

Article

# CCN-Based Inter-Vehicle Communication for Efficient Collection of Road and Traffic Information

Takanori Nakazawa, Suhua Tang \*  and Sadao Obana

Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu, Tokyo 182-8585, Japan; t.nakazawa@can.lab.uec.ac.jp (T.N.); sa-obana@uec.ac.jp (S.O.)

\* Correspondence: shtang@uec.ac.jp; Tel.: +81-42-443-5308

Received: 6 December 2019; Accepted: 30 December 2019; Published: 7 January 2020



**Abstract:** Recently, inter-vehicle communication, which helps to avoid collision accidents (by driving safety support system) and facilitate self-driving (by dissemination of road and traffic information), has attracted much attention. In this paper, in order to efficiently collect road/traffic information in the request/response manner, first a basic method, Content-centric network (CCN) for Vehicular network (CV), is proposed, which applies CCN cache function to inter-vehicle communication. Content naming and routing, which take vehicle mobility into account, are investigated. On this basis, the CV method is extended (called ECV) to avoid the cache miss problem caused by vehicle movement, and is further enhanced (called ECV+) to more efficiently exploit cache buffer in vehicles, caching content according to a probability decided by a channel usage rate. Extensive evaluations on the network simulator Scenargie, with a realistic open street map, confirm that the CV method and its extensions (ECV, ECV+) effectively reduce the average number of hops of data packets (by up to 47%, 63%, and 83%, respectively) and greatly improve the content acquisition success rate (by up to 356%, 444%, and 689%, respectively), compared to the method without a cache mechanism.

**Keywords:** inter-vehicle communication; content-centric network; cache miss; channel usage rate

## 1. Introduction

Inter-vehicle communication (IVC) is an important method for a driving safety support system, via which vehicles exchange positions and speed information to prevent collision accidents. This technique has been put into practical use in Japan since 2015, and still draws a lot of research attention in the world [1–4].

Various types of information are handled by the IVC, and can be roughly classified into two categories. One category is safety-related information (e.g., sudden brake of vehicles, emergency vehicle approach). Such information must be delivered to all vehicles fast and reliably, and failing to do so may lead to traffic accidents. For this reason, dissemination via broadcast in the push manner is effective for this category, which is characterized by the fact that information is disseminated without an explicit request [1]. The other category is driving comfort-related information (e.g., road closure, and congestion at forward intersections). Such information is not directly related to traffic accidents, and a delay in dissemination is allowed. In addition, vehicles with different routes usually have different interests. Therefore, dissemination via request/reply in the pull manner is often used for this category, which is characterized by the explicit request. This paper focuses on the 2nd category. However, as the number of vehicles requesting information increases, congestion due to repeated transmission of the same information becomes a problem.

In order to solve this problem, this paper suggests applying a content-centric network (CCN [5]) for the efficient dissemination of road/traffic information in IVC [6]. CCN, often used for content dissemination in the Internet, has some points in common with the pull type dissemination in IVC.

Content caching function of CCN is expected to prevent redundant dissemination and reduce the number of forwarding of data packets in IVC. However, there are also significant differences between IVC and CCN, as follows: (1) CCN usually assumes that the content is static and remains unchanged after creation, but, in IVC, the content (traffic event) changes dynamically because the traffic and road conditions change continuously [7]. Therefore, conventional cache management policy in CCN cannot be directly applied in IVC. (2) CCN is designed for content dissemination on the Internet, where content provider and routers almost do not move. In comparison, vehicles move at fast speeds in IVC. In addition, it is necessary to devise a unique name for information exchanged by IVC. For example, when a certain traffic jam is detected by multiple vehicles and assigned different names, other vehicles may think multiple traffic jams exist. In summary, in order to apply CCN for efficiently disseminating (Pull type) road/traffic information in IVC, the following issues must be taken into account:

- Problem 1: Content naming. How should traffic events be named uniquely (and in a way easy to understand) when it is possible for multiple vehicles to detect the same traffic event?
- Problem 2: Routing Interest/Data packets. Network topology is assumed to be almost unchanged in CCN, but, in IVC, it changes constantly because the content requester, provider, and relay vehicles always move. Thus, efficient routing is a challenge.
- Problem 3: Content cache management. How can the cache be managed efficiently when content changes frequently and vehicles move?

In this paper, we first propose a basic method, CCN for vehicular network (CV). For Problem 1, a unique name is given for a traffic event by using intersections and the type of content, and, for Problem 2, location based routing is adopted, and a relay vehicle is selected according to the progress made by each vehicle. As described in Problem 3, with location-based routing, it is preferred that the farthest vehicle be selected as the relay in order to reduce the number of hops. However, if the farthest vehicle does not have specified content in the cache buffer, while other nearer vehicles do, the cache miss problem occurs. In addition, it is inefficient for all vehicles to cache the same content. Therefore, the basic CV method is further extended to address the cache miss problem (the ECV method) and cache content according to a probability decided by the channel usage rate (the ECV+ method). Simulation evaluations on network simulator Scenargie, with a realistic open street map, confirm that the proposed methods significantly reduce the number of hops (number of content transmission) and improve the content acquisition success rate, and ECV+ achieves the best performance. The basic method was already presented at APCC (Asia-Pacific Conference on Communications) 2018 [8]. In this paper, a new cache management policy is introduced to better exploit the storage in vehicles to cache more contents, and more evaluations (e.g., the impact of cache buffer capacity, vehicle density) are added to evaluate the effectiveness of the proposed method in different scenarios.

The rest of this paper is organized as follows: Section 2 reviews background and related works on CCN. Section 3 proposes the basic method for applying CCN to IVC, analyzes the problem of cache miss, and makes extensions from different aspects. Then, Section 4 presents the evaluation results. Finally, Section 5 concludes this paper and points out future work.

## 2. Related Works

This section reviews related research on CCN and routing.

### 2.1. CCN

CCN is a network architecture that enables efficient content delivery [5]. When requesting content, the requester issues an Interest packet specifying the content name instead of the content provider's address, and sends it towards the content provider. The requested content is sent back in the Data packet (response) from the provider and cached in the intermediate relay nodes. The cache is used later when the same content is requested again by other vehicles, and in this way the dissemination efficiency is improved. To this end, each CCN node has three functions: a CS (Content Store) for

storing the content cache, an FIB (Forwarding Information Base) for delivering Interest packets towards the provider, and a PIT (Pending Interest Table) for forwarding Data packets towards the requesters. CCN is expected to reduce the number of content delivery hops and improve the content acquisition success rate.

Existing work on CCN [9,10] can be broadly divided into naming, routing, and caching [11]. In research on naming, it is possible to identify content information by a hierarchical prefix [12]. This method differs from the proposed method in that it focuses on naming and is only effective when the content is static. To apply this to IVC, it is necessary to consider what should be classified as road/traffic information identifiers. With respect to routing, it has been proposed that a city map be divided geographically and used for content search and routing by defining its own partition [13]. However, the CCN implementation is still an application layer overlay.

