and content can enable that the consumers to obtain the content from the content router more effectively.

Leave Copy Everywhere (LCE) is the default cache policy of CCN, which requires the content router on the delivery path between the consumer and the server to cache each passing content. That results in a large amount of content redundancy and less content diversity in the network. Prob(p) [10] is the probability that each content router cache’s content is p ; the noncaching probability is $1 - p$. When a content router receives a data packet, it randomly generates a number from 0 to 1. If the number is less than or equal to p , the content is cached; otherwise, it is directly forwarded to the next hop. ProbCache [11] caches the content in the content router according to probability, where the probability of each content router is different, which is inversely proportional to the distance from the consumer. Therefore, the closer to the consumer, the greater the probability that the content will be cached. The leave copy down (LCD) [12] caches content only in the next-hop content router when caching hits. However, the content needs to go through multiple requests to reach the edge of the network and will produce a large amount of content redundancy. In addition, when the cache hits, move copy down (MCD) [13] will move the cached content from the hitting nodes to the next-hop content router (except the server), which reduces the cache redundancy. On the other hand, when the request comes from consumers on different paths, there will be a swing of the content cache location, and this dynamicity will generate

more network overhead. Although MCD and LCD work in a similar manner, MCD will delete the cached content at the cache hitting nodes (except the server), reducing the content redundancy, but at the same time, the dynamic of the cache nodes will increase the network overhead. The authors in [14] proposed a centrality-based caching strategy which utilizes the betweenness of node centrality to improve the cache hit ratio. Despite that, only the caching content at the centrality node leads to idling of other content routers, and the content will be replaced frequently. In [15], the caching problem from content popularity is considered, where the popularity of content and cache content are classified with high popularity. Therefore, only the content with high popularity exists in the network; the remaining will be ignored, resulting in low content diversity.

The advantages of 6G-CCN can be summarized as follows: (1) The data can be obtained from any cache, not from the fixed channel; therefore, there is no data channel security in a CNN network. (2) Compared with TCP/IP network, CCN has higher flexibility, security, and robustness without performance loss. (3) Due to the ability of natural traffic regulation, when forwarding the data, CCN can choose the forwarding strategy according to the link condition to balance the whole network traffic. Therefore, one can conclude that CCN will simplify and empower network efficiency and improve the security, which is envisioned as a potential networking candidate for 6G-IoV communications. These motivate us to propose a caching strategy based on node value and content popularity (NVCP) for the massive VANET scenario. The implementations and contributions of this article are summarized in the following:

- (i) In NVCP, the value of a node is determined according to the connectivity, betweenness centrality, and eigenvector centrality. The importance of the content is determined by the popularity of the content; meanwhile, the choice of cache location and cache content is dependent on the value of the node and the popularity of the content. Nodes with different values cache content with different popularity, in which they are directly proportional
- (ii) On the one hand, the NVCP makes use of the differences between the popularity of different types of content to make sure the cached content is distributed evenly, which simultaneously reduces the content redundancy and increases the diversity of content. On the other hand, the value of nodes from multiple attributes is evaluated, and the differences between content routing locations are used, which can significantly improve the node utilization and cache hit ratios and reduce content acquisition hops and latency, as well as improve the user experience
- (iii) NVCP with LCE, Prob(b), and MPC (Most-Popular Content) is compared in three aspects: the cache hit ratio, average hop counts, and average transmission latency; the simulation results demonstrate that the

proposed NVCP is superior to the other cache strategies in all aspects.

The rest of this paper is organized as follows. Section 2 introduces the composition of the CCN and working mechanism. Section 3 focuses on the details of our proposed NVCP mechanism. We evaluate the performance of the proposed strategy and compare with the other strategies in Section 4. Finally, conclusions and future work are described in Section 5.

2. The Composition of CCN and Working Mechanism

Consider that the communication of CCN consists of two packets as shown in Figure 1: the interest packet including the content name, selector, and nonce to forward the requests of the consumers through the CCN nodes and the data packet composed of the content name, signature, and signed information and data, which is transmitted along the reverse path of interest packet to the satisfied consumers.

