Cooperative Content Precaching Scheme Based on the Mobility Information of Vehicles in Intermittently Connected Vehicular Networks

Youngju Nam 1, Jaejeong Bang 1, Hyunseok Choi 1, Yongje Shin 2 and Euisin Lee 1,*

1 School of Information and Communication Engineering, Chungbuk National University, Cheongju 28644, Korea
2 Research Institute for Computer and Information Communication, Chungbuk National University, Cheongju 28644, Korea
* Correspondence: eslee@chungbuk.ac.kr

Abstract: Intermittently connected vehicular networks (ICVNs) consist of vehicles moving on roads and stationary roadside units (RSUs) deployed along roads. In ICVNs, the long distances between RSUs and the large volume of vehicular content lead to long download delays to vehicles and high traffic overhead on backhaul links. Fortunately, the improved content storage size and the enhanced vehicular mobility prediction afford opportunities to ameliorate these problems by proactively caching (i.e., precaching) content. However, existing precaching schemes exploits RSUs and vehicles individually for content precaching, even though the cooperative precaching between them can reduce download delays and backhaul link traffic. Thus, this paper proposes a cooperative content precaching scheme that exploits the precaching ability of both vehicles and RSUs to enhance the performance of content downloads in ICVNs. Based on the trajectory and velocity information of vehicles, we first select the optimal relaying vehicle and the next RSUs to cache the requested content proactively and provide it to the requester vehicle optimally. Next, we calculate the optimal content precaching amount for each of the relaying vehicle and the downloading RSUs by using a mathematical model that exploits both the dwell time in an RSU and the contact time between vehicles. To compensate for the error of the mobility prediction in determining both the dwell time and the contact time, our scheme adds a guardband to the optimal content precaching amount by considering the expected reduced delay. Finally, we evaluate the proposed scheme in various simulation environments to prove the achievement of efficient content download performance by comparing with the existing schemes.

Keywords: vehicular networks; content; precaching; mobility

1. Introduction

With the rapid development of wireless communications and vehicular technologies, vehicular networks have become a fundamental building block for designing the architecture of intelligent transportation systems (ITS) [1–7]. Considering the various communication scenarios in ITS, vehicular networks help in providing drivers and passengers safety and convenience, and furthermore introduce next-generation applications such as multimedia entertainment and social interactions [8–10]. Generally, vehicular networks consist of vehicles moving on roads and stationary roadside units (RSUs) deployed along roads. Vehicles can communicate with each other through vehicle-to-vehicle (V2V) communication and connect to RSUs via vehicle-to-infrastructure (V2I) communications [11–16]. RSUs, which are one of the infrastructures in vehicular networks, and play a role in providing the functions of both routers for relaying content data and access points for connecting to content servers on the Internet [17,18].
With the ever-increasing scale of vehicular content, such as multimedia content of images, music, and video clips, the communication capacity of vehicular networks might not be able to completely deliver the entire content from a single RSU to vehicles within its communication coverage for V2I communication owing to their limited bandwidth [19,20]. To overcome this issue, vehicular networks are required to provide seamless connections to vehicles from multiple continuous RSUs communicated with backhaul links. However, deploying sufficient RSUs along roads may be difficult owing to several geographic constraints and high deployment costs [18]. Vehicular networks corresponding to this characteristic are defined as intermittently connected vehicular networks (ICVNs) [21,22], because vehicles can only intermittently connect to RSUs on roads during movement. When vehicles move in the area (called the outage zone) between two neighboring RSUs, they cannot download content from the RSUs [23]. Thus, long inter-RSU distances and large vehicular contents in ICVNs lead to long download delays to vehicles and high traffic overhead on backhaul links [24].

Fortunately, the increase in content storage size and the enhancement of vehicular mobility prediction afford opportunities to alleviate these problems by proactively caching (i.e., precaching) content [25]. To this end, research on content precaching based on mobility information has recently been conducted on ICVNs [20,23,26,27]. To reduce the delay of content download and the amount of traffic on backhaul links, content precaching protocols based on the mobility information of vehicles have been proposed using two approaches: RSU precaching and vehicle precaching. RSU precaching [20,26] exploits the content precaching of the next RSU on the predicted trajectory of a requester vehicle. Consequently, the requester vehicle can instantly download content from the next RSU when it enters the coverage of the next RSU. Vehicle precaching [23,27] exploits the content precaching of a vehicle in the coverage of the current RSU where a requester vehicle is currently located. Thus, when the requester vehicle meets the precaching vehicle in the outage zone of the current RSU coverage, the precaching vehicle relays the content to the requester vehicle. However, the existing RSU precaching protocols led to heavy backhaul link traffic because they focused only on the protocol process for the operation of content precaching [20] or used only simple information such as the locations of vehicles and RSUs for the calculation of the precaching amount [26] and thus, did not provide the optimal precaching amount. Additionally, the existing vehicle precaching protocols caused less efficient content precaching owing to the exploitation of half of the RSU coverage for downloading to a precaching vehicle [23] or for selecting a secure precaching vehicle to support secure content delivery [27]. Furthermore, the existing precaching protocols only exploited RSUs and vehicles individually for content precaching, even though cooperative precaching through both RSUs and vehicles could reduce content download delays and backhaul link traffic.

Therefore, this paper proposes a cooperative content precaching scheme that enhances the performance of content downloads in ICVNs. In contrast to existing schemes, the proposed scheme cooperatively exploits the precaching ability of both vehicles and RSUs to achieve optimal content precaching between them. Based on the mobility information of vehicles, our scheme first selects both the optimal vehicle for relaying content to a requester vehicle in the outage zone and the next RSU for downloading content to the requester vehicle in the coverage of the next RSU. Next, this scheme calculates the optimal content precaching amount for each optimal relaying vehicle, and downloads the next RSU using our mathematical model. The mathematical model uses both the dwell time of vehicles in an RSU and the contact time between vehicles in the outage zone. Subsequently, the mobility prediction of vehicles is used to calculate the dwell and contact times. To compensate for the error of the mobility prediction in determining both the dwell and contact times, our scheme adds a guardband to the optimal content precaching amount to prevent the degradation of the content download while minimizing the amount of the guardband. For calculating the guardband amount, our scheme considers the expected reduced delay value that is obtained by the theoretical delay value and the actual estimated delay value.
Simulation results conducted in various environments verify that the proposed scheme achieves better performance than the existing RSU and vehicle precaching schemes in terms of content download delay, backbone traffic, precaching traffic, RSU usage, and hit ratio.

The remainder of this paper is organized as follows. Section 2 presents the network model and an overview of the proposed scheme. Section 3 describes in detail the proposed scheme related to cooperative content precaching based on the mobility information of vehicles. Section 4 addresses the guardband for handling errors in vehicle mobility prediction. Section 5 evaluates the performance of the proposed scheme by comparing it with existing schemes through simulation results. Finally, Section 7 concludes the paper.

2. Network Model and Overview

In this paper, we propose a cooperative content-precaching scheme for ICVNs. Figure 1 shows the network model of the proposed scheme. We consider a network in which a large number of vehicles travel on roads in vehicular network fields where several RSUs are deployed [12,17]. In this network, every vehicle moves along its travel route to arrive at its destination by passing several RSUs. In our scheme, whenever a vehicle enters the communication coverage of an RSU, it registers its current location and mobility information with the RSU by periodically sending a beacon message to the RSU [28,29]. Therefore, all RSUs have a table to manage information about the mobility of all vehicles within their communication coverage. In the proposed scheme, each RSU can select the optimal relaying vehicle and the next downloading RSU to provide the requested content to the requester vehicle by precaching the content based on information about the mobility of the vehicle within its own coverage. RSUs arranged at regular intervals are connected to each other and to a content server that can provide all the requested content through wired backhaul links [17]. However, because RSUs cannot be deployed to cover an entire area of roads owing to several geographic constraints and the high cost of deployment, there are bound to be outage zones between two neighboring RSUs in ICVNs [30], as shown in Figure 1. In the outage zone, the requested content cannot be provided to the requester vehicle from any RSUs. We assume that the downlink transmission rate from an RSU to each vehicle in its coverage area is the same, and each vehicle has a different velocity and travels toward its destination on the road. All vehicles can connect with RSUs through cellular communications, as well as with other vehicles through IEEE 802.11p wireless access in vehicular environments (WAVE) communications [31,32]. In general, the communication coverage from an RSU to a vehicle (I2V) is longer than that between vehicles (V2V) because the transmission power and rate of I2V is higher than that of V2V.
If a vehicle requires to download the intended content while moving in the communication coverage of the RSU where the vehicle is located, the vehicle sends an interest packet to the RSU to request the intended content. In this paper, we named a vehicle that sends an interest packet to an RSU to download the intended content as a requester vehicle. Upon receiving the packet showing interest, the RSU requests and receives the intended content to and from the corresponding content server. After receiving the intended content, the RSU determines whether it can download the entire intended content to the requester vehicle. For this purpose, the proposed scheme calculates the downloadable amount of content for a requester vehicle in the communication coverage of the RSU where the requester vehicle is located. Then, the downloadable amount of content is calculated using the mobility information of the requester vehicle and the V2I transmission rate of the RSU. If the RSU can download the entire content to the requester vehicle, it sends the content to the requester vehicle. However, if the RSU cannot download the entire content to the requester vehicle, it selects the optimal relaying vehicle among vehicles except the requester vehicle in its communication coverage to precache the intended content for the requester vehicle. In the proposed scheme, the optimal relaying vehicle is determined by calculating both the downloadable time and amount from the RSU, and the relaying time and amount between the requester vehicle and each relaying vehicle. The RSU then precaches the optimal relaying amount of the intended content to the optimal relaying vehicle. Thus, the requester vehicle downloads the intended content from the RSU up to the amount that the RSU is enabled for sending. After exiting the communication coverage of the RSU, the requester vehicle receives the optimal relaying amount of intended content from the optimal relaying vehicle in the outage zone. Although the requester vehicle cannot fully download the entire size of the intended content from both the RSU and the optimal relaying vehicle, the RSU precaches the remaining amount of the intended content to the next RSU on the predicted trajectory of the requester vehicle through backhaul links. Thus, when the requester vehicle enters the next RSU, it can immediately download the remaining amount of intended content from the next RSU. If the next RSU cannot download the remaining amount of the intended content to the requester vehicle in its communication coverage, the aforementioned process continues until the downloading of the intended content is completed. Figure 1 shows an overview of the proposed scheme to describe the cooperative content precaching process based on both the optimal relaying vehicle and the next downloading RSU. The proposed scheme is described in detail in Section 3.