Many efforts have been devoted to cache. A cooperative caching algorithm is suggested to use cache buffer effectively [14]. Although it has achieved promising results with excellent cache efficiency, it has the disadvantage of not being able to handle dynamic content. Caching with a probability [15] is suggested to avoid redundant caching, and various methods have been put forward for calculating the cache probability, such as using the number of hops, using content popularity, or considering cache allocation. In [16], an efficient method of determining popularity is defined, and an excellent caching method is proposed by using intermediate routers and edge nodes. This method is a very good cache allocation method from the viewpoint of content popularity. However, the number of assumed contents is small, and it does not deal with the problems peculiar to IVC, and not consider routing in actual environments. In [17], in order to reduce cache redundancy and improve cache hit rate, the importance of each node associated with the shortest path is exploited. In order to reduce frequent cache exchanges and to guide paths that lead to cache hits, a new hierarchy caching scheme is considered. This is a strategy based on node importance and content popularity, and pairing using a probability matrix. This approach diversifies node usage for caches and avoids redundant caches. In [18], factors affecting the ICN (information centric network) cache in VANET (vehicular ad hoc network) are discussed, as follows: (1) Vehicles are reluctant to follow cache rules from a privacy perspective. (2) Network topology changes dynamically due to the high speed movement of vehicles. (3) Vehicle cache storage space and bandwidth resources are relatively small compared to base stations and routers, and cache redundancy needs to be reduced. This paper proposes community similarity and population base, introduces content popularity into hop number based caching vehicle selection method, and reduces cache exchange overhead through prediction-based coordinated cache replacement.

Table 1 below presents a comparison of aforementioned related works. As shown in this table, CCN is actively researched, but naming, routing, and cache usually are separately studied, and the mobility feature of IVC is not well considered. When CCN is implemented as an overlay in the application layer, it is necessary to update the FIB whenever the network topology changes, which causes a lot of overhead. In addition, the movement of vehicles also affects cache hit rate. Although different caching schemes have been explored, little work takes into account the impact of wireless channel. In this paper, we consider the influence of the channel usage rate on the cache instead of the popularity because the lifetime of the content exchanged by IVC is short.

## 2.2. Routing

When CCN is applied to IVC, content distribution depends on the choice of relay vehicles. As for relay selection based on location information, the farthest vehicle (within the transmission range of the sender) is generally preferentially selected as a relay vehicle in order to improve relay efficiency [1]. In this way, the transmission distance in one hop can be extended as much as possible, and the number of hops required for dissemination can be reduced.

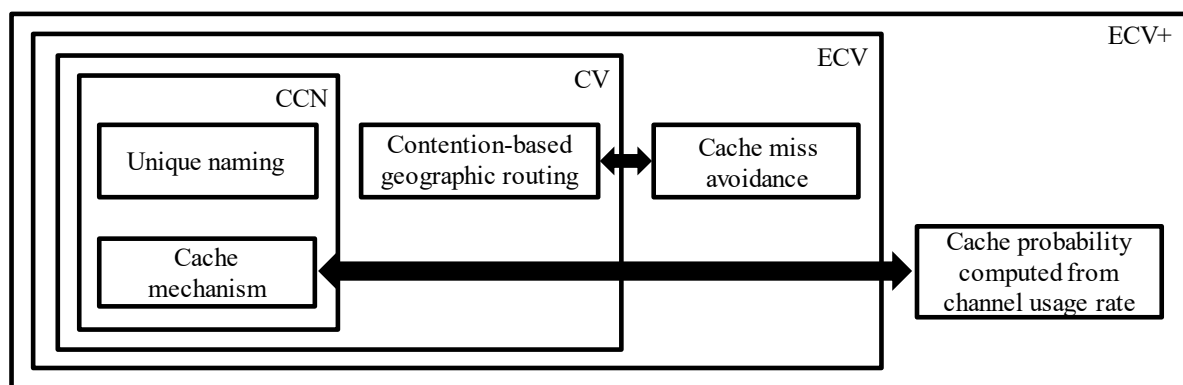
**Table 1.** Comparison of related works.

Related Works	Task	Pros	Cons
Hierarchical Semantics [12]	Naming	Hierarchical prefix that can identify content	Valid only if content is enacted
Hierarchical Bloom-Filter Routing [13]	Routing	Excellent content searchability with unique partition definition	CCN is still implemented by overlay
Collaborative Caching Based on Social Relations [14]	Cache	Cache placement algorithm with excellent cache efficiency	Cannot handle dynamic content
ProbCache [15]	Cache	Avoid redundant caches by exploiting hop count and popularity	Not consider content lifetime in inter-vehicle communication
Distributed Caching Strategy [16]	Cache	Excellent popularity definition and cache placement method	An efficient routing method for IVC is required.
Hierarchical Cluster-Based Caching [17]	Cache	Diversify node usage for caching	Need to be linked to specific naming and routing
Community Similarity and Population-based Cache Policy [18]	Cache	Reducing overhead of cache exchange by prediction-based coordinated cache replacement	Need complicated routing and naming to realize cache strategy

Packet reception status changes due to vehicle movement and channel fading in IVC. Vehicles that received a packet select the next relay vehicle in an autonomous and distributed manner based on the same position (distance) information. This share of position information between vehicles can be realized by using information exchange in the 700 MHz band for driving safety support system [19]. The number of hops is reduced by selecting the farthest vehicle within the communication range of the transmitter. Specifically, potential relay vehicles are sorted according to their distance (progress) from the transmitter, and their distance order is associated with different waiting time (in terms of slots) before forwarding. By assigning a shorter waiting time (an earlier slot) to a vehicle farther away from the transmitting vehicle, among all vehicles that are within the communication range of the transmitter and have received the packet, the farthest vehicle becomes the relay vehicle because it starts forwarding in the earliest time, which makes other potential relays cancel their forwarding.

### 3. Proposed Method

Figure 1 shows the framework of the proposed methods, where the left side corresponds to the basic CV method, and the right side is its extensions, which further solve the cache miss problem in the mobile environment (ECV), and presents the cache optimization method based on channel usage rate (ECV+).



**Figure 1.** Proposed framework for applying CCN in IVC.

### 3.1. CCN for Vehicle Network

The CCN for Vehicular Network (CV method) is the baseline. In this method, we propose a naming mechanism and a routing mechanism for efficient dissemination.

#### 3.1.1. Content Naming (Solution of Problem 1)

The content name is composed of (1) geographic information and (2) road/traffic information indicating the type of event.

##### (1) Geographic information naming

Each intersection on the road is assigned a name (using coordinates), and intersection names are used as high level geographical information. Then, the name of the road link (hereinafter referred to as link) from intersection A to B is "AB". Intersection A is simply "A", and "AB/BC/C" shows the entire route from intersection A to C via links AB and BC.

##### (2) Naming of road and traffic information types

The names and values of the information types are listed in Table 2.

