CSV: Content Service Offloading System with Vehicular Caching

Yeunwoong Kyung 1 and Taewon Song 2,*

1 Division of Information & Communication Engineering, Kongju National University, Cheonan 31080, Korea
2 Department of Internet of Things, Soonchunhyang University, Asan 31538, Korea
* Correspondence: twsong@sch.ac.kr

Abstract: Vehicular caching (VC) has been considered as a promising technology to provide low end-to-end service latency and reduce the load of networks. However, it is difficult for VC to provide service continuity because of its opportunistic availability according to mobility. To mitigate this problem, we introduce a CSV: Content service offloading System with VC which can opportunistically distribute the load of the content server and support the service continuity. In CSV, the content service can be migrated between fog node (FN) and VC while maintaining the ongoing service without service disruption, which can opportunistically distribute the load of the content server and support the service continuity during migration. To assess the performance of CSV, we develop an analytical model for VC offloading efficiency. Extensive simulation results demonstrate that CSV can reduce the load of the content server compared to the conventional system.

Keywords: content delivery network; vehicular caching; service continuity; service migration

1. Introduction

Recently, video traffic has explosively grown as numerous Internet of Things (IoT) devices consume massive amounts of video content such as higher definition 360-degree videos and augmented/virtual reality services [1]. To provide low end-to-end service latency and reduce the load of networks, contents are shifted from cloud to fog nodes (FNs) co-located with the access network device such as a cellular base station (BS) [2]. However, even though the FNs are promising to deliver the contents to users, the scalability issue can also occur because of limited resources and increasing number of content service users including IoT devices [3]. For instance, if many IoT devices within a similar location request content simultaneously, FN can be overloaded. To mitigate this challenge, content delivery via vehicular caching (VC) is introduced. VC can store the contents, and thus users can receive contents from distributed VC, which can reduce the load of FN. VC can provide content to users only when they are in the vicinity of users and have content that the users request [4,5]. This means that users can opportunistically receive the contents from VC when both conditions are satisfied.

For efficient utilization of VC, numerous works have been conducted to determine the content placement and resource allocation [5,6], provide the collaborative content dissemination [7,8], and consider social and economic attributes [4,9]. In these works, the content delivery within the connection time between vehicles is usually utilized. This means that these works did not consider the service continuity between the connection and disconnection time according to the mobility of VC. On one hand, when VC provides the contents to users, service disruption can unwillingly occur because of its mobility during the service. In other words, the ongoing session between the user and VC can be terminated when VC leaves, which can result in the quality of service (QoS) degradation. On the other hand, when static FN delivers the contents to users, service offloading might be required when VC becomes available to distribute the load of FN. Therefore, how VC can be exploited with service continuity as much as possible considering its mobility should be analyzed to efficiently distribute the load of FN.
Meanwhile, there have been works to optimize the service provisioning based on the users’ mobility. As an initial concept, follow-me-cloud has been introduced to allow cloud services to be migrated to the optimal data center according to the users’ mobility without service disruption [10]. In addition to cloud services, this concept has been extended to FN architecture [11,12]. However, these works have focused on the mobility of users. This means that the mobility of the content source such as VC is not taken into account. Although our previous work introduced a policy at each decision epoch to select VC or FN to provide contents to users [13], it cannot make the best use of VC and does not consider the service continuity for the content service offloading in the continuous time domain.

To offload more content services to VC with service continuity, we introduce the content service offloading system with VC, named CSV. In CSV, the content service can be migrated between FN and VC while maintaining the ongoing service without service disruption. That is, when VC moves close to the user and has the same content that the user requests, the ongoing content service is offloaded from FN to VC. Then, the content service can be provided through VC without affecting the user, which also makes the best use of VC. As the 3GPP standards have made progress in supporting vehicular communications, vehicle-to-everything (V2X) services, and vertical enabler architectures [14,15], CSV can be one of the network scenarios in the future mobile networks (i.e., 6G). The key contribution of this paper is twofold: (1) to the best of our knowledge, this is the first analytical study that considers the service continuity for the content service offloading with VC. Specifically, we develop the analytical models for the VC offloading efficiency considering the service continuity; and (2) based on the simulation works, we demonstrate how much content can be delivered with VC, which can be interpreted as the load distribution effect.

The remainder of this paper is organized as follows. The system description is introduced in Section 2. In Section 3, the performance analysis is conducted. Then, the simulation results and conclusion are provided in Section 4 and Section 5, respectively.