3. The Selection of the Optimal Relaying Vehicle

In this section, we describe the precaching scheme by selecting the optimal relaying vehicle and next RSU to provide as much content as possible to the requester vehicle. We first explain the decision to use precaching based on the downloadable amount of content in Section 3.1. Then, we select the optimal relaying vehicle based on the connectivity prediction in Section 3.2. Finally, in Section 3.3, we explain how the next RSU precaches the content to provide it to the requester vehicle. Table 1 shows notations for selecting the optimal relaying vehicle.

Table 1. Notations for selecting the optimal relaying vehicle.

| Notation | Description |
|----------|-------------|
| RSU<sub>j</sub> | j-th roadside unit (RSU), where j ∈ {1, · · · , J} |
| V<sub>i</sub> | i-th vehicle, where i ∈ {1, · · · , I} |
| V<sub>req</sub> | The requester vehicle |
| V<sub>rel</sub> | The selected optimal relaying vehicle |
| t<sub>max,i</sub> | The maximum time that V<sub>i</sub> can relay the content to V<sub>req</sub> within the outage zone |
| t<sub>D,i</sub> | The downloadable time for providing the requested content to V<sub>req</sub> within the coverage of RSU<sub>j</sub> |
| T<sub>D,i</sub> | The downloadable time for precaching the requested content to V<sub>i</sub> within the coverage of RSU<sub>j</sub> |
| T<sub>R,i</sub> | The relaying time of V<sub>i</sub> for providing the precached content to V<sub>req</sub> within the outage zone |
| s<sub>EX,i</sub> | The coverage diameter of RSU<sub>j</sub> |
3.1. Checking the Necessity for Precaching

If a vehicle traveling on the road within the coverage area of RSU\(_j\) wants to download the content, it sends a request message to RSU\(_j\), where \( j = \{1, \cdots, J\}\). When RSU\(_j\) receives the request message from the requester vehicle \( V_{req}\), it calculates the downloadable time \( t_{D_{req}}\) that \( V_{req}\) can download the content from the RSU\(_j\)'s coverage. \( t_{D_{req}}\) can be denoted as the time at which \( V_{req}\) is expected to leave RSU\(_j\)'s coverage, and it can be calculated based on the velocity and the location of \( V_{req}\) as follows:

\[
 t_{D_{req}} = \frac{x_{EX_i} - x_{req}}{v_{req}}, \tag{1}
\]

where \( x_{EX_i} \) is the communication coverage of the RSU\(_j\), \( x_{req} \) is the location of \( V_{req}\), and \( v_{req} \) is the velocity of \( V_{req}\), respectively. Based on \( t_{D_{req}}\), RSU\(_j\) calculates the downloadable amount \( C_R \) of the requested content that can be provided to the \( V_{req}\) within RSU\(_j\)'s coverage by using the transmission rate of the I2V communication, as follows:

\[
 C_R = t_{D_{req}} \times R_{I2V}, \tag{2}
\]

where \( R_{I2V}\) is the transmission rate of the I2V based on IEEE 802.11p WAVE [31].

RSU\(_j\) then decides whether the requested content has to be precached by comparing \( C_R \) with \( C_T \). If \( C_T \leq C_R \), it denotes that RSU\(_j\) can provide all of the requested content to \( V_{req}\) before \( V_{req}\) leaves the communication coverage of RSU\(_j\), where \( C_T \) is the size of the requested content by \( V_{req}\). In this case, RSU\(_j\) immediately provides the requested content to \( V_{req}\). Otherwise, it denotes that RSU\(_j\) does not have sufficient time to provide all of the requested content to \( V_{req}\) at time \( t_{D_{req}}\). Thus, RSU\(_j\) should select a relaying vehicle among the vehicles within its coverage to provide the requested content to \( V_{req}\) after \( V_{req}\) leaves the coverage of RSU\(_j\). The relaying vehicle, \( V_{rel}\), provides the remaining amount of content to \( V_{req}\) outside the communication coverage of RSU\(_j\).

3.2. Selection of the Optimal Relaying Vehicle

In this section, we describe the selection of the optimal relaying vehicle to provide the remaining requested content to \( V_{req}\). It is possible that RSU\(_j\) and \( V_{rel}\) cannot provide all the requested content to \( V_{req}\) owing to the large size of the content or the short connectivity. In this case, to provide the remaining amount of content to \( V_{req}\) outside the coverage, RSU\(_j\) should select the relaying vehicle while providing as much requested content as possible to \( V_{req}\) within its coverage. To determine the optimal relaying vehicle \( V_{rel}\), RSU\(_j\) calculates the
amount of the content that can be relayed by each candidate vehicle that is managed by the table of RSU_j within its coverage. Then, RSU_j selects the optimal relaying vehicle among the candidate vehicles and the amount C_V of the requested content that can be relayed by it.

For this selection through the table, RSU_j uses the beacon messages that include the mobility information of vehicles that remain in the coverage of RSU_j. Based on the location and velocity of each candidate vehicle, RSU_j calculates each candidate vehicle’s downloadable time and the downloadable amount of the content from RSU_j, as shown in Section 3.2.1. It also calculates the relaying time and the relaying amount of the content, which can be relayed to V_req from each candidate vehicle, as detailed in Section 3.2.2. In Section 3.2.3, using these values, RSU_j selects the optimal relaying vehicle as V_rel and its relaying amount as C_V. Therefore, RSU_j provides the C_V of the content to V_rel within its coverage and V_req relays it to V_req when V_req leaves RSU_j’s coverage.

3.2.1. The Downloadable Time and Amount

To determine the optimal relaying vehicle V_rel, RSU_j first has to know the downloadable time and amount of the content of each candidate vehicle V_i within RSU_j’s coverage. Similarly, to obtain T_{D_{req}}’, the downloadable time of each candidate vehicle can be calculated based on x_{EX}. However, when V_req leaves the coverage of RSU_j, the content must be relayed to V_req from the relaying vehicle. Therefore, RSU_j can obtain the downloadable time of each candidate vehicle from RSU_j in order to provide it to V_req, as follows:

$$T_{D_i} = \min \left( \frac{x_{EX_i} - x_i}{v_i}, T_{D_{req}} \right),$$

where x_i and v_i are the position and speed of V_i, respectively. Then, RSU_j calculates the downloadable amount C_{DV_i} of the content that the V_i can precache from RSU_j during the time T_{D_i} based on the transmission rate of I2V as follows:

$$C_{DV_i} = T_{D_i} \times R_{I2V},$$

where R_{I2V} is the V2I transmission rate that is based on IEEE 802.11p WAVE [31]. Even if V_i can precache the C_{DV_i} of the content, it may not relay it to V_req because of the short connectivity between V_req and V_rel. Hence, the relaying time and amount of content must be calculated.

3.2.2. The Relaying Time and Amount

It is also necessary to calculate the relaying time and relaying amount of the requested content of the candidate vehicles in the outage zone to select the optimal relaying vehicle. To determine the relaying time, RSU_j must first calculate the connection time between V_req and each V_i. However, when V_req enters the next RSU, that is, RSU_{j+1}’s coverage, the download from the RSU_{j+1} is better than that from V_i, owing to the difference in the power and the transmission performances. Therefore, the relaying should be conducted within the outage zone, which is the area between the coverages of RSU_j and RSU_{j+1}. Then, the relaying time T_{R_i} of V_i can be denoted based on the location and the velocity of both V_req and V_i, as follows:

$$T_{R_i} = \begin{cases} 
\frac{r_o + x_i - x_{req}}{v_{req} - v_i} - T_{D_i}, & \text{if } v_{req} > v_i \\
\frac{r_o - x_i + x_{req}}{v_{req} - v_i} - T_{D_i}, & \text{if } v_{req} < v_i \\
l_{max_i}, & \text{if } v_{req} = v_i 
\end{cases}$$

where r_o is the communication range of the vehicles, x_i is the location of V_i, and v_i is the velocity of V_i. T_{D_i} is the downloadable time of V_i. l_{max_i} is the maximum time that V_i can
relay the content to $V_{req}$ within the outage zone. However, if the relaying time $T_{R_i}$ of $V_i$ is larger than $t_{max,j}$, then $T_{R_i}$ is limited to $t_{max,j}$. $t_{max,j}$ is denoted as follows:

$$t_{max,j} = \frac{U}{v_{req}} - T_{D_j},$$

(6)

where $U$ is the distance of the outage zone between the $RSU_i$ and $RSU_{(j+1)}$. This indicates that $V_{req}$ can receive the content from the selected $V_i$ within the outage zone and from $RSU_{j+1}$ after passing through the outage zone. Using $T_{R_i}, RSU_j$ can calculate the relaying amount $C_{RV_i}$ of the content that can be provided to $V_{req}$ from $V_i$ within the outage zone, as follows:

$$C_{RV_i} = T_{R_i} \times R_{V2V},$$

(7)

where $R_{V2V}$ is the transmission rate of the V2V communications that is based on IEEE 802.11p WAVE [31]. $RSU_i$ can select the optimal relaying vehicle $V_{rel}$ because it now knows the amount of the content that can be downloaded to $V_i$ from $RSU_j$ and can be relayed to $V_{req}$ from $V_i$.

3.2.3. Selection of the Optimal Relaying Vehicle

To select the optimal relaying vehicle, $RSU_j$ creates each set of $V_i$ using the downloadable amount $C_{DV_i}$ and the relaying amount $C_{RV_i}$ of the content as follows:

$$Set(V_i) = \{C_{DV_i}, C_{RV_i}\}.$$  

(8)

The set $Set(V_i)$ is saved in the $RSU_j$’s table with the mobility information of $V_i$. Based on $Set(V_i)$, $RSU_j$ compares the $C_{DV_i}$ and $C_{RV_i}$ of each $V_i$ and selects a smaller value. In the case of $C_{DV_i} > C_{RV_i}$, it denotes that the connection time between $V_{req}$ and $V_i$ is insufficient to relay all the precached content that $V_i$ downloads from $RSU_j$. Otherwise, in the case of $C_{DV_i} < C_{RV_i}$, it denotes that the downloadable time of $V_i$ is not sufficient to compare the connection time that $V_i$ can provide the content to $V_{req}$. Therefore, $RSU_j$ selects the minimum value between the two values as the practical delivery amount $C_{PDV_i}$ of the content, as follows:

$$C_{PDV_i} = \min\{C_{DV_i}, C_{RV_i}\}.$$  

(9)

Thus, $C_{PDV_i}$ denotes that the maximum amount of the content that $V_i$ can practically provide to $V_{req}$ before $V_{req}$ reaches the coverage of $RSU_{j+1}$. Then, $RSU_j$ can determine the optimal relaying amount $C_V$ by selecting the optimal relaying vehicle that has the largest $C_{PDV_i}$, as follows:

$$C_V = C_{PDV_{vrel}} = \max\{C_{PDV_1}, \ldots, C_{PDV_{n}}\}.$$  

(10)

The $C_{PDV_{vrel}}$ of $V_{rel}$ denotes the maximum amount of the content that $V_{req}$ can receive from $V_{rel}$ within the outage zone.