Clearly, the greatest feature for CCN is that it no longer uses the host and interface addresses for routing but uses the content name as the unique identifier for identification and transmission. Therefore, the content can be cached in the CCN to support various functions including content distribution and multicast. The node records the corresponding status and interface information in the interest packet request process, and the data packet is hopped back to the consumer according to the information. Since the content store (CS), forwarding information base (FIB), and pending interest table (PIT) are maintained inside the CCN nodes, each node uses the above three types of data structures for content distribution. For these three data structures, the CS is used to cache a copy of content passing through the node in order to satisfy subsequent content requests; the role of PIT is to record interest packets that have not been satisfied, including the content name and the corresponding arrival interface, which is aimed at aggregating the same content request avoiding sending the same interest packet repeatedly; for FIB, it saves the next hop interface information to the provider for interest packet routing. The working mechanism of the CCN can be briefly summarized as follows:

- (i) When an interest packet arrives at the content router, it will be first queried whether the CS has cached the content. If it is, the data packet returns to the consumer directly; otherwise, the PIT is queried
- (ii) If there is an entry request from the content, the corresponding arrival face is added to the PIT. Otherwise, according to the information of FIB, the maximum matching query is performed in the FIB. Then, the interest packet is forwarded to the next hop, and a new PIT table will be established
- (iii) On the other hand, when the data packet is sent back, the requested content entry is checked whether it exists in the PIT. If existing, the data packets are

forwarded to consumers according to the arrival interface information (one or more) and the entry in the PIT is deleted. The cache placement policy determines whether the CS caches the content

3. The Proposed NVCP Caching Strategy

In this section, the cache locality realization is firstly introduced to satisfy the requirements of the connectivity, betweenness centrality, and eigenvector centrality. In the following, after defining the cache content, the proposed NVCP caching strategy will be discussed within two algorithms.

3.1. Cache Locality. It is known that how to select the cache locality is still an open issue, since the cache locality has a significant impact on the performance of the CCN. In this subsection, three node attributes are defined to evaluate the value of the node, which are based on the graph theory and described. Moreover, we further considered that the named-data link state routing protocol (NLSR) is adopted to query the shortest path information. Given an undirected graph $G = (V, E)$ with n vertexes and m edges, $V = \{v_1, v_2, \dots, v_n\}$ represents a set of content routers and $E = \{e_1, e_2, \dots, e_m\}$ denotes the links between the content routers. Moreover, $A = (a_{ij})_{n \times n}$ is the adjacency matrix of G ; for v_i , directly connect with v_j and $a_{ij} = 1$; otherwise, $a_{ij} = 0$.

3.1.1. Connectivity. Different forwarding strategies result in different routing paths for the requested content; cache nodes will play different roles in these strategies. And hence, we regard the number of paths that the requested content passes through the cache node as the connectivity of the node. Therefore, with the increasing paths, the request content becomes more important. Defending σ_{st} as the number of shortest paths between v_s and v_t and $\sigma_{st}(v_i)$ as the number of shortest paths from v_s to v_t through v_i , the betweenness centrality of v_i can be presented as

$$C_s(v_i) = \frac{c_s(v_i)}{c_s(v_i)_{\max}}, \quad (1)$$

where $c_s(v_i)_{\max}$ represents the maximum number of routing paths passing through v_i .

3.1.2. Betweenness Centrality. If a content router is on the shortest paths between the corresponding content routers, the content router is considered to be in a significant position. It is reasonable, since the content router in this position can affect the overall network by controlling or misinterpreting the transmission of information. The ability to characterize the content router control information transfer is the betweenness centrality (also known as node median) [16].

$$C_B(v_i) = \left(\frac{(n-1)(n-2)}{2} \right)^{-1} \sum_{s \neq t \neq i \in V} \frac{\sigma_{st}(v_i)}{\sigma_{st}}, \quad (2)$$

where $C_B(v_i)$ describes the betweenness centrality of v_i and σ_{st} is the number of shortest paths between v_s and v_t , while

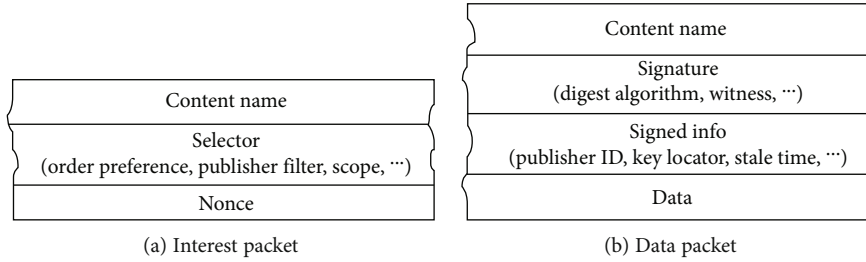


FIGURE 1: The composition of interest packet and data packet in CCN.

$\sigma_{st}(v_i)$ represents the number of shortest paths from v_s to v_t through v_i . Normalize $C_B(v_i)$ by dividing $n(n-1)/2$.

