Gap-based Caching for ICN-based Vehicular Networks

Yuhong Li, Yusen Yang
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China

Rahim Rahmani, Theo Kanter
Department of Computer and Systems Sciences
Stockholm University
Stockholm, Sweden

Xiang Su, Jukka Riekki
Center for Ubiquitous Computing
University of Oulu
Oulu, Finland

Abstract—Information-centric networking (ICN) has been introduced into vehicular networks recently on account of its convenience for information distribution. Caching plays an essential role in information-centric networks. However, the current caching techniques for ICN cannot be used in ICN-based vehicular networks. This paper presents a gap-based caching approach that can be used in ICN-based vehicular networks. The proposed approach takes into account both the characteristics of vehicular networks and the information to be distributed. By introducing the “gap” metric and distinguishing the types of information, cooperative caching among nodes can be realized without exchanging caching management information among them. Simulation results show that the proposed approach can increase the storage space utilization and has low data response time in vehicular networks.

Keywords — Caching; Information-centric networking; Information popularity; Information priority

I. INTRODUCTION

In most early work, “on-path” caching is used [6][8]. In other words, every network node caches all the data passing through it. There is no “cooperation” among the nodes. Consequently, the same information is cached redundantly in the nearby network nodes. Moreover, caching new information may cause the replacement of old information due to the limited storage space in each node. The result is that some cached information may be replaced even before it is requested. It may often happen that a certain information object is cached everywhere in neighbouring nodes for a

Through caching the information in the network nodes on the return path to the requesters, the information can be gradually distributed to the edge of the networks, which is much closer to users. Accordingly, caching can reduce the time required for users to obtain the information and reduce the network traffic. This may also reduce the load on the servers providing the information. Of course, depending on the frequencies at which the cached information objects are requested by other nodes or are replaced by new objects, caching may entail an overhead.

Caching plays an essential role in information-centric networks [7]. However, the caching strategies discussed for wired networks may cause some problems when used in vehicular networks due to the temporal and spatial constraints of the information and the highly dynamic topologies of vehicular networks. Nevertheless, few caching strategies have been discussed for vehicular networks so far.

In this paper, we concentrate on caching in ICN-based vehicular networks. We propose a new caching approach for ICN-based vehicular networks. The proposed approach considers the characteristics of vehicular networks and the information to be distributed. By introducing the “gap” metric and distinguishing the types of information, cooperative caching among nodes can be realized without exchanging extra management information among them. The rest of the paper is organized as follows. First, the related work is discussed and challenges for caching strategies in vehicular networks are summarized. Following this, in Section 3, we propose a gap-based caching approach for ICN-based vehicular networks. The implementation and validation of the proposed approach are described in Section 4, and conclusions are drawn in Section 5.

II. CACHING AND ICN-BASED VEHICULAR NETWORKS

A. Problems of Caching in Information-centric Networks

In most early work, “on-path” caching is used [6][8]. In other words, every network node caches all the data passing through it. There is no “cooperation” among the nodes. Consequently, the same information is cached redundantly in the nearby network nodes. Moreover, caching new information may cause the replacement of old information due to the limited storage space in each node. The result is that some cached information may be replaced even before it is requested. It may often happen that a certain information object is cached everywhere in neighbouring nodes for a
certain time and then disappears in all the nodes. Hence, the limited storage space in the network nodes cannot be used efficiently. In some cases, only the caching in the “on-path” nodes is checked when there is a request although the requested information may be cached in nearby nodes. In other words, the network caching has not been well used.

The above mentioned problems have led to the emergence of cooperative caching strategies [9]. The main purpose of these strategies is to reduce the redundant caching amongst neighbouring nodes and increase the efficiency of the storage usage without decreasing the response time of information too much. A node can ask its close neighbours for certain information objects very quickly and then decide to cache the objects or not by considering if and how many copies of the objects have been cached in the close neighbours. Thus, to a certain extent redundant caching can be avoided. In addition, a node may already know in advance where a certain object is cached when there is a request. In this way cached objects can be well used and the transmission distance of the objects can be minimized.