**Table 2.** Types of road and traffic information.

Information Type	Value
Passability judgment	Passable/not passable
Congestion	Normal/Slightly congested/Congested
Road surface condition	Normal/Freezing/Poor visibility, etc.

#### 3.1.2. Routing (Solution of Problem 2)

The original definition of FIB and PIT in CCN is modified. Any vehicle that received an Interest packet can be a relay vehicle. Here, in order to improve relay efficiency, among the vehicles that have received an Interest packet, the vehicle farthest away from the transmitter (with the longest progress) is preferentially selected as the relay vehicle [1]. The location information of surrounding vehicles is shared in advance and compared instantly. Within the communication range of the transmitter, the longer distance (progress) a vehicle is from the transmitter, the shorter waiting time it is set in the MAC (media access control) layer. In this way, the farthest vehicle is selected as the relay vehicle. The whole process consists of the following three steps, as shown in Figure 2:

- I. Vehicle(s) sends an Interest packet. After receiving an Interest packet, the relay vehicle is selected autonomously by setting the waiting time for the relay candidates. Figure 2 shows how the waiting time is set in terms of slots, where a vehicle with a longer distance has a shorter waiting time. For example, vehicle(a) is assigned the 1st slot (with the shortest waiting time), vehicle(b) is assigned the 2nd slot, vehicle(c) is assigned the 3rd slot, and vehicle(d) is assigned the 4th slot, according to their distance from the sending vehicle(s).
- II. If the farthest vehicle has the required content in its cache buffer, it returns the content in the Data packet, otherwise it forwards the Interest packet.
- III. A Data packet is forwarded in the same way as an Interest packet. A vehicle that overhears the Data packet keeps a copy in its cache buffer.

Algorithm 1 explains the whole process including how to process Interest (lines 1–15) and Data (lines 16–27) packets. When an Interest or Data packet is received, at first, all neighbors of the sender,  $n_{\text{neigh}}$ , within the transmission range of the sender, is obtained. Then, the distance (progress),  $d_{\text{neigh}}$ , from the sender to  $n_{\text{neigh}}$ , is computed, which is further sorted in decreasing order. Each vehicle finds its own distance order  $i_{\text{own}}$ , and, on this basis, sets a timer. When the timer expires and the channel

remains idle since its setting, the vehicle obtains the right to transmit, either a Data packet if it has the required content in its cache buffer, or the Interest packet otherwise.

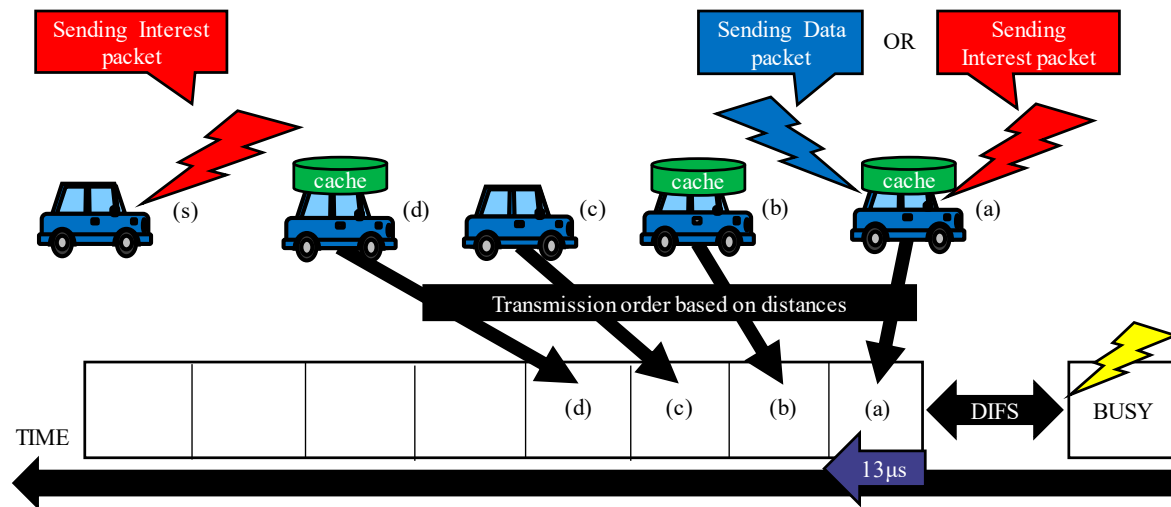


Figure 2. CV method.

**Algorithm 1** Processing Interest and Data packets in the CV method

```

1: procedure OnReceivingInterestPacket()
2:   Get current sender and target area from Interest packet
3:   Get all neighbors of sender,  $n_{neigh}$ , within the transmission range of the sender
4:   Compute progress (distance),  $d_{neigh}$ , from sender to  $n_{neigh}$  in the direction of target area
5:   Sort  $d_{neigh}$  in the decreasing order
6:   Find own order in the sorted  $d_{neigh}$  as  $i_{own}$ 
7:   Set a timer,  $t_{wait} = t_{slot} \times i_{own}$ ,  $t_{slot}$  is the slot length
8:   if (Timer expires and channel remains idle) then
9:     if (Required content is in the cache buffer) then
10:      Start transmitting Data packet as a response to the Interest packet
11:     else
12:      Start forwarding Interest packet
13:     end if
14:   end if
15:   return
16: procedure OnReceivingDataPacket()
17:   Get requester (destination of Data packet) and its current sender
18:   Get all neighbors of sender,  $n_{neigh}$ , within the transmission range of the sender
19:   Compute progress (distance),  $d_{neigh}$ , from sender to  $n_{neigh}$  in the direction of requester
20:   Sort  $d_{neigh}$  in the decreasing order
21:   Find own order in the sorted  $d_{neigh}$  as  $i_{own}$ 
22:   Set a timer,  $t_{wait} = t_{slot} \times i_{own}$ ,  $t_{slot}$  is the slot length
23:   if (Timer expires and channel remains idle) then
24:     Start forwarding Data packet
25:   end if
26:   Store Data packet in cache buffer with probability 1
27:   return

```

In the basic method, the cache miss problem occurs when the farthest vehicle in the communication range of the transmitter actually does not have the required content in its cache buffer, although other closer vehicles do. This problem is explained by Figure 3. When vehicle (b) is the farthest vehicle (within the neighbors of vehicle (s)), the cache of vehicle (b) is hit. However, when vehicle (a) moves

near and becomes the farthest vehicle, it has no required content in the cache buffer. Due to the nature of giving priority to the farthest vehicle, vehicle (a) obtains the right to transmit. However, it cannot detect whether there is a cache in other nearby vehicles, and decides to relay the Interest packet, which leads to the cache miss problem in the CV method.