3.1.3. Eigenvector Centrality. In fact, the influence of a content router is related not only to its own locality but also to the influence of its neighbors [17]. If the content router is chosen by a very popular actor, the corresponding influence will also be increased. On the other hand, there is an influence on an influential node; it is clear that the influence will be even greater, where the eigenvector centrality is used to characterize the influence. We define $C_E(v_i)$ as the eigenvector centrality of a node, indicating the influence of the neighbors of nodes. It is also defended that $C_E(v_i)$ not only reflects the relative centrality of the network but also reflects the long-term influence of the node.

$$C_E(v_i) = \frac{1}{\lambda} a_{ij} e_j, \quad i = 1, 2, \dots, N. \quad (3)$$

According to the existing research [18], the network is distributed in a power law, and the node in different positions plays different roles. The connectivity and betweenness centrality consider the value of nodes from routing paths of the requested contents; meanwhile, the eigenvector centrality takes the influence of neighbors into account. When selecting the cache locality, the NVCP considers the above three attributes simultaneously. The comprehensive attribute $M(v_i)$ is expressed as

$$M(v_i) = \alpha C_S(v_i) + \beta C_B(v_i) + \gamma C_E(v_i), \quad (4)$$

with the condition

$$\alpha + \beta + \gamma = 1, \quad (5)$$

where α , β , and γ , respectively, denote the weight of connectivity, betweenness centrality, and eigenvector centrality, and the sum of them is 1 ($\alpha \geq 0, \beta \geq 0, \gamma \geq 0$). It is worth noting that, in our proposed scheme, the three mentioned attributes have different influences on choosing the cache locality. Based on which, when different attributes are used to evaluate the importance of nodes in the same network, the corresponding different results will be obtained. Therefore, the coefficients in the comprehensive attribute $M(v_i)$ are determined by the related requirements of CCN.

3.2. Cache Content. Since whether caching every content which passes through the content router is another problem for the CCN, the popularity is a factor to draw the content. The popularity of the content can be estimated by the content request count during a measurement, which means that the more content request counts, the greater the popularity and probability that the content will be requested. The popularity of content k is given by

$$P_{v_i}(k) = \frac{f_{v_i,k}}{f_{v_i}^{\max}}, \quad (6)$$

where $f_{v_i,k}$ represents the count requesting for the content k at v_i , and $f_{v_i}^{\max}$ denotes the max count of v_i . Thus, the value of $P_{v_i}(k)$ should be smaller than 1. Specifically, as shown in [13], $P_{v_i}(k)$ needs to be over some time window to have significance, not over all history. Since this article is aimed at providing a new mentality to further improve the utilization of the cache space and achieve better cache performance in CCN, in our future work, we will take the time window into account.

3.3. The NVCP Cache Strategy. For the proposed NVCP, the core idea is based on the node value and content popularity; a table is considered to be added at each content node including the content name, the number of the routing path, and the count of content requests to store the information of the content and cache node. It is remarkable that, in CCN/NDN, PIT records the requests that have not been satisfied, including the content name and corresponding arrival interface, to ensure the returned response packet to the content requester along the reverse path. Therefore, the source of a request is identified through PIT. By this way, when a consumer requests content, the betweenness centrality and eigenvector centrality of the nodes on the delivery path will be calculated and normalized. Once the request is satisfied, the data packet is returned on the inverse delivery path. At this time, the content popularity will be calculated according to the count of content request. In our proposed scheme, we design a variable φ to match the content popularity and node value given as

$$\varphi = \frac{P_{v_i,k}}{M(v_i)}, \quad (7)$$

```

G: the network topology
Initialize  $c_S(v_i), C_B(v_i), C_E(v_i), f_{v_i,k}$ 
for node on the delivery path from consumer to server do
  if content in cache
    then send content back to the consumer
        discard interest packet
  else
    get the adjacency matrix of the nodes according G
     $\sigma_{st}$ : record the number of shortest paths between  $v_s$  and  $v_t$ 
     $\sigma_{st}(v_i)$ : record number of shortest paths from  $v_s$  to  $v_t$  through  $v_i$ 
    calculate  $C_B(v_i), C_E(v_i)$ 
     $c_S(v_i) \leftarrow c_S(v_i) + 1$ 
     $f_{v_i,k} \leftarrow f_{v_i,k} + 1$ 
    forward the interest packet to the next hop towards server
  end if
end for

```

ALGORITHM 1: The process to get betweenness centrality and eigenvector centrality. Set forward path.

```

G: the network topology
Input  $c_S(v_i), C_B(v_i), C_E(v_i), f_{v_i,k}$ 
for node on the delivery path from server to consumer do
  if the content is provided by server
    then send the data packet back directly
  else
    calculate  $C_S(v_i), P_{v_i,k}$ 
    get  $C_B(v_i), C_E(v_i)$ 
     $M_{v_i} \leftarrow \alpha C_S(v_i) + \beta C_B(v_i) + \gamma C_E(v_i)$ 
  end if
  if  $\varphi = P_{v_i,k}/M(v_i) \geq 1$ 
    then cache the contents
  else
    forward the data packet to the next hop to the consumer
  end if
end for