When $V_{req}$ leaves the coverage of $RSU_j$ and enters the outage zone, $V_{rel}$ immediately provides the $C_V$ of downloaded content to $V_{req}$. For example, as shown in Figure 1, if $V_{1rel}$ is selected as the optimal relaying vehicle for $V_{1req}$, because $V_{1req}$ wants to download the content, $RSU_j$ provides the $C_R$ of the requested content to $V_{1req}$ and the $C_V$ of the requested content to $V_{rel}$. Then, in the case of $V_{2rel}$, which is selected as the optimal relaying vehicle for $V_{2req}$ and downloaded $C_V$ of the content, it provides the content to $V_{2req}$ during the outage zone.

3.3. Optimal Content Precaching to Next RSU

If the $C_R + C_V$ is smaller than $C_T$, which is the total size of the requested content, $RSU_j$ judges that it is not sufficient for the content that can be provided before $V_{req}$ reaches $RSU_{j+1}$. In this case, $RSU_j$ sends a message to make the next RSU, that is, $RSU_{j+1}$ preaches
Cooperative Content Precaching

Algorithm 1 Cooperative Content Precaching

**Input:** the content $c$, the RSU $j$’s coverage $x_{E_{X_j}}$, the requester vehicle $V_{req}$’s location $x_{req}$ and the velocity $v_{req}$

**Output:** Action of the RSU $j$

1. $C_T$ ← (the size of $c$)
2. $C_R$ ← $C_R(x_{E_{X_j}}, x_{req}, v_{req})$
3. if $C_T \leq C_R$ then
4. [Providing $C_T$-sized $c$ to $V_{req}$]
5. else
6. $max = maxID = 0$
7. for $i \in 1, \ldots, n$ do
8. $C_{PDV_i} = \min\{C_{DV_i}, C_{RV_i}\}$
9. if $C_{PDV_i} > max$ then
10. $max = C_{PDV_i}$
11. $maxID = i$
12. end if
13. $V_{rel}$ ← $maxID$
14. $C_V = max$
15. [Providing $C_R$-sized $c$ to $V_{req}$]
16. [Precaching $C_V$-sized $c$ to $V_{rel}$]
17. if $C_T > C_R + C_V$ then
18. [Sending ($V_{req}$, $C_R + C_V$, $c$) information to the RSU $j + 1$]
19. end if
20. end for
21. end if

4. Guardband for Handling the Mobility Prediction Errors

To select the optimal relaying vehicle, the proposed scheme must calculate the downloadable time of a requester vehicle $V_{req}$ in an RSU and the relaying time between $V_{req}$ and the relaying vehicle $V_{rel}$ in the outage zone. The downloadable time is based on the dwell time of $V_{req}$ in the RSU and the relaying time is based on the contact time between $V_{req}$ and $V_{rel}$. However, accurately predicting both dwell and contact times is very difficult because vehicles change their speed irregularly. In this section, we describe a guardband to compensate for the mobility prediction error caused by the fact that accurately predicting...
dwell and contact times is not feasible. An error exists in predicting the connection time (the contact time between vehicles in an outage zone and the dwell time of a requester vehicle \(V_{req}\) in an RSU) based on their speed. Two mobility-prediction errors arise when the connection time is inaccurate. First, due to the shorter-than-expected download time in an RSU, \(V_{req}\) does not obtain the expected amount of precached content. Because the portion of the content that is not received is not precached in the next RSU, the next RSU must obtain it from the content server or the previous RSU and send it to \(V_{req}\). Obtaining content from the content server causes an additional delay. Second, the relay time between \(V_{req}\) and \(V_{rel}\) in the outage zone may deviate from the predicted value. If it is shorter than expected, the network performance will decrease for the same reason stated earlier.

If the relaying time is longer than expected, \(V_{rel}\) cannot provide extra content because it has only a precached portion of the content. In this instance, the delay increases owing to the increased time, which is required to provide the requested content. Therefore, a guardband against prediction errors of the contact time between vehicles can mitigate the corresponding problem. Table 2 shows notations for the calculation of the guardband.

In this study, there are two situations that predict the connection time: (1) in Section 4.1, the dwell time of \(V_{req}\) within the RSU communication range is calculated; (2) in Section 4.2, the contact time between \(V_{req}\) and \(V_{rel}\) outside the RSU range is determined. These two estimated connection times can be used to formulate the number of guardbands required, as shown in Section 4.3.

| Notation            | Description                                                                 |
|---------------------|-----------------------------------------------------------------------------|
| \(delay_{G,\varepsilon_{mob}}\) | The expected reduced delay value by \(G\) of guardband and \(\varepsilon_{mob}\) of mobility error ratio |
| \(t_{theo}\)        | The theoretical delay value in an ideal situation                           |
| \(t_{est}\)         | The actual estimated delay value that is obtained by the predicted situation |
| \(m_{theo}\)        | The theoretical distance value in an ideal situation                         |
| \(m_{est}\)         | The actual estimated distance value that is obtained by the predicted situation |
| \(m\)               | The transmission rate of backhaul links                                      |
| \(\varepsilon_{req}(m)\) | One or zero value whether RSU\(_{j}\) provides the requested content to \(V_{req}\) after \(V_{req}\) moves a distance of \(m\) meters |
| \(\delta_{req}(m)\)  | One or zero value whether \(V_{k}\) provides the precached content to \(V_{req}\) after \(V_{req}\) moves a distance of \(m\) meters |
| \(\varepsilon_{mob}\) | The mobility error ratio of the vehicles                                    |
| \(v_{req}(j)\)       | The expected speed based on \(v_{req}\) at the coverage of RSU\(_{j}\)       |
| \(v_{req,av}(j)\)    | The average expected speed based on \(v_{req}\) and \(\varepsilon_{mob}\) at the coverage of RSU\(_{j}\) |
| \(v_{out}(j)\)       | The average expected speed based on \(v_{req}\) and \(\varepsilon_{mob}\) at the outage zone between RSU\(_{j}\) and RSU\(_{j+1}\) |
| \(v_{out,av}(j)\)    | The average expected speed based on \(v_{req}\) and \(\varepsilon_{mob}\) at the outage zone between RSU\(_{j}\) and RSU\(_{j+1}\) |
| \(R_{req}(m)\)       | One or zero value whether RSU\(_{j}\) relays the requested content to \(V_{req}\) after \(V_{req}\) moves a distance of \(m\) meters |
| \(P_{req}(m)\)       | One or zero value whether \(V_{k}\) provides the precached content to \(V_{req}\) after \(V_{req}\) moves a distance of \(m\) meters |
| \(com_{req}(m)\)     | One or zero value whether \(V_{req}\) and RSU\(_{j}\) are connected after \(V_{req}\) moves a distance of \(m\) meters |
| \(out_{req}(m)\)     | One or zero value whether \(V_{req}\) stays within the outage zone between RSU\(_{j}\) and RSU\(_{j+1}\) after \(V_{req}\) moves a distance of \(m\) meters |
| \(traffic_{G,\varepsilon_{mob}}\) | The expected reduced traffic by \(G\) of guardband and \(\varepsilon_{mob}\) of mobility error ratio |
| \(traffic_{theo}\)   | The theoretical traffic value in ideal situation                            |
| \(traffic_{est}\)    | The actual estimated traffic value that is obtained by the predicted situation |

### 4.1. Dwell Time in an RSU

When \(V_{req}\) enters the coverage area of an RSU, its dwell time can be calculated using its position, velocity, and trajectory information. The proposed scheme then determines the amount of content \(V_{req}\) can download from the current RSU throughout its dwell time. Therefore, the proposed scheme allows the current RSU to precache the remaining portion of the requested content to the next RSU for \(V_{req}\). The remaining content is the total amount of content minus the amount of content downloaded during dwell time. However, if \(V_{req}\) increases its speed and, as a result, its dwell time decreases, the practical amount that can be downloaded is also reduced. Hence, when \(V_{req}\) connects to the next RSU, it should retrieve from the content server and transmit to \(V_{req}\) the portions of the content that
were not received by $V_{req}$ in the previous RSU. Because the data rate at which the content is brought from the content server and delivered to $V_{req}$ is slower than the data rate at which the precached content is delivered directly to $V_{req}$, this situation causes an additional delay for the content download. Thus, the next RSU can prevent this additional delay by preparing a guardband by considering the amount of content that cannot be received by $V_{req}$ in the previous RSU.

On the contrary, if the dwell time of $V_{req}$ is increased because of its increased speed, it can download a higher amount of content from the current RSU than expected. In this situation, $V_{req}$ can download a portion of the precached content that was already sent to the next RSU from the current RSU during the dwell time. Then, when $V_{req}$ connects to the next RSU, it does not need to download a portion of the content from the next RSU. Thus, $V_{req}$ can obtain the time required to download an additional amount of content from the next RSU. In particular, because the amount of precached content to the next RSU was already determined by the previous RSU, the additional amount that can be downloaded should be taken from the content server to the next RSU and delivered from the next RSU to $V_{req}$. This process is very inefficient in terms of the data rate and results in many delays. Therefore, preparing a guardband at the next RSU to download the additional amount can reduce delays.