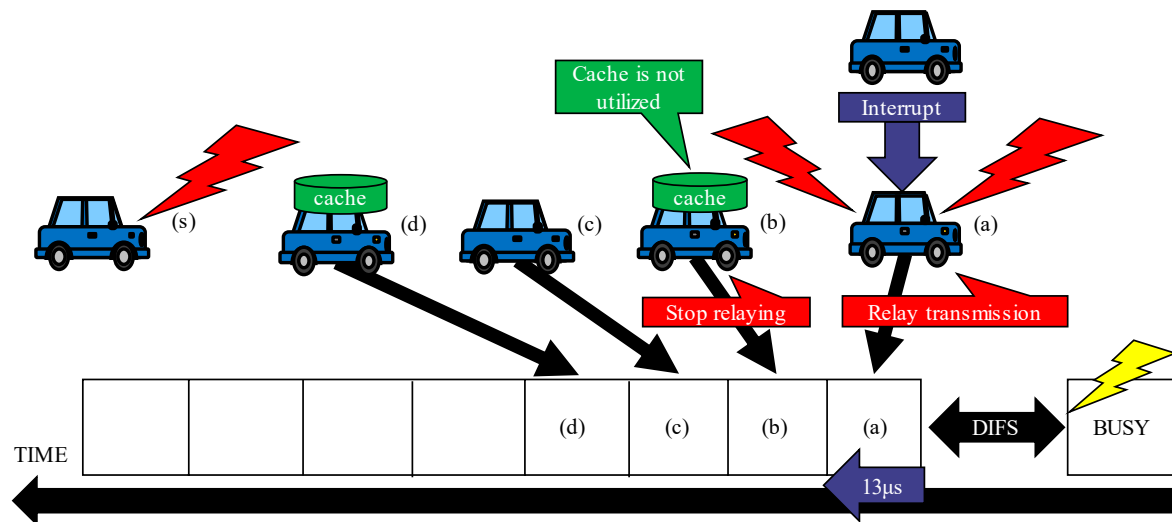


Figure 3. Problem case.

The other problem of the basic method is cache redundancy. A copy is kept on all vehicles that have overheard a Data packet. As a result, this hinders efficient use of vehicle storage to store more contents.

### 3.2. Extended CV Method

In order to solve the cache miss problem, we propose the Extended CV (ECV method) that realizes cache miss avoidance function. In the CV method, slot assignment for Interest/Data packet transmission is unified, and the decision of forwarding Interest packets and Data packets is based on the same slot. Because the vehicle farthest away from the transmitter does not know whether other closer vehicles have the required content, when it, on its own, does not have required content, it decides to start forwarding the Interest packet at its own slot, which may lead to the cache miss problem. In the ECV method, replying to an Interest packet (by a Data packet) and forwarding an Interest packet are divided into two separate stages, and replying a Data packet is given higher priority. Specifically, the ECV method first allocates slots for relay selection for replying a Data packet and then allocates different slots for relay selection for forwarding the Interest packet. If any vehicle with the required content replies a Data packet in the first stage, the relaying of the Interest packet is cancelled. Otherwise, the farthest vehicle continues to forward the Interest packet as in the basic CV method.

Figure 4 shows how each vehicle is assigned two slots, one for Data packet and the other for Interest packet. The slots for the Data packet are assigned before those for the Interest packet so that any vehicle with the required content can reply a Data packet. Now, the relay selection for replying the Data packet adopts a different policy, the closer a vehicle is to the transmitter (of the Interest packet), the higher priority it has because a shorter distance generally means a higher transmission reliability. After that, the farthest vehicle is selected for forwarding the Data packet in the same way as for Interest packets. The slot allocation for Interest packet is the same as in the basic CV method.

For example, in Figure 4, because vehicle (d) is the closest to the sender (s), its transmission priority of the Data packet is the highest and its transmission priority of the Interest packet is the lowest. Conversely, because vehicle (a) is the farthest from the sender (s), its transmission priority of

the Data packet is the lowest and its transmission priority of the Interest packet is the highest. In this way, the cache hits if any vehicle holds the required content.

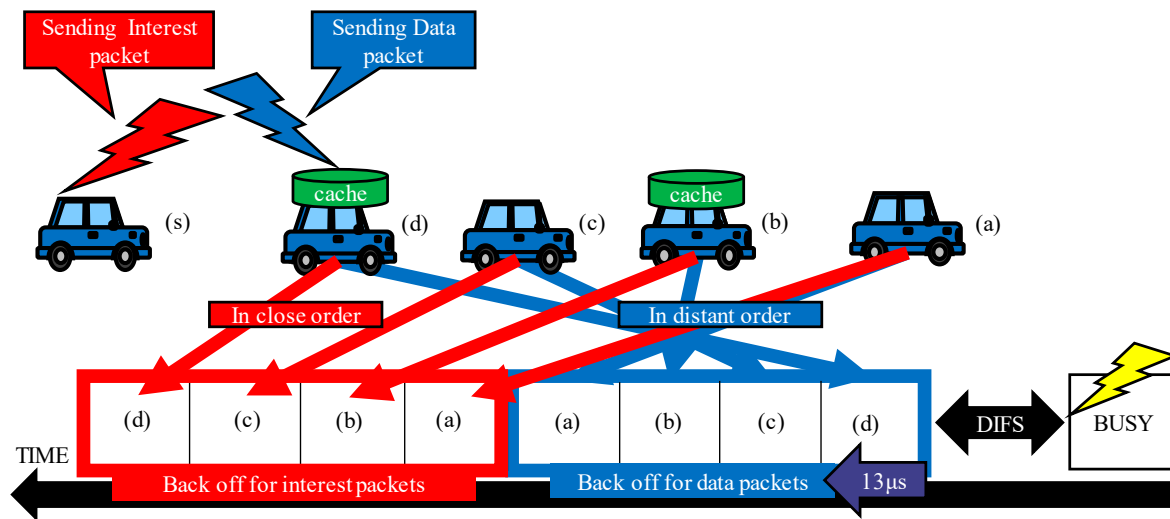


Figure 4. ECV method.

The difference between the ECV method and the CV method lies in the processing of Interest packet, which is described by Algorithm 2: the slots for replying a Data packet (1st stage, lines 6–13) and forwarding the Interest packet (2nd stage, lines 15–20) are separated.