```

ALGORITHM 2: The process to select the appropriate cache locality and cache content. Select cache locality and cache content.

where $P_{v_i}(k)$ is the popularity of content k at v_i , and from equations (4) and (6), we get that the values of $P_{v_i,k}$ and $M(v_i)$ are fixed and less than 1. In general, there are two cases: (1) $P_{v_i,k} \geq M(v_i)$; it means that the popularity of content is more important than the value of node. Therefore, caching the content in the content router can obtain a higher cache hit rate. (2) $P_{v_i,k} < M(v_i)$; it means that the value of the node is high but the corresponding popularity of the content is low. If caching the content with a lower popularity will result in a waste of the cache space, considering these two cases, in equation (7), φ is set to ≥ 1 .

The main idea of the proposed NVCP is presented in Algorithms 1 and 2. In our proposed scheme, considering that the location of the content router does not change, we have a fixed network topology. Therefore, the network can be seen as an undirected graph; the corresponding algorithms (such as the Brande algorithm and Power Iteration) will be

used to obtain $C_B(v_i)$ and $C_E(v_i)$ in advance, resulting in a computational complexity as $\mathcal{O}(VE)$ for these two algorithms. Algorithm 1 is the process to obtain the betweenness centrality and eigenvector centrality. It is clear that, when the interest packet arrives at a content router, if the CS has the content, it sends the content back to the consumer; otherwise, it calculates $C_B(v_i)$ and $C_E(v_i)$ according to the network topology. Meanwhile, the values of $C_S(v_i)$ and $f_{v_i,k}$ increase by 1. On the other hand, Algorithm 2 illustrates the process to select the appropriate cache locality and cache content. According to the results given in Algorithm 1, calculate φ . If $\varphi > 1$, the content is cached; otherwise, the data packet to the next hop is forwarded. In addition, considering the fixed locations of content routers, the values of $C_B(V_i)$ and $C_E(V_i)$ only need to be calculated once. In this way, when requested, the popularity of the content increases by 1, which is easy to realize. Clearly, compared with the existing works, our

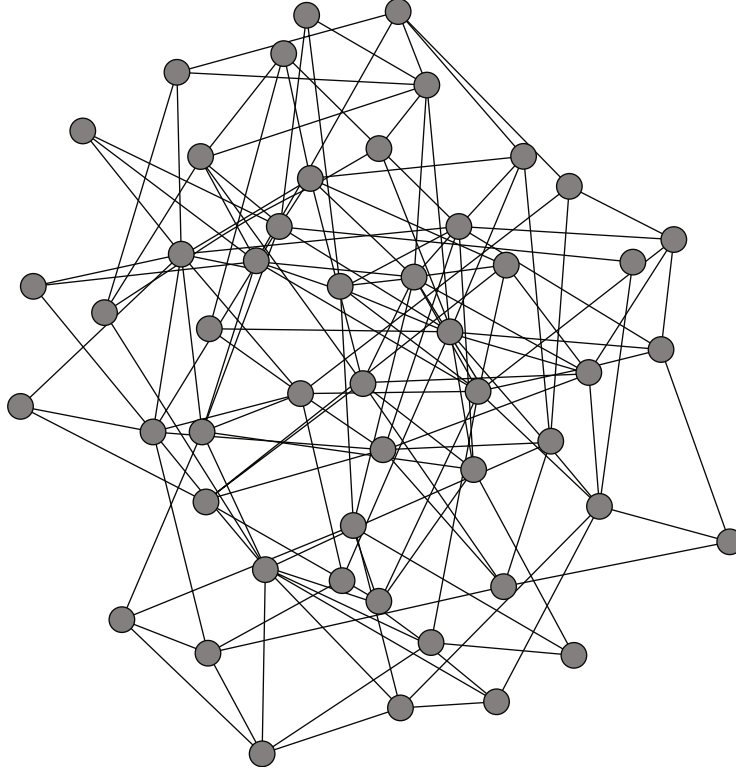


FIGURE 2: The network topology for our proposed scheme.

proposed algorithm significantly improves the efficiency for calculating the value of φ . Clearly, the computational complexities of Algorithms 1 and 2 are not extremely high, which are practical and acceptable.

4. Numerical Results

The proposed cache strategy is evaluated by ndnSIM simulator. The ndnSIM simulator is an NDN simulation module based on NS-3 which implements the basic functions of NDN. By modifying the code, the cache strategy proposed in this article is implemented, and the results are imported into MATLAB to provide the performance comparison between the existing cache strategies.