4.2. Contact Time with the Relaying Vehicle

The dwell time between $V_{req}$ and an RSU within the RSU coverage was calculated using the mobility prediction of $V_{req}$. The contact time between the vehicles within the outage zone was calculated using the mobility predictions of both $V_{req}$ and the candidate vehicle. Therefore, the contact time has more prediction errors, as it is more difficult to predict. Two scenarios can occur in terms of mobility prediction error. First, if the speed of $V_{req}$ decreases, the contact time between $V_{req}$ and $V_{rel}$ increases and the arrival time to the next RSU increases. However, $V_{rel}$ cannot send an additional amount of content to $V_{req}$ during the increased contact time because it has received only the precached amount of content. Hence, it results in a delay wherein $V_{req}$ does not receive the additional amount of content after it finishes receiving the precached amount of the content from $V_{rel}$. Thus, preparing the guardband in advance for the potentially increased contact time can reduce the overall delay because the additional content in the guardband can be provided to $V_{req}$ when the contact time increases. Subsequently, if $V_{req}$ has a higher speed than the expected speed and does not receive certain portions of the content to download from the current RSU, $V_{rel}$ is not able to send the portion to $V_{req}$ because $V_{rel}$ has only the precached amount of the content. Therefore, the next RSU should provide a portion of the content to $V_{req}$ preferentially before it provides its own precached amount of content. This causes an additional delay because the next RSU must take a portion of the content from the previous RSU or the content server. Therefore, if $V_{rel}$ has a portion of the content as a guardband, it reduces the burden on the next RSU and additional delay.

Consequently, if both $V_{rel}$ and the next RSU prepare a guardband for the content to be precached, the overall burden of the RSU is reduced because it can immediately send the guardband to $V_{req}$ without using the content server or the previous RSU. In addition, the guardband can increase the hit ratio to download the content when an error occurs. Furthermore, the guardband is very efficient in reducing the time delay involved in downloading the entire content by $V_{req}$.

4.3. Calculation of the Guardband

To use the guardband of precached content, we need to determine how much it needs to be prepared. We can minimize the delay that the requester vehicle may experience when the next RSU allocates the remaining portion of the requested content as a guardband. However, this causes a significant consumption of traffic. The unused guardband is discarded as wasted traffic. Consequently, there should be a proper trade-off between traffic and delay, according to the guardband.
The delay must be calculated first because the amount of traffic consumed is calculated by the delay. In this study, the delay was calculated using the mobility information of the requester vehicle, the size of the precached content, and the size of the guardband. The expected reduced delay value $\text{delay}_{E,\text{req}}(G, \xi_{\text{mob}})$ according to the guardband can be obtained by the difference between the theoretical delay value $t_{\text{theo}}$ and the actual estimated delay value, $t_{\text{est}}$. Thus, we need to calculate both $t_{\text{theo}}$ and $t_{\text{est}}$.

First, the theoretical delay value $t_{\text{theo}}$ consumed for content download is predicted by using the current speed of the requester vehicle. To derive $t_{\text{theo}}$ based on the total size of the requested content and the transmission rate of the RSUs, the speed of $V_{\text{req}}$ and the distance that $V_{\text{req}}$ moves are used rather than the connection time. From this point of view, because the time that $V_{\text{req}}$ traveled 1 m at the speed of $v_{\text{req}}$ is $v_{\text{req}}^{-1}$, the size at which $V_{\text{req}}$ receives the requested content from an RSU is $R_{V_{21}} \times v_{\text{req}}^{-1}$. The first RSU, $RSU_1$, should receive the requested content from the content server via backhaul links before providing it to $V_{\text{req}}$. In this case, the size at which $V_{\text{req}}$ can receive content from $RSU_1$ is $(R_{V_{21}}+l^{-1})^{-1} \times v_{\text{req}}^{-1}$, where $l$ is the transmission rate of the backhaul link. However, because $t_{\text{theo}}$ is a time value, the theoretical distance value $m_{\text{theo}}$ consumed for content download should be defined first.

Here, $t_{\text{theo}} \equiv m_{\text{theo}} / v_{\text{req}}$, where $v_{\text{req}}$ is the speed of $V_{\text{req}}$. Therefore, the relationship between the $C_T$ and $m_{\text{theo}}$ is defined as follows:

$$
C_T = \sum_{m=0}^{m_1} \left( \left( R_{V_{21}}^{-1} + l^{-1} \right)^{-1} \cdot \frac{\xi_{\text{req},j}(m)}{v_{\text{req}}} + \sum_{j=2}^{m_{\text{theo}}} \left( R_{V_{21}} \cdot \frac{\sum_{k=1}^{\xi_{\text{req},j}-1} \delta_{\text{req},k}(m)}{v_{\text{req}}} + R_{V_{21}} \cdot \frac{\xi_{\text{req},j}(m)}{v_{\text{req}}} \right) \right),
$$

(12)

where $\xi_{\text{req},j}(m)$ denotes whether $RSU_j$ provides the requested content to $V_{\text{req}}$ after $V_{\text{req}}$ moves a distance of $m$. If $RSU_j$ provides the requested content to $V_{\text{req}}$ after $V_{\text{req}}$ moves $m$, $\xi_{\text{req},j}(m)$ is one. Otherwise, $\xi_{\text{req},j}(l)$ is zero. $R_{V_{21}}$ is the transmission rate between an RSU and a vehicle. $l$ is the transmission rate of backhaul links. $\delta_{\text{req},k}(m)$ indicates whether $V_{\text{rel}}$ communicates with $V_{\text{req}}$ after $V_{\text{req}}$ moves $m$. $\delta_{\text{req},j}(m)$ and $\delta_{\text{req},k}(m)$ can be either zero or one. $m_1$ is the distance at which $V_{\text{req}}$ reaches the end of the $RSU_1$’s coverage. To calculate $t_{\text{theo}}$, $\xi_{\text{req},j}(m)$ and $\delta_{\text{req},k}(m)$ can be known in advance because the mobility information of the vehicles is provided to the RSUs by accessing the server that manages the mobility information. The first sum represents the size of the content that $RSU_1$ receives from the content server over the backhaul link and provides to $V_{\text{req}}$. The second sum represents the sizes of the content that $V_k$ and $RSU_j$ provide $V_{\text{req}}$. In addition, because each node does not have an overlapped connection (in other words, a node cannot connect with another vehicle during connecting one vehicle), $\sum_{j=1}^{J} \sum_{k=1}^{K_{\text{req},j-1}} (\delta_{\text{req},k}(m) + \xi_{\text{req},j}(m)) \leq 1, \forall m$. $K_{\text{req},j-1}$ is the number of candidate vehicles that can become relaying vehicles within the coverage of $RSU_{j-1}$. In addition, because all values in sigma except $\xi_{\text{req},j}(m)$ are fixed, $C_T$ is derived through the sum of values that are defined by $\xi_{\text{req},j}(m)$, which can be zero or one. In addition, if the requester vehicle remains within $RSU_j$’s communication coverage at each $m$, $\xi_{\text{req},j}(m)$ must be 1. The relaying vehicle selection is conducted only outside the coverage of RSUs, where only vehicles that can provide the requested content to $V_{\text{req}}$ at each $m$ are considered candidate vehicles for the relaying vehicle. At each $m$, because only one vehicle is selected among the candidate relaying vehicles, $\sum_{k=1}^{K_{\text{req},j-1}} \delta_{\text{req},k}(m)$ cannot be greater than one. Moreover, when there are no candidate vehicles that can provide the requested content to the requester vehicle, $\sum_{k=1}^{K_{\text{req},j-1}} \delta_{\text{req},k}(m)$ becomes 0. Based on these constraints, the optimal $\xi_{\text{req},j}(m)$ can be calculated. The difference between $t_{\text{theo}}$ and the actual estimated content download delay $t_{\text{est}}$ depends on the magnitude of the mobility prediction error with respect to speed. The mobility prediction error for speed $v_{\text{req}}$ of the requester vehicle $V_{\text{req}}$ at the time of leaving $RSU_j$ is as follows:
where \( v_{\text{req}}(j) \) is the velocity of \( V_{\text{req}} \) within the coverage of RSU\(_j\) and \( \epsilon_{\text{mob}} \) is an error ratio for the speed of \( V_{\text{req}} \). Because it has a value of \([-n, 1]\), a value of one indicates that the vehicle stops. Because \( V_{\text{req}} \) will have a different average speed for each interval, the actual estimated speed \( v_{\text{req,av}}(j) \) of \( V_{\text{req}} \) within RSU\(_j\)'s coverage based on \( \epsilon_{\text{mob}} \) are as follows:

\[
v_{\text{req,av}}(j) = |v_{\text{req}}(j) \cdot \epsilon_{\text{mob}}|.
\]  

To calculate the actual estimated content download delay \( t_{\text{est}} \), the actual estimated content download distance \( m_{\text{est}} \) should first be defined because we use the distance \( V_{\text{req}} \) moves. \( m_{\text{est}} \) can be defined as follows:

\[
C_T = \sum_{m=0}^{m_1} \left( \left( R_{V2I}^1 + l^{-1} \right)^{-1} \cdot \frac{R_{\text{req},l}(t)}{v_{\text{req,av}}(1)} \right) + \sum_{j=2}^{l} \sum_{m=m_1}^{m_{\text{est}}-1} G \cdot R_{V2V} \cdot \frac{\sum_{k=1}^{K_{\text{req},l}} R_{\text{req},k}(m)}{v_{\text{req,av}}(j-1)}
+ G \cdot R_{V2V} \cdot \frac{P_{\text{req,j}}(t)}{v_{\text{req,av}}(m)} + (1 - G) \cdot \left( R_{V2I}^1 + l^{-1} \right)^{-1} \cdot \frac{R_{\text{req},l}(t)}{v_{\text{req,av}}(m)},
\]

where \( R_{\text{req},j}(m) \) indicates whether RSU\(_j\) relays the content from the content server to \( V_{\text{req}} \) after \( V_{\text{req}} \) moves \( m \) meters. \( v_{\text{req,av}}(j) \) is the average speed when \( V_{\text{req}} \) reaches RSU\(_{j+1}\) after leaving RSU\(_j\). \( G \) is the guardband ratio of the precached content and has a value of \([0, 1]\). \( P_{\text{req,j}}(m) \) indicates whether \( V_{\text{req}} \) receives the precached content through a node \( j \) (that is, an RSU or a candidate relaying vehicle) after \( V_{\text{req}} \) moves \( m \) m. \( R_{\text{req},j}(m) \) and \( P_{\text{req,j}}(m) \) can also be known in advance because the mobility information of the vehicles is provided to the RSUs. Therefore, the actual connection time between vehicle \( V_{\text{req}} \) and a candidate relaying vehicle \( V_k \) after \( V_{\text{req}} \) moves \( m \) m can be estimated by \( t_{\text{est}} = \sum_{m=0}^{m_{\text{est}}} \left( \frac{\text{conn}_{\text{req},j}(m)}{v_{\text{req,av}}(j)} + \frac{\text{out}_{\text{req},j}(m)}{v_{\text{req,av}}(j)} \right) \),