---

**Algorithm 2** Processing Interest packet in the ECV method

---

- 1: procedure OnReceivingInterestPacket()
  - 2: Get requester, sender and target area from Interest packet
  - 3: #1st stage, contend to reply Data packet, nearest distance first
  - 4: Get all neighbors of sender,  $n_{neigh}$ , within the transmission range of the sender
  - 5: Compute progress (distance),  $d_{neigh}$ , from sender to  $n_{neigh}$  in the direction of requester
  - 6: Sort  $d_{neigh}$  in the increasing order
  - 7: Find own order in the sorted  $d_{neigh}$  as  $i_{own}$
  - 8: Set a timer,  $t_{wait} = t_{slot} \times i_{own}$ ,  $t_{slot}$  is the slot length
  - 9: **if** (Timer expires and channel remains idle) **then** # no other node replies
  - 10:     **if** (Required content is in the cache buffer) **then**
  - 11:         Start transmitting Data packet as a response to the Interest packet
  - 12:     **end if**
  - 13: **end if**
  - 14: #2nd stage, contend to forward Interest packet, longest distance first
  - 15: Sort  $d_{neigh}$  in the decreasing order
  - 16: Find own order in the sorted  $d_{neigh}$  as  $i_{own}$
  - 17: Set a timer,  $t_{wait} = t_{slot} \times i_{own}$ ,  $t_{slot}$  is the slot length
  - 18: **if** (Timer expires and channel remains idle) **then**
  - 19:     Start forwarding Interest packet
  - 20: **end if**
  - 21: **return**
- 

### 3.3. Further Enhancement of the ECV Method

We notice that, when vehicle density is high enough, the cache hit performance will not deteriorate much even if the same content is cached only at some vehicles, compared to the case where all vehicles cache the same content. Therefore, probability-based cache management is introduced to better share



vehicle cache buffer and improve cache efficiency, and this method is called ECV+. In the ECV method, cache probability is always 1, whereas, in the ECV+ method, cache probability is  $p_{cache} < 1$  ( $p_{cache}$  changes dynamically). This method does not consider the popularity which reflects how often a content is required. This is because a content in IVC has a short lifetime, and the popularity becomes meaningless whenever the content reaches its lifetime.

If cache probability is too small in order to increase the diversity of cached content, the cache hit rate may decrease, when a content is not cached by any vehicle, and the number of hops will increase. As a result, the amount of communication increases and the channel usage rate increases. On the other hand, if the cache probability is too large, vehicles tend to cache the same content, and cache hit rate decreases if diverse contents are required but not hit in any vehicle, which also affects the amount of communication. Therefore, it is necessary to dynamically adjust the cache probability according to a channel usage rate.

Based on the above description, the ECV+ method compares the channel usage rate before and after content delivery, and performs a heuristic feedback control of cache probability, in order to increase cache hit rate and avoid channel congestion. The whole process is described by Algorithm 3, where the process is run periodically. Specifically, cache probability  $p_{cache}$  is adjusted by comparing the difference between past ( $\rho_{old}$ ) and current ( $\rho_{new}$ ) channel usage rate with a threshold ( $\rho_{th}$ ), in order to evaluate the effectiveness of the previous control. It is expected that the period is long enough so that channel usage rate already becomes stable under the new cache probability and also short enough so that the algorithm can track the change in network traffic. Similarly, as for the adjustment value  $\Delta$ , too large a value makes the system unstable while too small a value makes it difficult to track traffic change. The period parameter and adjustment value  $\Delta$  are set to 1s and 0.1, respectively, by initial experiments.

Since caching decision is probabilistic, redundant cache can be avoided, and the amount of cache is adjusted to an appropriate value by dynamically changing cache probability. Vehicle density, another factor that may affect cache probability, although not explicitly considered, is indirectly involved in the control.

---

**Algorithm 3** Control of cache probability

---

```

1: procedure UpdateCacheProbability()
2:    $\rho_{old} \leftarrow 0.5$            # initialize channel usage rate
3:    $p_{cache} \leftarrow 0.5$       # initialize cache probability
4:   while (true) do
5:      $\rho_{new} = \text{GetChannelUsageRate}()$ 
6:     if ( $|\rho_{new} - \rho_{old}| > \rho_{th}$ ) then
7:       if ( $\rho_{new} < \rho_{old}$  and  $\rho_{new} < \rho_{max}$ ) then
8:          $p_{cache} \leftarrow p_{cache} - \Delta$ 
9:       else if ( $(\rho_{new} > \rho_{old}$  and  $\rho_{new} < \rho_{max})$  or  $\rho_{new} > \rho_{max}$ ) then
10:         $p_{cache} \leftarrow p_{cache} + \Delta$ 
11:       end if
12:     end if
13:      $\rho_{old} \leftarrow \rho_{new}$ 
14:     Sleep for a period (1 second)
15:   end while
16: procedure GetChannelUsageRate(): Function to get channel usage rate

```

---

### 3.4. A Short Comparison of Three Methods

A short comparison of three methods is shown in Table 3. Generally, CV is a baseline, which implements the basic CCN functions, although it has the cache miss issue. ECV solves the cache miss issue by giving higher priority to the response of Data packet. On this basis, ECV+ tries to better improve the efficiency of cache buffer in neighbor vehicles by computing cache probability from channel usage rate.

**Table 3.** Comparison of three methods.

Method	Task	Pros	Cons
CV	Apply CCN (naming and caching) to IVC	Tight integration of CCN with geographic routing	Cache miss issue
ECV	Prioritize response over request	Avoid cache miss	Extra overhead, and a slight increase in delay
ECV+	Control cache probability by using channel usage rate	Share buffer of adjacent vehicles	Limited performance when the number of vehicles is small

#### 4. Simulation Evaluation

Simulation evaluation is performed to evaluate the effectiveness of the proposed method. The number of hops (the average number of transmissions per data packet), and the content acquisition success rate per interest packet, are used as main performance metrics.

##### 4.1. Comparison Methods

Four methods, Non-Cache, CV, ECV, and ECV+, will be compared. Non-Cache does not use the cache mechanism of CCN. CV (Section 3.1) is the baseline method that realizes the CCN function for IVC. ECV (Section 3.2) is an extension of CV and solves the cache miss problem. ECV+ (Section 3.3) includes all the proposed functions and improves cache efficiency by monitoring the channel usage rate.

##### 4.2. Simulation Conditions

In the simulation, the actual map data obtained from the open street map is read into the network simulator Scenargie [20], and the shielding of the building is emulated. The area used for the simulation is the Ginza area in Tokyo (Figure 5), which represents the urban area of Japan. It is assumed that each vehicle is interested in road/traffic information along its trajectory in order for smooth travel. Therefore, each vehicle queries for forward intersections (for the purpose of determining whether it will pass that intersection or take an alternative path if congestion will occur there). All vehicles issue Interest packets, and the target intersection of Interest packet is specified based on the popularity of intersections set according to the types of roads that make up the intersection.

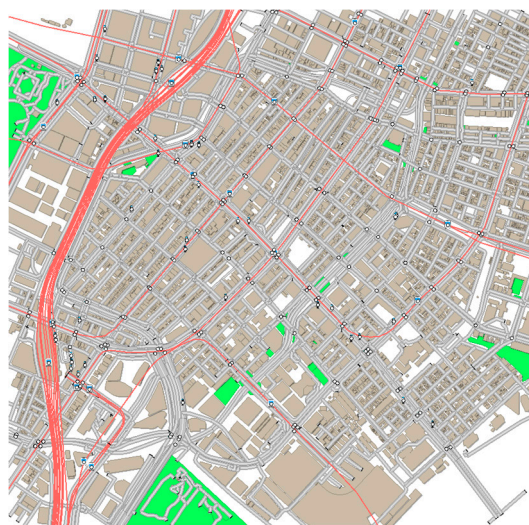
**Figure 5.** Simulation area (Ginza in Tokyo).