4.1. Simulation Settings. The simulation uses a network topology generated randomly as shown in Figure 2, which consists of 50 nodes and 150 links. There is a source server in the network, which is connected to a node randomly, and the edge nodes are connected to the consumers. Content requests are generated following the Zipf-Mandelbrot distribution with $a=0.7$. The total number of different types of content will be requested in the network as 10,000. Furthermore, we assume that the interests of each consumer are generated following the Poisson distribution with $\lambda=100/s$. With regard to the comprehensive consideration of the various attributes of the node, for simplicity and fairness, in this article, the specific weight values of α (connectivity), β (betweenness central-

TABLE 1: Simulation parameters.

Parameter	Default value	Variation range
Nodes	50	—
Links	150	—
Delay (ms)	10	—
Bandwidth (mbps)	10	—
Content	10000	—
Consumers	18	—
Cache size	1000	100~2000
Zipf (a)	0.7	0.1~1.0
Simulation time (s)	100	—

ity), and γ (eigenvector centrality) in the presented simulation results are equivalently given as 1/3. The Least Recently Used (LRU) is employed as the cache replacement strategy, and the total simulation time is 100 s. More specifically, the simulation results have been evaluated for various values of the cache size. The main simulation parameters are listed in Table 1.

4.2. Performance Index of Simulations. The proposed NVCP strategy is compared with the LCE, Prob(0.5), and MPC in terms of the cache hit ratio, average hop count, and average transmission latency, which are described in details as follows [19]:

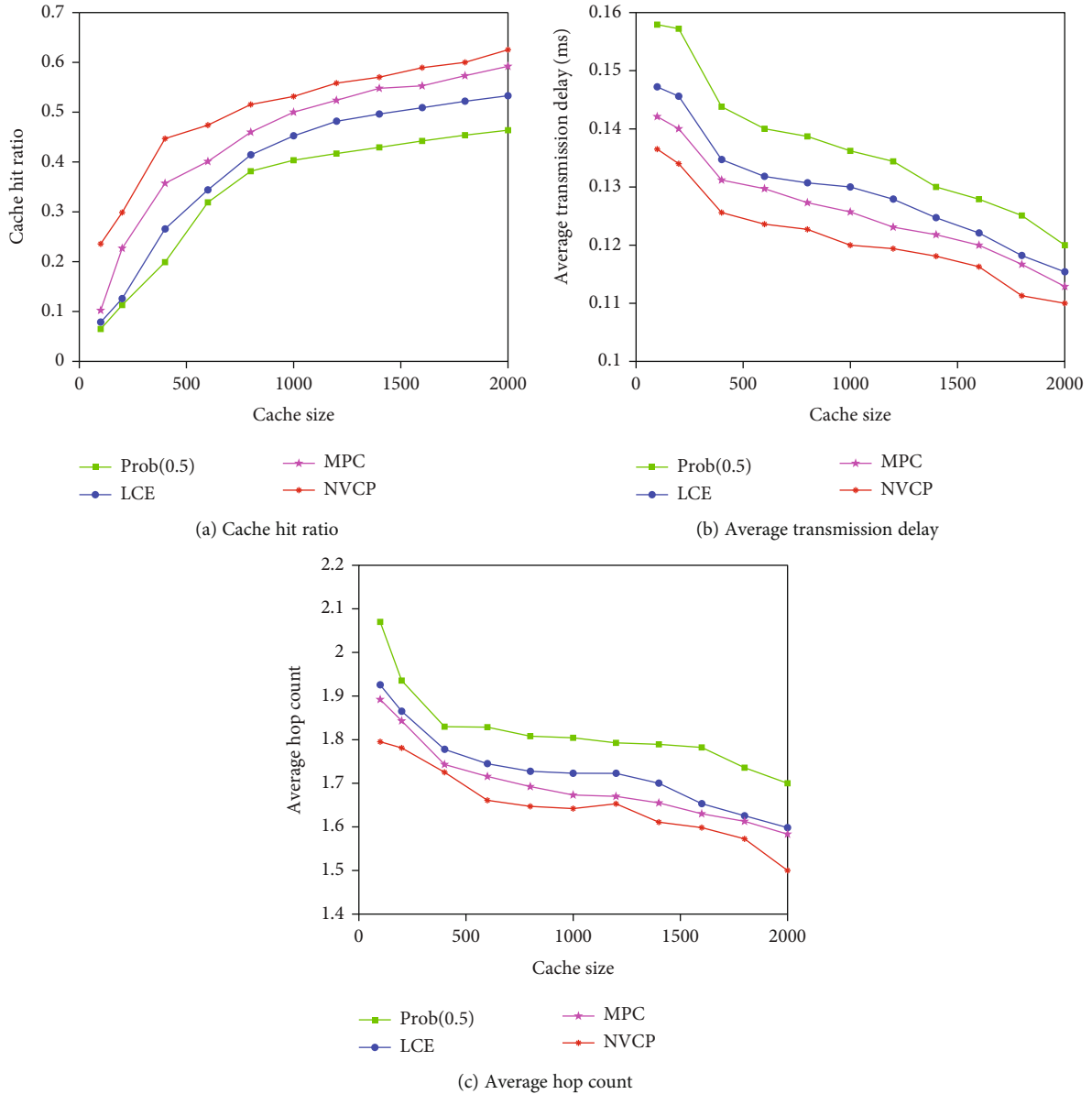


FIGURE 3: The impact of cache size on the system performance.

(i) *Cache hit ratio*: refers to the probability that the consumer request is satisfied by the cache node instead of the server. It is a typical parameter, which reflects the performance of the cache strategy. The higher the cache hit ratio, the greater the probability that the consumer request will be satisfied by the cache nodes. Defending the number of requests satisfied by cache nodes and the total number of content requested by the consumers as n and N , respectively, the cache hit ratio can be obtained from the ratio of n to N .

(ii) *Average hop count*: refers to the hop count in which a consumer's request reaches a cache node or the source server. It reflects the distance between the cache node and the consumer. The smaller the hop counts, the closer the cache node

to the consumer and the higher the efficiency of the entire system.

(iii) *Average transmission latency*: refers to the latency experienced by a consumer when the content request is provided to obtain the data. It can reflect the speed in which the network meets the request from consumers. Since the cache node is closer to the consumers than the source server, a smaller transmission latency and faster response to the requests will be achieved, which improves the quality of service (QoS).

5. Result Analysis

As shown in Figure 3, when the size of cache node varies from 100 to 2,000, the system performance has changed a