where \( \text{conn}_{\text{req},j}(m) \) denotes the connection state between \( V_{\text{req}} \) and RSU\(_j\). At \( m \), if \( V_{\text{req}} \) enters the coverage of RSU\(_j\), \( \text{conn}_{\text{req},j}(m) \) is equal to one. Otherwise, \( \text{conn}_{\text{req},j}(m) \) is zero. \( \text{out}_{\text{req},j}(m) \) represents the state in which \( V_{\text{req}} \) is within the outage area between RSU\(_j\) and RSU\(_{j+1}\). At \( m \), if \( V_{\text{req}} \) remains within the outage zone between RSU\(_j\) and RSU\(_{j+1}\), \( \text{out}_{\text{req},j}(m) \) is one. Otherwise, \( \text{out}_{\text{req},j}(m) \) is zero. Subsequently, by recalculating the actual connection time, \( P_{\text{req,j}}(m) \) can be redefined from zero to one or from one to zero. Consequently, the expected reduced delay \( \text{delay}_{E,\text{re}}(G, \epsilon_{\text{mob}}) \) according to \( G \) and \( \epsilon_{\text{mob}} \) can be estimated by the difference between \( t_{\text{theo}} \) and \( t_{\text{est}} \), as follows:

\[
\text{delay}_{E,\text{re}}(G, \epsilon_{\text{mob}}) = t_{\text{theo}} - t_{\text{est}}.
\]

On a similar principle to the expected reduced delay, we can also calculate the expected reduced backhaul traffic \( \text{traffic}_{E,\text{re}}(G, \epsilon_{\text{mob}}) \) according to the guardband, which is consumed by precaching the content from both the content server to the RSUs and from the RSUs to the relaying vehicles, and by sending the precached content to \( V_{\text{req}} \). The \( \text{traffic}_{E,\text{re}}(G, \epsilon_{\text{mob}}) \) can be obtained from the difference between the theoretical backhaul traffic value, \( \text{traffic}_{\text{theo}} \), and the actual backhaul traffic value \( \text{traffic}_{\text{est}} \). Using the current speed of \( V_{\text{req}} \), the theoretical backhaul traffic was obtained by \( m_{\text{theo}} \) as follows:
where $H(j)$ is the number of hops between the content server and the $RSU_j$. Because it uses the resources of the links between backhaul nodes to obtain the content, the traffic consumption is proportional to the number of hops. For example, if the content needs to be delivered from $RSU_{(j-2)}$ to $RSU_j$, $RSU_{(j-2)}$ first passes the content to $RSU_{(j-1)}$ via the backhaul link and $RSU_{(j-1)}$ delivers the received content back to $RSU_j$. Thus, it consumes twice the amount of backhaul traffic.

The actual estimated traffic $traffic_{est}$ is also calculated by $m_{est}$, as follows:

$$traffic_{est} = \sum_{m=0}^{m_{est}} \left( H(1) \cdot (R_{V2V}^{-1} + I^{-1})^{-1} \cdot \frac{\xi_{req,1}(m)}{v_{req}} \right) + \sum_{j=2}^{m_{est}} \sum_{m=m_1}^{m_{est}} \left( H(j) \cdot \frac{\xi_{req,j}(m)}{v_{req}} \right) + H(j-1) \cdot R_{V2V} \cdot \sum_{k=1}^{K_{req,j-1}} \frac{\delta_{req,k}(m)}{v_{req}} ,$$

(19)

As a result, by using $traffic_{theo}$ and $traffic_{est}$ based on $m_{theo}$ and $m_{est}$, we can formulate the expected reduced backhaul traffic $traffic_{E, re}(G, \epsilon_{mob})$ based on $G$ and $\epsilon_{mob}$ by the difference between $traffic_{theo}$ and $traffic_{est}$ as follows:

$$traffic_{E, re}(G, \epsilon_{mob}) = traffic_{theo} - traffic_{est}.$$

(20)

To address guardband $G$, $G$ can be added to compensate for the increased delay and backhaul traffic owing to $\epsilon_{mob}$. However, we cannot determine $\epsilon_{mob}$ for the prediction of the vehicle’s mobility in advance. The position of each RSU and the characteristics of each requester vehicle shall be applied to effectively predict $\epsilon_{mob}$ and appropriately adjust the size of $G$. Thus, we evaluated the performance of the proposed scheme by adding $G$ through our simulations in Section 5.4. The delay in downloading the content may depend on the characteristics of the content requested by $V_{req}$. The traffic available to $V_{req}$ may also be limited according to the communication service provided by the RSUs. Because there is a trade-off between the delay and the traffic, if one of them is limited, we can find the size of $G$, which can optimize the other one.

5. Performance Evaluation

In this section, we compare the performance of the proposed scheme with that of the existing schemes. First, we describe the simulation model and scenario implemented in our simulations. Next, we evaluated the performance of the proposed scheme and that of the existing schemes through simulation results conducted in various environments.

5.1. Simulation Model and Scenario

We employed simulations of NS3 [33] to evaluate the performance of the proposed scheme and the existing schemes in various environmental variables. We simulated requests for video content in the Center for Brooklyn History using NS3. Table 3 lists the simulation parameters used. In the network area of $5 \text{ km} \times 5 \text{ km}$, 25 RSUs are located at the intersections, and each RSU has an $800 \text{ m-radius}$ circle as its coverage. A total of
100 vehicles per 1 km² are randomly distributed on the network at the beginning of the simulations, and they move on the roads of the network according to the RoutesMobilityModel [34], which provides traces of vehicles from Google Maps Platform using the API key. To improve the accuracy of our simulations, we increased the granularity to one second by linear interpolation. We also used this trace to extract the necessary mobility statistics for our model. The wireless resource includes four 10 MHz channels, and the transmission rate is obtained from the IEEE 802.11p standard [35], which is provided by default in NS3. We set content popularity according to Zipf’s law [36].

| Parameters                         | Value                      |
|------------------------------------|----------------------------|
| link latency                       | 10 ms                      |
| radio channel rate                 | 54 Mbps                    |
| backhaul rate                      | 1 Gbps                     |
| distance between RSUs              | [1.6, 4] km                |
| vehicle density                    | [25, 250] per km²          |
| size of the content catalog        | 10                         |
| exponent of content popularity     | 0.75                       |
| chunk size                         | 25 kbytes                  |
| content size                       | [500, 3500] MB             |
| vehicle speed                      | [10, 120] km/h             |
| V2I Communication range            | 800 m                      |
| V2V Communication range            | 200 m                      |
| vehicle cache storage              | 512 GB                     |
| RSU cache storage                  | 1 TB                       |
| prediction error                   | [0, 1]                     |
| G guardband                        | [0, 1]                     |

We compared the proposed scheme with three existing schemes. (1) The first scheme is the basic CCVN scheme, which did not use precaching to provide the requested content to $V_{req}$. (2) The second scheme is the only RSU scheme in which RSUs precache the content to provide it to $V_{req}$. Because the only RSU scheme did not use other vehicles as relaying vehicles, they did nothing to provide content to $V_{req}$. (3) The third scheme is a random selection scheme that provided the content to $V_{req}$ using precaching in both RSUs and relaying vehicles. However, the random selection scheme randomly selects relaying vehicles, which is different from the proposed scheme. We used four metrics to compare the performance of the proposed scheme and the three existing schemes as follows:

- The content download delay [s], the time in which $V_{req}$ receives the entire amount of the content from the RSUs and the relaying vehicles;
- The backbone traffic except precached traffic [MB]: It is the total amount of backhaul traffic consumed to provide the content to $V_{req}$ using a backbone;
- The precached traffic by the next RSUs [MB]: It is through the prediction, the total amount of traffic consumed by the next RSUs to precache the remaining amount of content that $V_{req}$ has to receive when the current RSU cannot provide the entire amount of content;
- The RSU usage ratio, $U_c \in [0,1]$: It is the proportion of the amount of content provided from RSUs among the total amount of requested content. The time slot consumed by the RSU to provide content to $V_{req}$ cannot be used for other vehicles. Therefore, if the RSU usage ratio is small, the time slot that other vehicles can use to receive the content will increase. In addition, RSUs benefit from a storage utilization perspective because they reduce the amount of content that needs to be precached and held;
- The hit ratio $\in [0,1]$: It is the proportion of the content used by $V_{req}$ among the total amount of precached content at both RSUs and relaying vehicles. Some portion of the content that is precached at the RSUs and relaying vehicles may not be provided to $V_{req}$ because of the limited connection between $V_{req}$ and each of them. Even if they are
not provided to \( V_{req} \), they consume the storage of the RSUs and relaying vehicles and are eventually removed. This causes wasted backhaul traffic from the content server and wasted storage of RSUs and relaying vehicles. Therefore, a high hit ratio implies that storage and traffic are saved.

We evaluated the performance from three different perspectives to compare the proposed scheme with existing schemes. (1) The first perspective concerns the various environmental factors in the network because they can significantly influence performance. As influential environmental factors, we considered the density of vehicles and the distance between the RSUs in our performance evaluation. Vehicle density refers to the number of vehicles within the simulation area. This is implemented by adjusting the number of vehicle trajectories obtained from NS3 using the RoutesMobilityModel. A high density of vehicles is considered to represent a busy urban situation, whereas a low density of vehicles is considered to represent a rural situation. The distance between the RSUs is the distance between two neighboring RSUs. By adjusting the number of RSUs, we varied the distance between RSUs in the simulation area. If the number of RSUs decreased, the distance between the RSUs increased within the simulation area. In this study, the distance between the RSUs affected the size of the outage zone between them. (2) The second perspective concerns the characteristics related to requester vehicles because they should be supported in the view of services. We considered the size of the requested content and the speed of \( V_{req} \) in our performance evaluation. The size of the requested content refers to the size of the content that requester vehicles require to download from the content servers. Usually, users in requester vehicles can request content of various sizes. To vary the size of the contents, we adjusted the number of fixed-size chunks that are consistent with the requested content. A larger size of the requested content increases the time required to complete the download. Thus, while receiving the content, requester vehicles that request larger content should pass more RSUs and outage zones. The speed of \( V_{req} \) is the average speed of requester vehicles in the network. They can have variable average speeds depending on the driving style of their drivers. We varied the average speed of the requester vehicles by adjusting their speeds or by appropriately compressing their trajectories from a time perspective. If the average speed of the requester vehicles is higher, they pass faster RSUs and outage zones. (3) The third perspective is the accuracy of vehicle mobility prediction. Because the existing schemes did not consider errors in the prediction of vehicle mobility, they should request un-precached chunks because of the prediction errors, to the content server again. The un-precached chunks are inefficient in content download because they cannot be delivered directly to \( V_{req} \). The probability of requesting un-precached chunks depends on the error rate of vehicle mobility prediction. The proposed scheme uses guardbands to compensate for errors in the prediction of the vehicle mobility. Therefore, we evaluated the performance according to the size of the guardband based on the error rate of vehicle mobility prediction.