Table 4 shows a detailed simulation setting. The moving speed of vehicles ranges from 20 km/h to 30 km/h. Traffic lights at intersections change periodically (period is 90 s). The information lifetime is 30 s.

**Table 4.** Simulation setting.

Item	Value
Network simulator	Scenargie [20]
Field	1600 m × 1600 m
Area	Ginza
Vehicle initial position	random
Vehicle speed	20 km/h~30 km/h
Total number of vehicles	300–700 units
Cache update	Least Recently Used (LRU)
Communication method	IEEE802.11p (5.9 GHz)
Transmission rate	3 Mbps
Packet size	Interest 256 bytes Data 512 bytes
Radio propagation model	Nakagami (with fading)
Information generation interval for each vehicle	1 s–16 s
Simulation execution time	300 s
Number of executions	720 times
Buffer size	1 GB–10,000 GB

### 4.3. Simulation Results

Here, we show the simulation results using the cache buffer capacity, information generation interval, and vehicle density as parameters.

#### 4.3.1. Results under Different Cache Buffer Capacity

When cache buffer capacity of each vehicle is changed from 1 GB, 10 GB, 100 GB, 1000 GB, to 10,000 GB, Figure 6 shows the variation of average number of hops, content acquisition success rate, cache hit rate, and cache probability, respectively. In addition, 95% confidence interval is also shown in each figure.

As the cache buffer capacity increases, the average number of hops decreases (Figure 6a), while content acquisition success rate increases (Figure 6b). This is because more contents can be stored in the cache buffer and the cache hit rate improves as the cache buffer capacity increases (Figure 6c).

The average number of hops of the three methods using CCN is almost equal at 1 GB. This is because only a few contents can be cached and cache hit rate is low. The average hop count decreases as the cache buffer capacity increases to 10 GB and 100 GB, but approaches steady values when the cache buffer capacity is increased from 1000 GB to 10,000 GB because, with 1000 GB, most contents can be cached. In the vicinity of 100 GB, the difference among different methods in the average number of hops is large: 13.4 for Non-Cache, 7.1 for CV, 4.9 for ECV, and 2.3 for ECV+. ECV+, ECV, and CV, respectively, reduce the average number of hops by up to 83%, 63%, and 47%, compared with Non-Cache.

With the increase of cache buffer capacity, cache hit rate improves and the average number of hops decreases. Accordingly, the probability of packet loss decreases, which leads to a higher content acquisition success rate. When cache buffer capacity is 100 GB, ECV+ improves the content acquisition success rate by about 45% compared with ECV.

The results in Figure 6d show that the cache probability in ECV+ tends to increase with cache buffer capacity. When the cache buffer capacity is 100 GB, the average cache probability is 36%, which means nearly three times of contents can be cached compared with ECV. As a result, ECV+ improves cache hit rate by about 68% compared with ECV. The cache hit rate of ECV+ is very high

at 1000 GB (Figure 6c), and is almost 100% at 10,000 GB because the actual message volume is about 5000 GB.

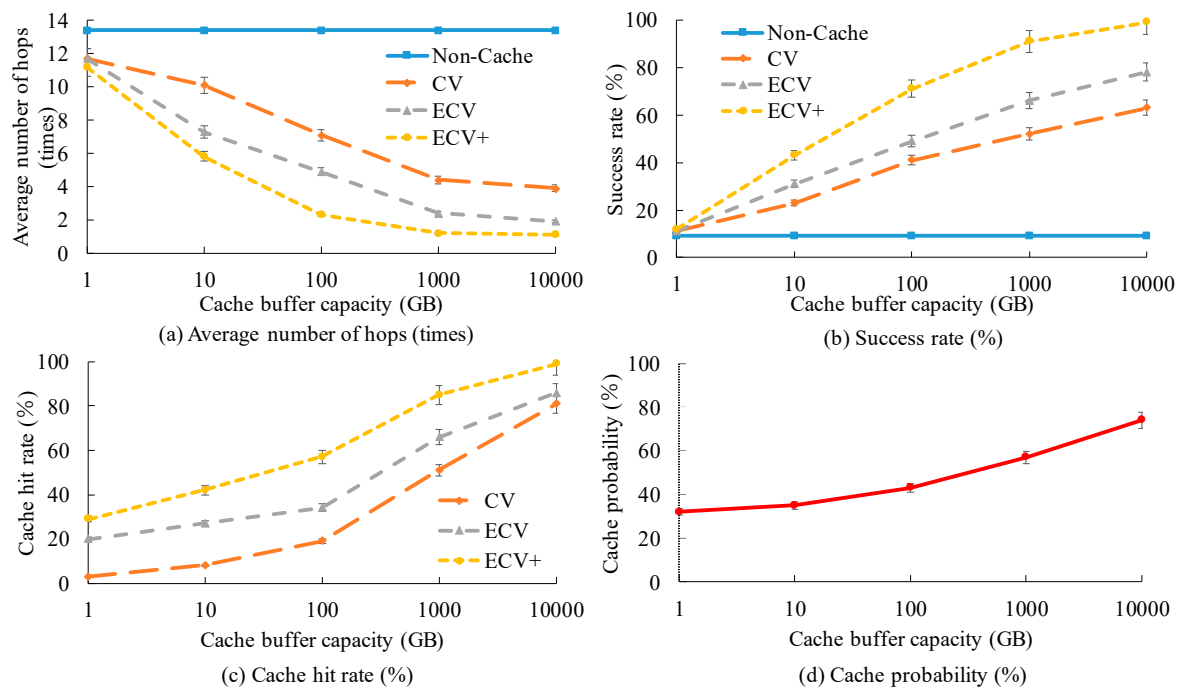


Figure 6. Results under different cache buffer capacity (information generation interval is 4 s and the number of vehicles is 500).

#### 4.3.2. Results under Different Information Generation Intervals

When the information generation interval is changed from 1, 2, 4, 8, to 16 seconds, Figure 7 shows average number of hops, content acquisition success rate, cache hit rate, and cache probability, respectively.