2. System Description

Figure 1 shows the system model of this paper. It is assumed that FN connected to BS is always available whereas VC is only available when it moves close enough to users. In addition, the contents that users request can exist or not in VC according to the popularity of contents and the limited caching capacity of VC. Although there can be a caching capacity constraint for FN, we assume that all the contents for users are cached in FN to focus on the load distribution effect by VC. Considering the central cloud node to request the contents from FN will be one of our future works. Note that users can recognize the connection and disconnection with VC by using periodic signaling messages [16–18]. Moreover, for the service migration, this paper utilizes session migration between FN and VC which transfers the session information such as socket and state information to provide light and fast service continuity [12,19]. Since the session migration is not for the full content synchronization, it can be performed between nodes that have the same content currently being served.

As an example in Figure 1, users 1 and 2 want to download the specific content. In the case of user 1, the session starts at $\tau_{s1}$. At the time, since VC is not available, the content service is provided directly from FN. After $\tau_1$, to utilize VC which becomes available for user 1, the service migration is performed from FN to VC, which results in the period of no change for the downloaded content size, $t_{m1}$. After the migration, the content service is continuously provided to user 1 from VC until the end of the session $\tau_{e1}$ without service disruption. On the other hand, the session of user 2 starts at $\tau_{s2}$ with VC and content service can be provided from VC because of its proximity. The session should be maintained after VC leaves at $\tau_4$. In this case, after the service migration from VC to FN is performed (i.e., $t_{m2}$), user 2 can continuously use the content service from FN until the end of session $\tau_{e2}$. In this way, CSV can opportunistically utilize the VC which can distribute the load of FN while maintaining the ongoing content service without affecting users.
For the tactical analysis, we assume that the contact time between VC and the user (e.g., from $\tau_1$ to $\tau_2$ and from $\tau_3$ to $\tau_4$ in Figure 1) denoted by $t_v$ follows an exponential distribution with mean $1/\mu_v$ as it is widely utilized for VC residence time [20]. In addition, it is assumed that the period when VC is not available (i.e., non-contact time) represented by $t_f$ follows an exponential distribution with mean $1/\mu_f$ as it is usually assumed for the period in a FN (BS)-only area [21,22]. From the start, the session continues during the session duration represented by $t_d$ (i.e., $t_{d1}$ and $t_{d2}$ in Figure 1) which follows an exponential distribution with mean $1/\lambda_d$ [23]. During the session duration, the content service can be provided to the user. The elapsed time from the session start to the migration is represented by $t_{rf}$ (i.e., from the session start with FN and migration to VC) and $t_{rv}$ (i.e., from the session start with VC and migration to FN) as shown in Figure 1. According to the residual life theory [24], the PDF of $t_{rf}$ and $t_{rv}$ can be given by

$$f_{rf}(t) = \mu_f(1 - F_f(t))$$  \hspace{1cm} (1)

and

$$f_{rv}(t) = \mu_v(1 - F_v(t)).$$  \hspace{1cm} (2)

The VC offloading time denoted by $t_{off}$ is defined as the elapsed session duration in the contact time between VC and the user (e.g., from $\tau_1$ to $\tau_2$ and from $\tau_3$ to $\tau_4$ in Figure 1). The VC offloading time can be exploited to provide the content service to users through VC, which consequently can reduce the load on FN. Based on the assumption that the user meets a VC once in session duration, $t_{off}$ is less or equal to $t_v$. In addition, we assume that FN and VC can deliver the contents to users with the data rate of BS (i.e., $d_b$) and that of VC (i.e., $d_v$), respectively.

Let $Q$ denote the total number of contents. Since we assume that the popularity of the content follows a Zipf distribution [5,25], the request probability of content $q$ is given by

$$p_q = \frac{1}{r_q^\epsilon} / \left( \sum_{q=1}^{Q} \frac{1}{r_q^\epsilon} \right)$$  \hspace{1cm} (3)

where $\epsilon$ is the exponent characterizing the distribution and $r_q$ is the popularity ranks of the content $q$. Moreover, due to the limited caching capacity of VC (i.e., $C_v$), the least-recently-used (LRU) is assumed as it is generally utilized in the content caching process for the replacement policy [26].
3. Performance Analysis

In this section, the analytical model of CSV is introduced in order to evaluate the performance of CSV compared to that of the non-migration system. In the non-migration system, service migration is not considered. That is, when the session starts with FN, it maintains with FN during the session duration even though VC becomes available. On the other hand, if the session starts with VC, it terminates when VC leaves and a new session re-starts with FN. Specifically, VC offloading efficiency, \( E \), which is defined as the ratio of the expected amount of downloaded content from VC to the expected total amount of downloaded content during the session period, is derived as an important performance metric.