Despite these advantages, there are some problems associated with cooperative caching. First, it is difficult to determine the extent of cooperation; in other words, to determine the groups of network nodes that should cooperate with each other. It is not easy to determine how many nodes should belong to a cooperating group or how big the cooperative extent should be. The dynamic nature of vehicular network topologies may make it even worse. Meanwhile, some networking overhead may be caused by caching management communication among the nodes. Second, most caching strategies concentrate only on reducing redundant caching, they do not consider the popularity of the information objects. As a result, a copy of an information object may be cached on several nodes, at a high price of communication, but it is seldom requested. Although in some work caching based on popularity has been proposed [10], vehicular networks were not discussed.

B. Challenges for Caching in Vehicular Networks

The large amount of data generated continuously place demands on the size of the cache in each vehicle. Also the overhead will decrease the performance for searching and replacing the cached information objects if too many are cached. Thus, caching is an important issue in ICN-based vehicular networks. Moreover, the highly dynamic topology of vehicular networks, the non-uniform distribution of vehicles, and the various types of information in the networks also pose challenges for caching in vehicular networks.

First, the topology of vehicular networks changes frequently when vehicles run at different speeds and in different areas such as motor ways and city roads. The vehicular networks with highly dynamic topology and non-uniform distribution of networking nodes require a flexible caching mechanism, which can adapt itself to the frequent changes of topology with variable vehicle densities.

Secondly, the various types of information have different temporal and spatial constraints. For example, an accident notification needs to be cached by vehicles within a two kilometre radius and for only about 30 minutes. On the other hand, a photograph of a landscape may be cached for days but is of interest only within a range of several hundred metres. Therefore, different caching strategies should be applied to different types of information in vehicular networks.

The highly dynamic topology and non-uniform distribution of networking nodes make it hard to use cooperative caching strategies in vehicular networks. On the one hand, the cooperative group may change too quickly and, as a result, the caching information from the neighbours may be not reliable so that the cooperation may fail even though the overhead for the cooperation has been paid. On the other hand, the density of vehicles in different areas makes it difficult to determine the extent of the cooperation groups. Thus, a non-cooperative caching is necessary in vehicular networks.

In order to overcome the disadvantages of non-cooperative caching, while achieving the advantages of cooperative caching, we propose a “gap” based caching approach for vehicular networks. We use the word “gap” because by using our method the information is not cached in every node. There are a certain number of gaps among the nodes that will cache the information depending on the distribution of vehicles and the types of information.

III. A CACHING APPROACH BASED ON GAP

From the data dissemination point of view, an important task of caching is to make the information reach the users as quickly as possible. This requires the information to be cached as widely as possible. However, if all the nodes have a copy of the same objects, storage is wasted. Therefore, our basic idea is first to classify the information according to type. The information objects that will frequently be requested by users or are most important for people on the road will be cached more widely than those not needed by most users. Secondly, the density of the vehicles on the road will be considered. The denser the vehicles in a certain area, the fewer the nodes need to cache the same objects, since it is relatively quick and easy for the nodes very close to each other to exchange different information objects, and more information can be stored in the nodes as a whole. Thirdly, we will realize the effect of cooperative caching without introducing much overhead. For this reason, caching instructions are delivered in the data packets instead of being exchanged separately. In the following sections, we will first describe how to classify the information in vehicles and then explain how to differentiate the information. Based on these, we will introduce the concept of “gap” and describe the caching approach based on “gap”.

A. Classification of Information in Vehicular Networks

Naming is the basis of ICN. Classifying and organizing the information properly can make the naming more logical and therefore help greatly in the forwarding and caching of information. According to the temporal and spatial constraints, the information in vehicular networks can be divided into two categories.

1) Safety-related information. This category includes information for safe driving and travel on the road. For
example, braking and lane change warnings are generated by events. This subcategory has normally small scope and short lifetime. Only vehicles nearby need cache this information and then only for a very short time; caching might even be unnecessary. Another subcategory is notifications, such as accident or road status (icy or under construction) notification. This subcategory needs to be cached relatively longer and in a bigger scope and both the temporal and spatial scope vary a lot.

2) Infotainment-related information. This category covers a variety of information for comfortable traveling on the road, such as music, jokes and video clips. This category may be of interested to different people for a long time. Weather, news and some location-based information services belong also to this category. But they have a relatively short lifetime compared with music and video clips.

Fig. 1 shows some typical values for different types of information. In order to make the information readily understood and used by all the vehicles, we name the hierarchically organized information in a way similar to a URI, as shown in (1).

\[
\text{categoryLevel1_id/categoryLevel2_id/.../info_id/chunk_id}
\]

(1)

For example, a piece of Schubert’s music is named /Infotainment/Entertainment/music/schubert/file1. Note that here the depth of each information object can be different.