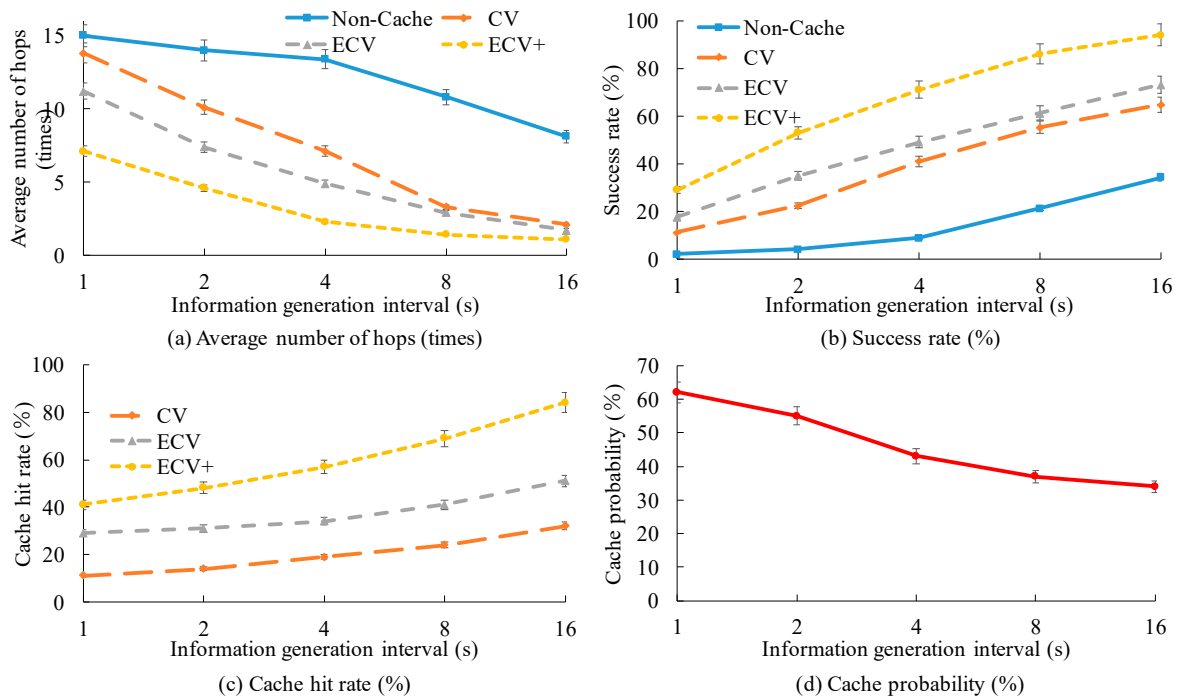
As the information generation interval increases, channel usage rate decreases, and cache hit rate increases (Figure 7c). Accordingly, the average number of hops decreases (Figure 7a) while content acquisition success rate increases (Figure 7b). When the information generation interval is large (16 s), most contents can be cached, and the performance difference among ECV+, ECV, and CV for the average number of hops is not large. However, when the information generation interval is small (1 s and 2 s), many messages lead to channel congestion. Under these severe circumstances, the average number of hops in all methods gets large, but ECV+ helps to greatly reduce the average number of hops, and achieves a much higher content acquisition success rate, compared with other methods.

Figure 7d shows that cache probability decreases as the information generation interval increases. When the information generation interval is small, too much information leads to a high channel usage rate. Thus, the cache probability is increased to suppress the channel usage rate.

#### 4.3.3. Results under Different Vehicle Density

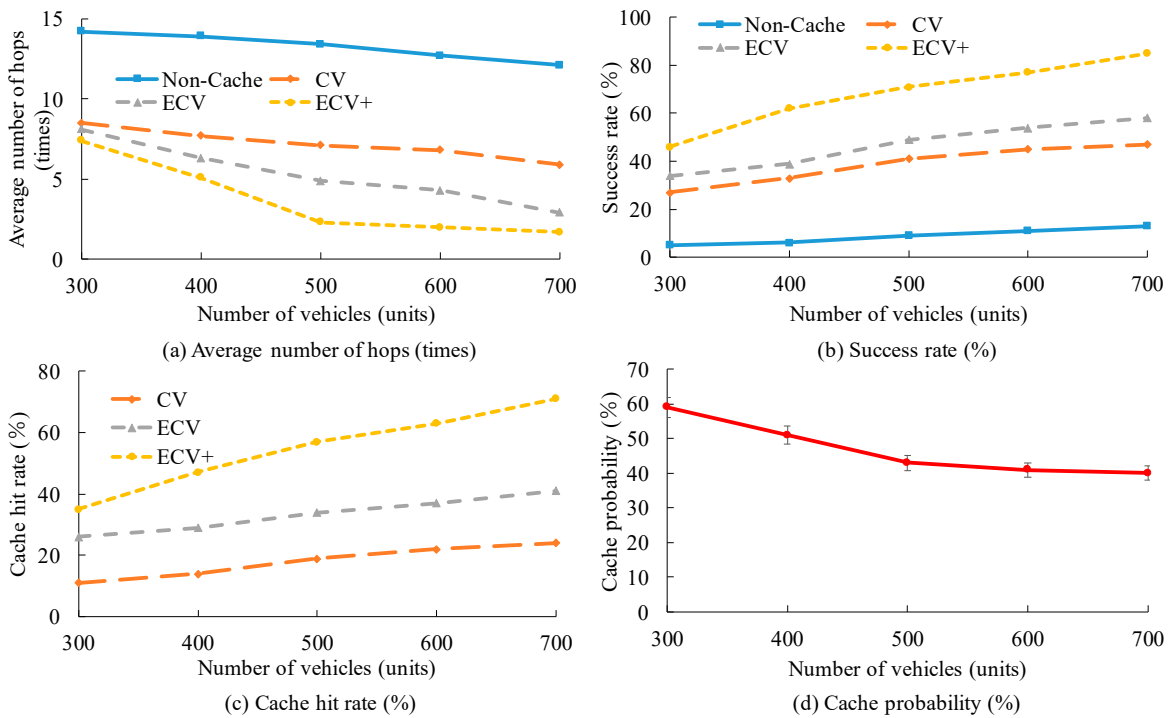
When vehicle density changes, Figure 8 shows the variation in the average number of hops, content acquisition success rate, cache hit rate, and cache probability, respectively.

As the vehicle density increases, the average number of hops decreases a little (Figure 8a). This is because more vehicles help improve network connectivity. Accordingly, the content acquisition success rate increases (Figure 8b). ECV+ further exploits more vehicles to cache more contents (cache hit rate increases with vehicle density in Figure 8c), and achieves much higher performance than other methods.



**Figure 7.** Results under different information generation intervals (cache buffer capacity is 100 GB and the number of vehicles is 500).

Figure 8d shows that cache probability decreases as the vehicle density increases. This is because ECV+ tries to exploit vehicle buffer to cache more contents, and, accordingly, the amount of content to be cached per vehicle decreases.



**Figure 8.** Results under different number of vehicles (cache buffer capacity is 100 GB and information generation interval is 4 s).

#### 4.4. Discussion

Of the three parameters (cache buffer capacity, information generation interval, and vehicle density), the parameter that most affects the results is cache buffer capacity. With plain CCN (cache function), cache function helps to reduce the average number of hops and improve the success rate of dissemination. When cache buffer capacity is too small (frequent update of cache buffer leads to low hit rate) or too large (most contents can be cached), the superiority of ECV+ is not very obvious. However, with a moderate cache buffer capacity, ECV+ far outperforms other methods, and achieves better cost performance. The cache buffer capacity of 10,000 GB is not realistic at this stage yet. However, advances in storage technology surely will reduce the cost of cache memory. Meanwhile more IVC messages are expected to be generated in the era of self-driving. Therefore, a large cache buffer capacity will improve the performance of the proposed method when it becomes practical.