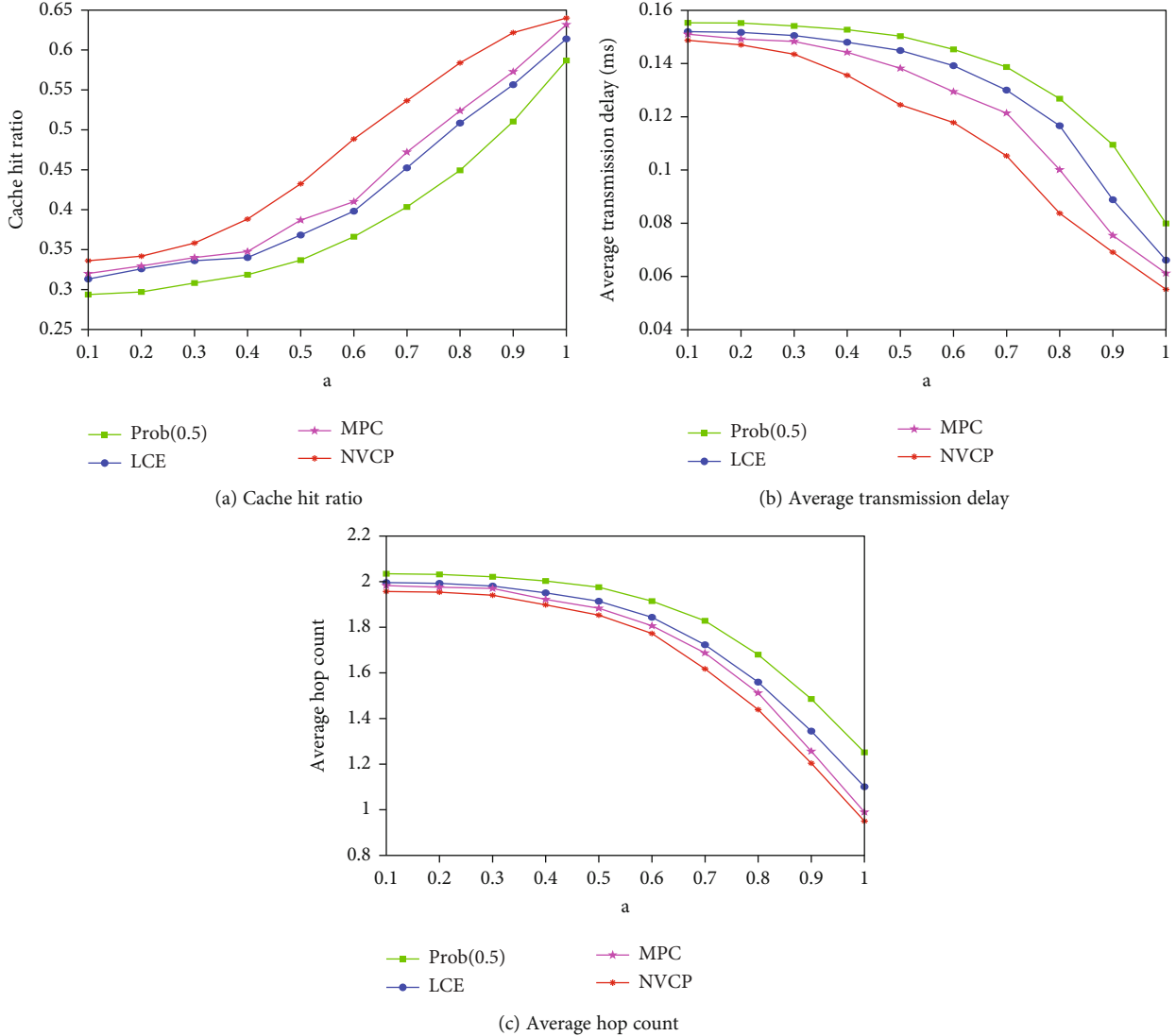


FIGURE 4: The impact of the index a on the system performance.

lot. Figure 3(a) shows that the cache hit ratios of the four cache strategies are gradually increased, and the cache hit ratio of the NVCP is significantly higher than the others. This is because the LCE requires all nodes on the delivery path cache content with no difference, which results in a large amount of content redundancy and frequent replacement. Prob(0.5) caches the content passing through the cache nodes with a fixed probability. Although the cache space is reduced, it still causes content redundancy and low content diversity. Instead of storing all the content at every node on the path, MPC [15] caches the content with high popularity. On the contrary, our proposed NVCP simultaneously considers the node value and content popularity. The content with higher popularity will be cached in nodes with a higher value, while those with lower popularity will be cached in nodes with a lower value; in this way, the update frequency and content redundancy can be reduced, and the content diversity will be improved. Compared to LCE, Prob(0.5), and MPC, the NVCP cache hit rate has increased by 11% to 15%.

Figures 3(b) and 3(c) show that as the cache capacity of the node increases, the average hop count and the average transmission delay decrease gradually, and the performance of NVCP is better than the others. This is because LCE caches content indiscriminately, Prob(0.5) takes the probability caching, and MPC caches content with the highest popularity, without requirements for caching nodes. However, our proposed NVCP evaluates the node value based on connectivity, betweenness centrality, and eigenvector centrality, and the weights are introduced for nodes with different requirements. Combining both content popularity and node value, the proposed NVCP is able to reduce the response time of content requests and the overhead of network management. Compared with the traditional cache strategies in CCN, the NVCP has a great improvement of the average hop count and the average transmission delay. Compared with LCE, Prob(0.5), and MPC, the average hop count of NVCP is reduced by 0.08~0.17 hops and the average transmission delay is reduced by 8~15 ms.

The Zipf distribution [20] was first proposed by the American linguist Zipf when studying the occurrence frequency of English words. If the occurrence frequency of words is arranged in descending order; there is a simple inverse relationship between the occurrence frequency and the rank of the word. Researches have shown that users' preference for content obeys Zipf distribution, and in this manuscript, we let index a indicate the concentration of the content, and the content with the bigger value of a indicates higher concentration of distribution. Our proposed NVCP makes use of the differences between the popularity of types of content, to make sure the cached content is distributed evenly, which simultaneously reduces the content redundancy and increases the diversity of content. We set the number of cache nodes as 1,000 and index a varying from 0.1 to 1.0. As shown in Figure 4, the performance of the proposed NVCP caching strategy is better than that of others, since NVCP takes the content popularity into account, leading to the decrease of content redundancy and the improvement in content diversity. Compared with LCE, Prob(0.5), and MPC, NVCP improves the cache hit ratio by 5%~8%, reduces average hop count by 0.08~0.16 hops, and decreases average transmission delay by 7~18 ms.

6. Concluding Remarks and Future Challenges

This paper investigated a novel NVCP-based collaborative caching strategy in massive VANET networks, which solves the replacement frequently or large amount of content redundancy provided by the conventional caching strategies, significantly reducing the cache redundancy and content replacement frequency, and improves the diversity of content. In this article, we also considered the influence of neighbor nodes when selecting cache nodes, without placing the content in neighbor nodes. By means of the simulation results, it is shown that the NVCP outperforms the LCE, Prob(0.5), and MPC in terms of the cache hit ratio, average hop count, and average transmission latency.