<table>
<thead>
<tr>
<th>Category</th>
<th>Sub-Category</th>
<th>Caching time (scope)</th>
<th>P</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety</td>
<td>Event</td>
<td>braking warning, lane change warning: 30s, 500m</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>emergency stop, e.g. ambulance, fire: 5 min, 300m</td>
<td>10</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>accident</td>
<td>30 min, 2km</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>traffic status</td>
<td>1 hour, 3km</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>road status (icy, oily, obstacle)</td>
<td>24 hour, 5km</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Info-</td>
<td>Information service</td>
<td>sehen advertising, scenery guide etc.: 3 days, 500km</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>tainment</td>
<td>news</td>
<td>1 hour, 5km</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>weather</td>
<td>2 hour, 5km</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>interactive game</td>
<td>3 days, 6km</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>music/film</td>
<td>3 months, 1km</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>text/books etc.</td>
<td>1 year, 1km</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

Fig. 1. Categories of information and their caching constraint

B. Information Priority and Popularity

For safety-related information, the temporal and spatial constraints of each information object may differ very much. We use an integer value \( pr \) (priority) to differentiate the objects of this type of information. The smaller the scope and the shorter the time, the larger the \( pr \) of the objects. A larger \( pr \) means the objects will be cached in more nodes in a certain area (i.e., more densely). When an information object of this type is generated through sensing or monitoring the environment, the value \( pr \) is determined. Thus, we have \( pr \in \mathbb{R}, R = \{pr_1, pr_2, ..., pr_n\} \), \( nr \) is the number of priority levels defined for the safety-related information.

For the infotainment-related information, there is no big difference in the temporal and spatial constraints among the individual objects of the same subcategory. But the number of requests for a certain information object can well reflect the need and efficiency for caching them in the networks. Thus, we use “popularity”, which indicates the number of requests for an object in a certain unit of time, in our caching approach. Considering the continuity and dynamicity of the information requests, the number of requests in the current time unit \( t_n \) and the previous time unit \( t_{n-1} \) will be both stored in the Popularity Table. Then, we calculate the current popularity of an information object using: \( pop = t_n \times \alpha + t_{n-1} \times (1-\alpha) \). Here \( \alpha \) is the weight of the popularity of an object in the history. It can be given according to the characteristics of the information and/or the length of the unit of time.

In order to unify the popularity with the priorities, we define popularity thresholds \( th_1 < th_2 < ... < th_{np} \) and quantize the popularity, namely we have \( p \in \mathbb{P}, P = \{pop_{grad_1}, pop_{grad_2}, ..., pop_{grad_{np}}\} \), and \( pop_{grad} \) is the level of the popularity, and \( np \) is the number of popularity levels defined for the information. If \( th_{np} < pop < th_{np-1} \), then \( p = pop_{grad} \).

Similar to the discrete popularity, we use also the discrete density for the vehicle distribution. Namely, we have density \( d \), \( d \in \mathbb{D}, D = \{density_{grad_1}, density_{grad_2}, ..., density_{grad_{nd}}\} \), \( nd \) is the number of density levels of the vehicle distribution.

C. Caching Based on GAP

We introduce the “gap” metric, where \( gap = f_1(p,d) \) or \( gap = f_2(p,d) \) to denote in which node a certain information object should be cached. In other words, \( gap \) is a function of the popularity or priority of the information disseminated in the networks and the density of the vehicle distribution at a certain time. It can be defined in a flexible way depending on the amount of information in the network and the scale of the network. In our approach, a node will use \( gap \) to determine if it should cache an object when forwarding it together with the parameter of distance between the requester and the node that owns or caches the requested information.

The principle of our approach is to add one “hop” field in the information request packet (i.e., Interest Packet), and two fields, “hop” and “gap”, in the response packet (i.e., Data Packet), after the “name” field in the two types of packets.

After a node receives an Interest Packet requesting an object, the node will check the Caching Store (CS) and the Pending Interest Table (PIT), similar to the processing in NDN [6]. The biggest difference is that there is a “hop” field recording the distances in hops from the information requester to the current node in the Interest Packet right after the “name” field. At the requester node, the hop field is inserted and its value is set to zero. When a node cannot satisfy the request from its own CS, it will forward the Interest Packet with the hop value incremented by one. Thus, when the requested object is found in the network, all the intermediate nodes know the number of hops to the node where the information object is cached.