Information generation interval affects channel usage rate. The information generation interval of several seconds is used to evaluate the proposed method in severe conditions. Due to the nature of Pull-type delivery, information with low urgency is delivered by request, so the information generation interval is expected to increase in real scenarios, where the proposed method is expected to work well too.

Vehicle density has less impact on the results than the other parameters. One potential reason is that vehicle density is not explicitly taken into account in adjusting cache probability. However, a big change can be seen when the number of vehicles gets larger than a threshold. On the other hand, if the number of vehicles is too small (which affects the connectivity), the result will be greatly deteriorated.

From the above discussions, the proposed ECV+ method improves performance more or less in different scenarios. If the cache probability can be controlled in a better way by machine learning (e.g., reinforcement learning), it is expected that the superiority of the proposed method will be more obvious.

## 5. Conclusions

We have proposed to apply CCN for efficient collection of road and traffic information via IVC. The importance of unique naming and topology adaptive routing has been verified before. By further including the dynamic adjustment of cache probability, the proposed ECV+ method greatly reduces the average number of hops by up to 83% and improves content acquisition success rate by up to 689% compared with the method without a cache mechanism. The effect of the proposed method strongly depends on the cache buffer capacity of each vehicle. In order to maximize the effectiveness of the proposed method with limited cache buffer, cache probability is adjusted, based on the channel usage rate.

In the future, in order to cope with more complicated vehicle-to-vehicle communication, we aim to apply reinforcement learning to adjust cache probability. In addition, since delivery by multi-hop communication depends on the vehicle topology, experiments in other urban areas and on highways will also be conducted.

**Author Contributions:** Conceptualization, S.T.; methodology, T.N. and S.T.; validation, T.N.; writing—original draft preparation, T.N.; writing—review and editing, S.T. and S.O.; supervision, S.T.; project administration, S.T.; funding acquisition, S.T. and S.O. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by JSPS KAKENHI Grant Number 19K11937.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Mori, T.; Tang, S.; Obana, S. Fine-grained integration of priority control and relay selection for fast and reliable inter-vehicle communication. In *Third International Congress on Information and Communication Technology*; Springer: London, UK, 2018.

2. Williams, B.; Camp, T. Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks. In Proceedings of the 3rd ACM International Symposium on Mobile ad Hoc Networking & Computing (MobiHoc'02), Lausanne, Switzerland, 9–11 June 2002; pp. 194–205.
3. Panichpapiboon, S.; Pattara-Atikom, W. A Review of Information Dissemination Protocols for Vehicular Ad Hoc Networks. *IEEE Commun. Surv. Tutor.* **2011**, *14*, 784–798. [[CrossRef](#)]
4. Tahmasbi-Sarvestani, A.; Fallah, Y.P.; Kulathumani, V. Network-aware double-layer distance-dependent broadcast protocol for VANETs. *IEEE Trans. Veh. Technol.* **2015**, *64*, 5536–5546. [[CrossRef](#)]
5. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plasee, M.F.; Briggs, N.; Braynard, R. Networking named content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (ACM CoNEXT'09), Rome, Italy, 1–4 December 2009; pp. 1–12.
6. Louiza, M.; Samira, M. A New Framework for Request-driven Data Harvesting in Vehicular Sensor Networks. *Netw. Protoc. Algorithms* **2013**, *5*, 1–18. [[CrossRef](#)]
7. Yonemura, A.; Hasegawa, T.; Obana, S. Proposal of In-Network Caching System with Dynamic Content Pre-fetching Mechanism. In Proceedings of the International Conference of Mobile Computing and Ubiquitous Networking (ICMU'15), Hakodate, Japan, 20–22 January 2015; pp. 100–101.
8. Nakazawa, T.; Obana, S.; Tang, S. Efficient Collection of Road and Traffic Information by CCN-based Inter-Vehicle Communications, In Proceedings of the 2018 24th Asia-Pacific Conference on Communications (APCC'18). Ningbo, China, 12–14 November 2018.
9. Paik, E.; Mahadevan, P.; Jang, M.; Cho, E. Benefits and Research Challenges of Content-Centric Networking. In *Internet-Draft-Paik-Icn-Challenges-00*; IETF: Fremont, CA, USA, 2013.
10. Xylomenos, G.; Ververidis, C.N.; Siris, V.A. A Survey of Information-Centric Networking Research. *IEEE Commun. Surv.* **2013**, *16*, 1024–1049. [[CrossRef](#)]
11. Amadeo, M.; Campolo, C.; Molinaro, A. Information-Centric Networking for Connected Vehicles: A Survey and Future Perspectives. *IEEE Commun. Mag.* **2016**, *54*, 98–104. [[CrossRef](#)]
12. Adhatarao, S.S.; Chen, J.; Arumathurai, M.; Fu, X.; Ramakrishnan, K.K. Comparison of naming schema in ICN. In Proceedings of the 2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN'16), Rome, Italy, 13–15 June 2016; pp. 1–6.
13. Yu, Y.-T.; Li, X.; Gerla, M.; Sanadidi, M. Scalable VANET Content Routing Using Hierarchical Bloom Filters. *Wirel. Commun. Mob. Comput.* **2015**, *15*, 1001–1014. [[CrossRef](#)]
14. Liu, L.C.; Xie, D.; Wang, S.; Zhang, Z. CCN-based Cooperative Caching in VANET. In Proceedings of the International Conference on Connected Vehicles and Expo (ICCVE'15), Shenzhen, China, 19–23 October 2015; pp. 198–203.
15. Psaras, I.; Chai, W.K.; Pavlou, G. Probabilistic In-Network Caching for Information-Centric Networks. In Proceedings of the Second Edition of the ICN Workshop on Information-Centric Networking (ICN'12), Helsinki, Finland, 13–17 August 2012; pp. 55–60.
16. Meng, Y.; Naeem, M.A.; Ali, R.; Zikria, Y.B.; Kim, S.W. DCS: Distributed Caching Strategy at the Edge of Vehicular Sensor Networks in Information-Centric Networking. *Sensors* **2019**, *19*, 4407. [[CrossRef](#)] [[PubMed](#)]
17. Yan, H.; Gao, D.; Su, W.; Foh, C.H.; Zhang, H.; Vasilakos, A.V. Caching strategy based on hierarchical cluster for named data networking. *IEEE Access* **2017**, *5*, 8433–8443. [[CrossRef](#)]
18. Zhao, W.; Qin, Y.; Gao, D.; Foh, C.H.; Chao, H.C. An Efficient Cache Strategy in Information Centric Networking Vehicle-to-Vehicle Scenario. *IEEE Access* **2017**, *5*, 12657–12667. [[CrossRef](#)]
19. *700 MHz Band Intelligent Transport Systems*; ARIB Standards STD-T109, Version 1.3-E1; Association of Radio Industries and Businesses: Tokyo, Japan, 2017.
20. Takai, M.; Martin, J.; Kaneda, S.; Maeno, T. Scenargie as a Network Simulator and Beyond. *J. Inf. Process.* **2019**, *27*, 2–9. [[CrossRef](#)]