In the future work, we will consider placing the content in the neighbor nodes, to further improve the utilization of the cache space and achieve better cache performance. It is clear that, when the cache node is full, the cached content in the node is sorted by popularity; when new content arrives, its popularity is compared with the minimum content popularity. If that is less than the minimum content popularity, it will be forwarded to the consumer directly without caching; otherwise, the content with the minimum popularity is replaced and the replaced content will be placed in the neighbor nodes. After forwarding or caching the new content, the top 20% of the content popularity is multiplied by 0.5, and the content popularity within the cache node is reranked. In this way, the cache space of the neighbor node is utilized to reduce the frequency of replacing of the central node. Reprocessing the popularity of the content to prevent some content from becoming unpopular after some time, the cache space is occupied to improve the utilization of the cache space of cache nodes. Placing content on nodes which are highly centralized and close to the consumer can effectively reduce the data redundancy and latency, improving the QoS.

Data Availability

The authors declare that all the data and materials in this manuscript are available. In addition, a MATLAB tool has been used to simulate our concept.

Conflicts of Interest

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61971245 and Grant 61801249 and the Class C project funded by the 16th Batch of Six Talent Peaks High Level Talent Selection and Training in Jiangsu Province, grant number XYDXX245, sponsored by the Qinglan Project of Jiangsu Province of China and supported by the training object of the 6th "521 High-Level Talents Training Project" in Lianyungang.

References

- [1] A. Gatherer, *What Will 6G Be?*, IEEE ComSoc Technology News, 2018.
- [2] Y. Ji, W. Duan, M. Wen et al., "Spectral efficiency enhanced cooperative device-to-device systems with NOMA," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [3] W. Duan, J. Gu, M. Wen, G. Zhang, Y. Ji, and S. Mumtaz, "Emerging technologies for 5G-IoV networks: applications, trends and opportunities," *IEEE Network*, vol. 34, no. 5, pp. 283–289, 2020.
- [4] W. Duan, Y. Ji, J. Hou, B. Zhuo, M. Wen, and G. Zhang, "Partial-DF full-duplex D2D-NOMA systems for IoT with/without eavesdropper," *IEEE Internet of Things Journal*, to appear, 2020.
- [5] Y. Zhang, M. S. Hossain, A. Ghoneim, and M. Guizani, "COCME: content-oriented caching on the mobile edge for wireless communications," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 26–31, 2019.
- [6] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 121–127, 2011.
- [7] R. Steinmetz and K. Wehrle, "Peer-to-peer-networking & -computing," *Informatik-Spektrum*, vol. 27, no. 1, pp. 51–54, 2004.
- [8] M. Brunner, H. Abramowicz, N. Niebert, and L. M. Correia, "4WARD: a European perspective towards the future internet," *IEICE Transactions on Communications*, vol. E93-B, no. 3, pp. 442–445, 2010.
- [9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," *Communications of the ACM*, vol. 55, no. 1, pp. 117–124, 2012.
- [10] I. Psaras, W. K. Chai, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2920–2931, 2014.
- [11] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking,"

- IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2847–2886, 2016.
- [12] 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.
- [13] 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.
- [14] Y. Zeng, M. Jin, and H. Luo, “LICA: a segment-popularity based caching scheme in ICN,” *Acta Electronica*, vol. 44, no. 2, pp. 358–364, 2016.
- [15] C. Bernardini, T. Silverston, and O. Fester, “MPC: popularity-based caching strategy for content centric networks,” in *Proc. IEEE ICC 2013*, pp. 3619–3623, Budapest, Hungary, 2013.
- [16] M. Riondato and E. M. Kornaropoulos, “Fast approximation of betweenness centrality through sampling,” *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 438–475, 2016.
- [17] D. Vimalajeewa, S. Balasubramaniam, B. O’Brien, C. Kulatunga, and D. P. Berry, “Leveraging social network analysis for characterizing cohesion of human-managed animals,” *IEEE Transactions on Computational Social Systems*, vol. 6, no. 2, pp. 323–337, 2019.
- [18] J. Ren, W. Qi, and C. Westphal, “Magic: a distributed max-gain in-network caching strategy in information-centric networks,” in *In Computer Commun. Workshop (INFOCOM WKSHP)*, pp. 470–475, Toronto, ON, Canada, 2014.
- [19] Y. Ji, X. Zhang, W. Liu, and G. Zhang, “Replacement based content popularity and cache gain for 6G content-centric network,” *Physical Communication*, vol. 44, article 101238, 2021.
- [20] R. A. Fairthorne, “Empirical hyperbolic distributions (Bradford-Zipf-Mandelbrot) for bibliometric description and prediction,” *Journal of Documentation*, vol. 61, no. 2, pp. 171–193, 2005.