5.2. Performance for Various Environmental Factors

In Figure 2, we evaluate the performance of the schemes by using the density of vehicles. Figure 2a depicts the effect of vehicle density on the time it takes for \( V_{req} \) to receive all the requested content. For the proposed scheme and the random selection scheme, a higher density of vehicles increases the probability of providing additional requested content because there are many vehicle candidates to deliver the content to \( V_{req} \) outside the RSU communication range. Therefore, the delay is reduced because the remaining amount of content to be received in the next RSU is reduced. Furthermore, the delay is reduced because the proposed scheme selects better candidates than the random-selection scheme. Conversely, the basic CCVN and RSU schemes were not affected by the density of the vehicles because they did not use relaying vehicles. Even though the basic CCVN scheme does not use a precache, there is a disadvantage in terms of the transmission speed, which consumes the most time.
Figure 2. Simulation results for different densities of vehicles: (a) the content download delay, (b) the backhaul traffic except precached traffic, (c) the precached traffic by next RSU and (d) the RSU usage rate.

In Figure 2b, we evaluate the effect of vehicle density on the consumed backhaul traffic, except for precached traffic. The basic CCVN scheme consumes the resources of the links between the current RSU and the content server because it takes the requested content from the content server and provides it to $V_{req}$. Therefore, the performance is severely affected by the number of hops in terms of transmission speed. Only the RSU scheme performed better than the basic CCVN scheme because it precaches the content in the next RSU. However, these two schemes are not affected by vehicle density. The random selection scheme has improved performance because more relaying vehicles are available when the density of the vehicles is high. The proposed scheme exhibits the best performance because it can select the optimized relaying vehicles. Because the area between the current and next RSU is limited, the performance of the proposed scheme converges when there is a sufficient number of candidates.

Figure 2c shows the precasting traffic according to the vehicle density. In the proposed scheme, a higher density of vehicles allows the selection of more optimal relaying vehicles, thereby reducing the size of the content that the next RSU should precache. This leads to a reduction in precasting traffic. In the random selection scheme, a lower density of vehicles increases the probability of using the worst or no candidate for the relaying vehicles; therefore, a larger amount of content should be precached at the next RSU. However, the RSU scheme is not affected by vehicle density. Furthermore, as the basic CCVN scheme does not use precasting, there is no precasting traffic.

Figure 2d shows the percentage of RSUs used to provide content according to the density of the vehicle. The basic CCVN and RSU schemes depend only on RSUs for content delivery because they do not use vehicles as relay nodes. The random selection scheme improves performance because the probability of selecting a better relaying vehicle at a higher density of vehicles increases. The proposed scheme effectively reduced the rate of...
dependence on RSUs because optimal relaying vehicles can be selected at a higher density of vehicles.

In Figure 3, we evaluated the performance according to the density of the RSUs by adjusting the distance between the RSUs. Figure 3a depicts the effect of the density of RSUs on the content download delay. In the basic CCVN scheme and the only RSU scheme, because vehicles are not used as relay nodes, the content download delay increases as the time that \( V_{req} \) reaches the next RSU increases when the distance between RSUs is long. Additionally, because the basic CCVN scheme is inefficient in terms of the transmission rate, it has the longest delay. However, in the proposed scheme and the random selection scheme, because a sufficient number of relaying vehicles are used to provide the content for \( V_{req} \) outside the RSUs’ communication range, it is less affected by the density of RSUs. Because the random selection scheme does not select the optimal relaying vehicles, \( V_{req} \) must receive the remaining portion of the content from the next RSU. Hence, on average, the proposed scheme has less delay than the random-selection scheme.

Figure 3. Simulation results for different distances between RSUs: (a) the content download delay, (b) the backhaul traffic except precached traffic, (c) the precached traffic by next RSU and (d) the RSU usage rate.

Figure 3b shows the effect of the density of RSUs on traffic that relays the content directly from the content server to \( V_{req} \), except for the traffic consumed by RSUs for pre-caching. The relayed traffic consumed by the RSU to relay content directly from the content server to \( V_{req} \) has a slow transmission rate, consuming more links than the RSU directly delivering the precached content to the vehicle. Because the content cannot be precached in the first RSU, all schemes relayed traffic from the content server. In particular, the basic CCVN scheme provides the entire content by relaying it from the content server to the vehicle as it does not use precaching. In the case of the basic CCVN scheme and the only RSU scheme, the backhaul traffic generated is not related to the distance between RSUs, because the content can only be provided within the RSU communication range. However, if the next RSU is far-off, the RSU consumes less traffic to relay directly from the content.
server to $V_{req}$ because the proposed scheme and random selection scheme precache most of the content on the relaying vehicles. Because the proposed scheme prepares more content for optimized relaying vehicles, the relayed traffic is the least.

Figure 3c presents the traffic consumed by RSUs to precache the content according to the density of the RSUs. The basic CCVN scheme does not use precaching. The RSU scheme is not affected by the distance between the RSUs because $V_{req}$ only receives the content within the RSU communication range. However, in the proposed and random selection schemes, the traffic consumed by the precache to be borne by RSUs is reduced because $V_{req}$ receives most of the content via V2V communication, except for the first RSU. With the optimal candidate selection algorithm, the proposed scheme significantly relieves the burden of the next RSUs.

In Figure 3d, we present the proposed scheme and the random selection scheme for mitigating the burden on the RSU. Because these two schemes provide almost all of the content to $V_{req}$ via V2V communication if the distance between RSUs is large, this reduces the amount of content that the RSUs have to support. Furthermore, the proposed scheme is more effective than other schemes because it uses a larger number of available relaying vehicles.

5.3. Performance Based on Characteristics of Requester Vehicles

In Figure 4, we evaluate the performance of the proposed scheme according to various sizes of content because the user can request various types of content. Figure 4a shows that the content download delay is proportional to the size of the requested content. Because the transmission rate is limited, the delay in receiving the entire content increases in all schemes when the size of the requested content is large. In the basic CCVN scheme, because the entire requested content is relayed to $V_{req}$ from the content server, it is the most inefficient in terms of transmission rate. Therefore, this scheme has the greatest delay in downloading the content. Because only the RSU scheme receives the requested content from the RSU, additional delays are accumulated as they move outside the RSUs’ communication range. However, the proposed scheme and the random selection scheme consumed less delay because they can provide the content when $V_{req}$ is not only in the RSUs’ communication range but also out of the range. The random selection scheme, which randomly selects relaying vehicles, is inefficient compared with the proposed scheme.

In Figure 4b, we evaluate the impact of the increase in the size of the requested content on backhaul traffic. Because the backhaul link consumed increases proportionally to the number of hops between the content servers and the next RSUs, the traffic consumption is larger than the size of the requested content. All schemes, except the basic CCVN scheme, consumed the most backhaul traffic when the requested content was provided in the first RSU. In addition, in these schemes, additional traffic increases with the difference between the size of the precached content and the actual received content. Because the RSU scheme uses only the RSU, it consumes more backhaul traffic than the two schemes using relaying vehicles. On the other hand, the basic CCVN scheme receives all chunks of content from the content server through relaying. Therefore, it consumes the most relayed traffic.

Figure 4c shows the relationship between the size of the requested content and precaching traffic. To precache the content, the next RSU takes chunks of the content from the content server. Hence, the consumed traffic relies on the number of hops between the content servers and the content server. The basic CCVN scheme does not use a precache. In the case of the RSU scheme, most of the traffic is consumed because all chunks of the content, except the portion of the received content from the first RSU, are provided to $V_{req}$ by the next RSUs precaching the content. Because the other two schemes use relaying vehicles to precache the content, the traffic that RSUs are burdened with precaching the content is reduced.

Figure 4d shows the burden of RSUs according to the size of the requested content. Because the proposed scheme selects optimized relaying vehicles, the RSUs’ burden is reduced the most. In comparison, the random selection scheme that selects any relaying vehicle cannot guarantee that $V_{req}$ is served with maximum efficiency outside the RSUs’
communication range. Thus, there is a greater burden on the RSUs than in the proposed scheme. The size of the requested content that can be provided is limited because \( V_{\text{req}} \) has available time owing to the mobility of the vehicle outside of the RSUs’ communication range. Therefore, a sufficient size of the requested content converges to alleviate the burden of the RSUs.

In Figure 5, we also evaluate the performance of various user characteristics. Figure 5a shows the content download delay according to the speed of the vehicle. If the speed of the vehicle is low, the next RSU and \( V_{\text{rel}} \) do not have to precache the content because \( V_{\text{req}} \) has sufficient dwell time to receive the entire content from the content server in the first RSU. Otherwise, if the speed of the vehicle is high, the number of content chunks not received by the first RSU increases. Therefore, the content download delay is increased because \( V_{\text{req}} \) requires time to reach the next RSU to receive the rest of the content. Subsequently, the delay continues to decrease because the time remaining in the outage zone decreases with increasing speed before additional RSUs are required. Because only the RSU scheme and the basic CCVN scheme that do not use the relaying vehicles are provided with content only through the RSU to \( V_{\text{req}} \), the content download delay in both schemes depends more on the time \( V_{\text{rel}} \) required to reach the next RSU than in the other two schemes. The faster speed of the requester vehicle can decrease the download delay because it can reduce the time within the outage zone. However, at some speed of the requester vehicle, the increased delay due to the decreased time that the requester vehicle can receive the requested content within the coverage of RSUs has more impact than the decreased delay by reduced time within the outage zone. That is why the content download delay of the two schemes significantly fluctuated. The RSU scheme has a shorter delay than the basic CCVN scheme because the next RSU precaches the content. In the random selection scheme, the delay...
is shorter because it is provided by the randomly selected $V_{rel}$ in the outage zone. The proposed scheme has the least delay because it selects the optimal $V_{rel}$.