When the Interest Packet reaches the server having the requested object or an intermediate node caching the object, one or more Data Packets will be created and sent. The value of the “hop” field in the Interest Packet will be assigned to the “hop” field in the Data Packet. Meanwhile, the request number of the same object will be accumulated and the value of the gap will be calculated and inserted in the “gap” field.

When a node receives a Data Packet, as shown in Fig. 3, if the corresponding data has been requested and has not been delivered, the node will check the value of the hop and gap
carried in the packet to determine whether it should forward and/or cache the data. In this case the result of hop modulo gap (i.e., hop % gap) is used by the node to determine whether it should cache the data. The result of hop modulo gap determines how frequently a copy of data should be cached on its path to the requester. The larger the gap, the fewer the nodes cache the data. Here, PIT denotes the Pending Interest Table, and CS denotes the Caching Store.

IV. IMPLEMENTATION AND VALIDATION

We have implemented the gap-based caching approach and integrated it as an independent module in the ndnSIM. ndnSIM [11] is an NDN simulator for ns-3 [12]. It is a ns-3 module that implements the NDN communication’s model. The simulator implemented all the entities supporting an ICN network infrastructure. Meanwhile, we use VanetMobiSim, a vehicular Ad Hoc networks mobility simulator [13] to simulate the movements of vehicles and generate the tracks of vehicle movement. Fig. 3 illustrates an example of the vehicle network topology when there are 50 vehicles in an area of 1000m×1000m. The speed of the vehicles is between 10-20 m/s. The vehicles use WiFi to send and receive data.

We evaluate the gap-based caching approach according to the following metrics:

- Redundancy: the number of copies of an object that are cached in the networks after a certain number of requests.
- Space Utilization Ratio: the proportion of used storage space to the total space of a node.
- Cache Hit Ratio: the probability to obtain a cache hit along the path from a request node to a node caching the requested information object.

Fig. 4 illustrates the average redundancy of information when there are different numbers of vehicles in the network. From this figure, we can see that the average number of copies of data when using gap-based caching is less than that when the data are cached everywhere. This may decrease the ratio of space utilization at the nodes and more information objects can be cached in the networks.

Fig. 5 shows the redundancy of information objects with different popularities and priorities. Here the objects with higher opportunities of requests (denoted in percentage) are more popular. We can see that objects with higher popularity or priority have more copies in the network than those with lower popularity or priorities. In addition, when vehicles are more densely distributed (the same number of vehicles in a smaller area), more copies of objects are cached.

Fig. 2. Caching based on gap when receiving data

Fig. 3. Random distribution of vehicles

Fig. 4. Average redundancy of information

Fig. 5. Redundancy of information with different popularities and priorities
The average space utilization of the vehicle nodes is shown in Fig. 6. Here we can see that the space utilization ratio of the nodes when gap-based caching is used is much less than that when information is cached everywhere. This is because information with lower priorities or not often requested are cached in the network more sparsely.

Fig. 6. Average space utilization ratio in the network

Fig. 7 shows the cache hit ratio of caching based on gap and caching everywhere. Here we can see that when gap-based caching is used, the hit ratio decreases slightly. This is because the objects not often visited or with lower priority are cached more sparsely in the networks in order to increase the efficiency of storage. However, the average response time of information request, as shown in Fig. 8, is not affected by the poor cache hit ratio because these objects are seldom requested.

Fig. 7. Average cache hit ratio in the network

We also compared the response times for two methods. The response time is the time from when a request is sent when the first packet of the requested data is received. As shown in Fig. 8 our gap-based caching approach did not introduce much average delay for the information in the network.

Fig. 8. Average response time for obtaining information

Fig. 9 shows the average distance (in hops) for an Interest Packet to information objects in the network. From this figure, we can see that using the gap-based caching the average distances for finding information in the network will not increase much.

Fig. 9. Average distance to information in the network

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new method for caching data in ICN-based vehicular networks. The characteristics of the information and its popularity and priority are considered during caching. In addition, the distribution density of the vehicles has also been taken into account. Simulation shows that differentiation of information reduces markedly caching redundancy thus saving storage space in the networks. This saving hardly affects the average response time for obtaining information from the network. Our future work is to perform a more thorough analysis and evaluation of the characteristics of information and further consider the dynamicity of network topology on caching.

REFERENCES