![Figure 5. Simulation results for different speeds of the requester vehicle: (a) the content download delay, (b) the backhaul traffic except precached traffic, (c) the precached traffic by next RSU, and (d) the RSU usage rate.](image)

Figure 5b shows the impact of backhaul traffic, excluding precaching traffic, on the speed of $V_{req}$. The backhaul traffic, except for the precaching traffic, is the backhaul traffic consumed by the RSU to relay content from the content server to $V_{req}$. Because relaying the content from the content server to $V_{req}$ is inefficient in terms of the transmission rate, small backhaul traffic indicates good performance. First, in terms of this indicator, the basic CCVN scheme is unaffected by mobility prediction error because it does not use precaching. In the RSU scheme, only the RSU precaches the content; therefore, it outperforms the basic CCVN scheme. In the case of the random selection scheme, inefficient traffic is consumed less than in the RSU scheme because $V_{rel}$ precaches the content. The proposed scheme has the smallest backhaul traffic because $V_{rel}$ precaches the maximum available content size in the outage zone.

Figure 5c shows the relationship between the speed of $V_{req}$ and the backhaul traffic consumed by the RSU for precaching. The basic CCVN scheme does not precache the content. Because the proposed scheme uses optimized $V_{rel}$, the RSU has the least traffic to precache the content. Compared with the proposed scheme, the random selection scheme performs poorly because it randomly selects $V_{rel}$. In the RSU scheme, the next RSU precaches the entire content, except for some of the content that is relayed by the first RSU from the content server to the vehicle. Thus, in this scheme, RSUs consume the most backhaul traffic to precache the content.

Figure 5d depicts the trends in the involvement of RSUs in providing content according to the speed of the vehicle. $V_{req}$ is provided by the RSU in only two schemes that do not use $V_{rel}$. Because the proposed scheme uses the optimal $V_{rel}$, it uses the least number of RSUs in terms of providing content to $V_{req}$. In the case of the random selection scheme, more
RSUs are used than in the proposed scheme, because $V_{rel}$ is chosen randomly. In these two schemes using $V_{rel}$, the performance converges as the connection time with other vehicles, and the dwell time in the outage zone is reduced if $V_{req}$ is too fast.

5.4. Performance for Accuracy of Vehicle Mobility Prediction

In Figure 6, we evaluated the performance according to the size of the guard band applied to compensate for the difference between the predicted and actual mobilities of the vehicle. Figure 6a presents the content download delay caused by the prediction error when the guard band is applied. The basic CCVN scheme is not affected by prediction errors in vehicle mobility, because it does not use precaching. Only the RSU scheme has an increased size of the content taken from the former RSU or content server owing to prediction error. The two schemes using relaying vehicles take more time to download the entire content, as the size of the content provided by the relaying vehicles decreases if the prediction error of the vehicle mobility is large. However, as the size of the guard band increases, performance degradation from a delay perspective is mitigated.

Figure 6b depicts the required backhaul traffic, except for precaching, using a guard band to mitigate performance degradation owing to the mobility prediction error. In the basic CCVN scheme, because the RSU does not precache the requested content using the prediction of vehicle mobility, it provides the requested content to the vehicle from the content server via relaying. In the other schemes, if the prediction error increases, backhaul traffic is consumed as much as it differs from the prediction. Because the proposed scheme uses optimized relaying vehicles to provide content to $V_{req}$, it consumes the lowest backhaul traffic. However, because accurate prediction is impossible in a real environment, we use a guard band to cover the prediction error. This reduces the amount received using precaching. Therefore, as the size of the additional content to be received increases, so does the size of the guardband prepared to compensate for the error. In addition, if the size of the guard band is increased to cover more prediction errors, the traffic required for preparing the guard band is further increased.

Figure 6c shows the traffic for precaching owing to the use of the guard band. In all the schemes, if the prediction is perfect, the next RSU precaches the appropriate size content requested by the vehicle. The mobility prediction error is caused by predicting faster or slower than the actual speed of $V_{req}$. Because $V_{req}$, which is faster than expected, causes a smaller dwell time than the actual dwell time, $V_{req}$ is not fully provided with precached content. $V_{req}$, which was slower than the actual speed, had a longer dwell time. The RSU precaches a smaller-sized content, consuming less traffic than the actual size of the content that can be received by vehicles with longer dwell times. Thus, if the mobility prediction is incorrect, the traffic consumed by precaching decreases. Because the proposed scheme uses optimized relaying vehicles, the least amount of traffic is consumed to perform the precaching. It consumes more traffic to prepare the guard band by applying it to cope with prediction errors.

Figure 6d presents the RSU usage ratio with the adjustment of the guard band. The two schemes that did not use relaying vehicles provided content to $V_{req}$ using only the RSU. The proposed scheme uses less RSU than the random selection scheme, which randomly selects relaying vehicles. With a large mobility prediction error, the guardband can cover the error to reduce the RSU usage ratio by utilizing the guardband contained in the relaying vehicles. If the mobility prediction error exceeds the size covered by the guard band, the performance improvement converges.

In Figure 7, we evaluate the hit ratio for the precached content by adding a guard band. The basic CCVN scheme does not use precaching. If the mobility prediction error is small, the hit ratio is more affected by the connection time between $V_{req}$ and $V_{rel}$ than by the dwell time on the RSU. Therefore, the hit ratio in the RSU scheme was higher than that in random selection in that case. Otherwise, if the mobility prediction error is large, the hit rate in the RSU scheme decreases dramatically because of the large impact of the dwell time on the RSU. The proposed scheme had the highest hit ratio because of the selection...
of optimized relaying vehicles. Furthermore, the large size of the guard band alleviates the reduced hit ratio owing to the mobility prediction error. This indicates that most of the traffic consumed was directed to $V_{req}$.

---

**Figure 6.** Simulation results for different error rates of the mobility predictions with various guard-bands: (a) the content download delay, (b) the backhaul traffic except precached traffic, (c) the precached traffic by next RSU and (d) the RSU usage rate.

**Figure 7.** The hit ratio for the error rate of the mobility prediction.
6. Discussion

In this section, we discuss several issues related to the cooperative content precaching scheme and its viability in a practical context.

6.1. Security of Privacy

In the proposed scheme, the trajectory and mobility information of vehicles are shared with RSUs and other vehicles to achieve better precaching performance. In addition, the content that a vehicle wishes to download is cached or precached distributively to the RSUs or other vehicles. In this case, the requester vehicle’s private information, which includes the mobility and preference for the content, is disclosed to others. Because disclosed information about the requester vehicle can be used to attack or threaten the user, it must be protected. Therefore, many studies have proposed protection schemes to hide private information, while the precaching service is served to the requester vehicle using blockchain or other technologies [37,38]. Reliable precaching services can only be used when safety is guaranteed. Thus, when the proposed scheme uses the proper security protocol for precaching, it can achieve reliable service while maintaining high network performance.

6.2. Energy Consumption

In vehicular networks, saving energy is one of the most important issues to achieve a longer driving time owing to the limitation of the battery. In the proposed scheme, almost all the computing processes to precache the content are conducted at RSUs. However, the relaying vehicle spends the energy required to reach the requested content and provide it to the requester vehicle. Therefore, several studies have been conducted to reduce the energy consumed in precaching the requested content [39–42]. To save energy, it is possible to increase the delay in content download and backhaul traffic used to serve the precaching service [40]. Thus, this is a very sensitive issue for reducing the consumed energy to precache the content at RSUs or other vehicles. When a proper energy-saving method is considered in the proposed scheme, the scheme can be improved in terms of energy and other network performances.

6.3. Combination with 5G Technology

For viability in the practical context of a precaching scheme, it is necessary to consider 5G or 6G technologies. To provide precaching service to vehicles in 5G, numerous studies have been conducted [43,44]. Because many parameters and environments have become standards for 5G, these must be considered when conducting a precaching service in a real scenario. In 5G technology, many modules are provided for each purpose. These modules can be considered to improve the network performance, such as delay, traffic, or security problems in the proposed scheme.

7. Conclusions

In this study, we propose a cooperative content precaching scheme based on the mobility information of vehicles to enhance the performance of content downloads in ICVNs. In contrast to the existing schemes, our scheme cooperatively used both vehicles and RSUs for the precaching content. For content precaching based on the mobility information of vehicles, our scheme selected both the optimal vehicle for relaying content in the outage zone and the next RSU to download the content in the coverage of the RSU. Our scheme calculated the optimal content precaching amount for each optimal relaying vehicle and the next downloading RSU using a mathematical model that used both the dwell time of vehicles in an RSU and contact time between vehicles. To compensate for the error of the vehicle mobility prediction to determine both the dwell and contact times, our scheme added a guard band to the optimal content precaching amount. For mobility prediction errors, the guard band is calculated with the error rate of both the dwell and contact times, and it is adjusted in the trade-off between the expected reduced delay and backhaul traffic. We conducted simulations to compare the performance of our scheme with the performances
of existing schemes from three perspectives: Various environmental factors, characteristics related to requester vehicles, and accuracy of vehicle mobility prediction. Simulation results verify that our scheme achieves better performance than existing schemes in terms of the content download delay, backbone traffic, precaching traffic, RSU usage ratio, and hit ratio. In future work, we will consider other vehicles or drones that have already cached the requested content, as providers without using additional backhaul traffic. In addition, using an additional server, we can employ a machine learning method to improve the performance of the proposed scheme.

Author Contributions: Conceptualization, J.B.; methodology, Y.N.; software, Y.N.; validation, H.C., Y.S. and J.B.; writing—original draft preparation, Y.N.; writing—review and editing, H.C. and E.L.; supervision, E.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by: the National Research Foundation of Korea (NRF), grant funded by the Korea government (Ministry of Science and ICT) (No. NRF-2022R1A2C1010121); the support of the Korea Institute of Industrial Technology (P0008421) funded by the government (Ministry of Trade, Industry and Energy).

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

References
1. Karagiannis, G.; Altintas, O.; Ekici, E.; Heijen, G.; Jarupan, B.; Lin, K.; Weil, T. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, and Standards. IEEE Commun. Surv. Tutor. 2011, 13, 584–616. [CrossRef]
2. Awang, A.; Hussain, K.; Kamel, N.; Aissa, S. Routing in Vehicular Ad-hoc Networks: A Survey on Single- and Cross-Layer Design Techniques, and Perspectives. IEEE Access 2017, 5, 9497–9517. [CrossRef]
3. Shahwani, H.; Shah, S.A.; Ashraf, M.; Akram, M.; Jeong, J.; Shin, J. A comprehensive survey on data dissemination in Vehicular Ad Hoc Networks. Veh. Commun. 2021, 34, 100420. [CrossRef]
4. An, K.; Lin, M.; Ouyang, J.; Zhu, W.-P. Secure Transmission in Cognitive Satellite Terrestrial Networks. IEEE J. Sel. Areas Commun. 2016, 34, 3025–3037. [CrossRef]
5. Lin, Z.; Niu, H.; An, K.; Wang, Y.; Zheng, G.; Chatzinotas, S.; Hu, Y. Refracting RIS-Aided Hybrid Satellite–Terrestrial Relay Networks: Joint Beamforming Design and Optimization. IEEE Trans. Aerosp. Electron. Syst. 2022, 58, 3717–3724. [CrossRef]
6. Lin, Z.; Lin, M.; de Cola, T.; Wang, J.-B.; Zhu, W.-P.; Cheng, J. Supporting IoT With Rate-Splitting Multiple Access in Satellite and Aerial-Integrated Networks. IEEE Internet Things J. 2021, 8, 11123–11134. [CrossRef]
7. Lin, Z.; An, K.; Niu, H.; Hu, Y.; Chatzinotas, S.; Zheng, G.; Wang, J. SLNR-based Secure Energy Efficient Beamforming in Multibeam Satellite Systems. IEEE Trans. Aerosp. Electron. Syst. 2022, 1–4. [CrossRef]
8. Lee, E.; Lee, E.-K.; Gerla, M.; Oh, S.Y. Vehicular cloud networking: Architecture and design principles. IEEE Commun. Mag. 2014, 52, 148–155. [CrossRef]
9. Zhang, S.; Chen, J.; Lyu, F.; Cheng, N.; Shi, W.; Shen, X. Vehicular Communication Networks in the Automated Driving Era. IEEE Commun. Mag. 2018, 56, 26–32. [CrossRef]
10. Khelifi, H.; Luo, S.; Nour, B.; Mouniga, H.; Faheem, Y.; Hussain, R.; Ksentini, A. Named Data Networking in Vehicular Ad Hoc Networks: State-of-the-Art and Challenges. IEEE Commun. Surv. Tutorials 2019, 22, 320–351. [CrossRef]
11. Cheng, X.; Yang, L.; Shen, X. D2D for Intelligent Transportation Systems: A Feasibility Study. IEEE Trans. Intell. Transp. Syst. 2015, 16, 1784–1793. [CrossRef]
12. Wang, Y.; Zheng, J.; Mitton, N. Delivery Delay Analysis for Roadside Unit Deployment in Vehicular Ad Hoc Networks With Intermittent Connectivity. IEEE Trans. Veh. Technol. 2015, 65, 8591–8602. [CrossRef]
13. Lopes, L.H.S.; Mini, R.A.F.; Cunha, F. A V2X Approach for Data Dissemination in Vehicular Ad Hoc Networks. In Proceedings of the 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 29 June–3 July 2019; pp. 1–6. [CrossRef]
14. Wang, X.; Li, Y. Content Delivery Based on Vehicular Cloud. IEEE Trans. Veh. Technol. 2019, 69, 2105–2113. [CrossRef]
15. Ko, B.; Liu, K.; Son, S.H.; Park, K.-J. RSU-Assisted Adaptive Scheduling for Vehicle-to-Vehicle Data Sharing in Bidirectional Road Scenarios. IEEE Trans. Intell. Transp. Syst. 2020, 22, 977–989. [CrossRef]
16. Huang, W.; Li, P.; Li, B.; Zhang, T. DMP: Content Delivery with Dynamic Movement Pattern in Vehicular Networks. IEEE Trans. Big Data 2021, 8, 1. [CrossRef]
17. Reis, A.B.; Sargento, S.; Neves, F.; Tonguz, O.K. Deploying Roadside Units in Sparse Vehicular Networks: What Really Works and What Does Not. IEEE Trans. Veh. Technol. 2013, 63, 2794–2806. [CrossRef]
18. Kim, D.; Velasco, Y.; Wang, W.; Uma, R.N.; Hussain, R.; Lee, S. A New Comprehensive RSU Installation Strategy for Cost-Efficient VANET Deployment. IEEE Trans. Veh. Technol. 2016, 66, 4200–4211. [CrossRef]
19. Su, Z.; Hui, Y.; Yang, Q. The Next Generation Vehicular Networks: A Content-Centric Framework. IEEE Wirel. Commun. 2017, 24, 60–66. [CrossRef]
20. Rao, Y.; Zhou, H.; Gao, D.; Luo, H.; Liu, Y. Proactive Caching for Enhancing User-Side Mobility Support in Named Data Networking. In Proceedings of the 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Taichung, Taiwan, 1–3 July 2013; pp. 37–42. [CrossRef]

21. Wang, Y.; Liu, Y.; Zhang, J.; Ye, H.; Tan, Z. Cooperative Store-Carry-Forward Scheme for Intermittently Connected Vehicular Networks. IEEE Trans. Veh. Technol. 2017, 66, 777–784. [CrossRef]

22. Suthaputchakun, C.; Sun, Z. Multihop Broadcast Protocol in Intermittently Connected Vehicular Networks. IEEE Trans. Aerosp. Electron. Syst. 2017, 54, 616–628. [CrossRef]

23. Wu, D.; Zhu, G.; Zhao, D. Adaptive Carry-Store Forward Scheme in Two-Hop Vehicular Delay Tolerant Networks. IEEE Commun. Lett. 2013, 17, 721–724. [CrossRef]

24. Zhou, H.; Cheng, N.; Wang, J.; Chen, J.; Yu, Q.; Shen, X. Toward Dynamic Link Utilization for Efficient Vehicular Edge Content Distribution. IEEE Trans. Veh. Technol. 2019, 68, 8301–8313. [CrossRef]

25. Wang, R.; Peng, X.; Zhang, J.; Letait, K.B. Mobility-aware caching for content-centric wireless networks: Modeling and methodology. IEEE Commun. Mag. 2016, 54, 77–83. [CrossRef]

26. Grewe, D.; Wagner, M.; Frey, H. PeRCeIVE: Proactive caching in ICN-based VANETs. In Proceedings of the 2016 IEEE Vehicular Networking Conference (VNC), Columbus, OH, USA, 8–10 December 2016; pp. 1–8. [CrossRef]

27. Lai, C.; Zhang, K.; Cheng, N.; Li, H.; Shen, X. SIRC: A Secure Incentive Scheme for Reliable Cooperative Downloading in Highway VANETs. IEEE Trans. Intell. Transp. Syst. 2016, 18, 1559–1574. [CrossRef]

28. Mershad, K.; Artail, H.; Gerla, M. ROAMER: Roadside Units as message routers in VANETs. Ad Hoc Networks 2012, 10, 479–496. [CrossRef]

29. Choi, J.-H.; Han, Y.-H.; Min, S.-G. A Network-Based Seamless Handover Scheme for VANETs. IEEE Access 2018, 6, 56311–56322. [CrossRef]

30. Paridel, K.; Balen, J.; Berbers, Y.; Martinovic, G. VVID: A Delay Tolerant Data Dissemination Architecture for VANETs Using V2V and V2I Communication. In +MOBILITY, International Conference on Mobile Services, Resources, and Users; Xpert Publishing Services: Wilmington, DE, USA, 2012; pp. 151–156.

31. Giang, A.T.; Busson, A.; Lambert, A.; Gruyer, D. Spatial Capacity of IEEE 802.11p-Based VANET: Models, Simulations, and Experimentations. IEEE Trans. Veh. Technol. 2015, 65, 6454–6467. [CrossRef]

32. Arena, F.; Pau, G.; Severino, A. A Review on IEEE 802.11p for Intelligent Transportation Systems. J. Sens. Actuator Netw. 2020, 9, 22. [CrossRef]

33. Network Simulator 3 (ns3). Available online: https://www.nsnam.org (accessed on 17 August 2022).

34. RouteMobilityModel. Available online: https://www.nsnam.org/wiki/RouteMobilityModel (accessed on 17 August 2022).

35. Paier, A.; Tesch, R.; Alonso, A.; Smely, D.; Meckel, P.; Zhou, Y.; Czink, N. Average Downstream Performance of Measured Internet Access. In Proceedings of the 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Taichung, Taiwan, 1–3 July 2013; pp. 37–42. [CrossRef]

36. Mehteroglu, C.; Durmus, Y.; Onur, E. Semantic edge caching and prefetching in 5G. In Proceedings of the 2013 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Orlando, FL, USA, 22–25 October 2013; pp. 461–465. [CrossRef]

37. Ahani, G.; Yuan, D. On Optimal Proactive and Retention-Aware Caching with User Mobility. In Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, USA, 27–30 August 2018; pp. 1–5. [CrossRef]

38. Dong, L.; Ni, Q.; Wu, W.; Huang, C.; Znati, T.; Du, D.Z. A Proactive Reliable Mechanism-Based Vehicular Fog Computing Network. IEEE Internet Things J. 2020, 7, 3427–3440. [CrossRef]

39. Tran, T.X.; Kazemi, F.; Karimi, E.; Pompili, D. Mobeec: Mobility-Aware Energy-Efficient Coded Caching in Cloud Radio Access Networks. In Proceedings of the 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Orlando, FL, USA, 22–25 October 2017; pp. 461–465. [CrossRef]

40. Ahani, G.; Yuan, D. On Optimal Proactive and Retention-Aware Caching with User Mobility. In Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, USA, 27–30 August 2018; pp. 1–5. [CrossRef]

41. Mehteroglu, C.; Durmus, Y.; Onur, E. Semantic edge caching and prefetching in 5G. In Proceedings of the 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2017; pp. 692–695. [CrossRef]

42. Dong, L.; Ni, Q.; Wu, W.; Huang, C.; Znati, T.; Du, D.Z. A Proactive Reliable Mechanism-Based Vehicular Fog Computing Network. IEEE Internet Things J. 2020, 7, 3427–3440. [CrossRef]

43. Mehteroglu, C.; Durmus, Y.; Onur, E. Semantic edge caching and prefetching in 5G. In Proceedings of the 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Orlando, FL, USA, 22–25 October 2017; pp. 461–465. [CrossRef]