3.1. Effective VC Offloading Time

To obtain the offloading efficiency by VC of CSV, the expected VC offloading time, \( E[t_{off}] \), can be derived considering five disjoint cases in Figure 2: (a) \( \tau_s \in t_f, t_d < t_{rf} \), (b) \( \tau_s \in t_f, t_{rf} \leq t_d < t_{rf} + t_v \), (c) \( \tau_s \in t_f, t_{rf} + t_v \leq t_d \), (d) \( \tau_s \in t_v, t_d < t_{rv} \), and (e) \( \tau_s \in t_v, t_{rv} \leq t_d \). Let \( E_{Y,C} \) be the partial offloading time in the case \( Y \in \{a, b, c, d, e\} \) for CSV. Consequently, \( E_{Y,C} \) can be derived as follows.

Figure 2. Timing diagrams of CSV.
3.1.1. \(E_{a,C}\)

When \(\tau_s \in t_f, t_d < t_{rf}\), the content service is provided only by FN because the session starts and ends before VC is available. Therefore, \(E_{a,C}\) is simply 0.

3.1.2. \(E_{b,C}\)

When \(\tau_s \in t_f, t_{rf} \leq t_d < t_{rf} + t_v\), the ongoing session is migrated to VC and completed during the VC contact time \(t_v\). As a result, \(t_{off}\) is given by \(t_d - t_{rf} - t_m\). The joint PDF of the \(t_d, t_{rf}\), and \(t_v\) is obtained by \(f_{d,rf,v}(t_d, t_{rf}, t_v)\). Since \(t_d, t_{rf}\), and \(t_v\) are independent, the joint PDF can be the product of each PDF (i.e., \(f_d(t_d)f_{rf}(t_{rf})f_v(t_v)\)). Moreover, since \(\int_{t_d=0}^{t_{rf}} \int_{t_{rf}=0}^{t_v} f_d(t_d)f_{rf}(t_{rf})f_v(t_v)dt_ddt_{rf}dt_v\) is commonly required for \(E_{b,C}, E_{c,C}, E_{d,C}\) and \(E_{e,C}\), we omit the common part for simplicity. Consequently, \(E_{b,C}\) can be defined as

\[
E_{b,C} = \int_{t_d=t_{rf}+t_v}^{t_{rf}+t_d} (t_d - t_{rf} - t_m)f_d(t_d)dt_d.
\] (4)

Note that \(E_{b,C}\) is valid when \(t_d - t_{rf} \geq t_m\). In other words, if \(t_d - t_{rf} < t_m\), \(E_{b,C}\) is assumed to be 0 (i.e., \(t_{off}\) is not effective) due to the long migration time, which can be considered as an offloading failure. The offloading failure will be described in Section 3.3.

3.1.3. \(E_{c,C}\)

When \(\tau_s \in t_f, t_{rf} + t_v \leq t_d\), the ongoing session is migrated to VC and completed after the VC contact time \(t_v\). Consequently, \(t_{off}\) is obtained by \(t_v - t_m\). Therefore, \(E_{c,C}\) can be given by

\[
E_{c,C} = \int_{t_v=t_{rf}+t_d}^{\infty} (t_v - t_m)f_d(t_d)dt_d.
\] (5)

Note that \(E_{c,C}\) is valid when \(t_v > t_m\). In other words, if \(t_v \leq t_m\), \(E_{c,C}\) is assumed to be 0 (i.e., \(t_{off}\) is not effective), which can be considered as an offloading failure and also will be described in Section 3.3.

3.1.4. \(E_{d,C}\)

If \(\tau_s \in t_v, t_d < t_{rv}\), the session starts and ends during the VC contact time, which means that \(t_{off}\) is same to \(t_d\). As a result, \(E_{d,C}\) can be obtained by

\[
E_{d,C} = \int_{t_d=0}^{t_{rv}} t_df_d(t_d)dt_d.
\] (6)

3.1.5. \(E_{e,C}\)

If \(\tau_s \in t_v, t_{rv} \leq t_d, t_{off}\) is same with \(t_{rv}\) because the session starts in the VC contact time and ends after the contact time. Therefore, \(E_{e,C}\) can be defined

\[
E_{e,C} = \int_{t_d=t_{rv}}^{\infty} t_{rv}f_d(t_d)dt_d.
\] (7)

Because the cases from (a) to (e) are disjoint events, the expected VC offloading time of CSV, \(E[t_{off}]\), can be obtained by \(E_{a,C} + E_{b,C} + E_{c,C} + E_{d,C} + E_{e,C}\).

On the other hand, \(E[t_{off}]\) in the non-migration system, can be the same with the sum of \(E_{d,C}\) and \(E_{e,C}\) because it does not consider the service migration when the VC becomes available. Specifically, when the session starts with FN (like cases from (a) to (c) in Figure 2), the session maintains with FN (i.e., no migration to VC) until it completes. Therefore, the offloading by VC in the non-migration system is only valid when the session starts during VC contact time.

To utilize the expected VC offloading time (i.e., \(E[t_{off}]\)) as derived above for both systems, it is required for VC to have the same content as that of the user’s ongoing session.
Note that the content $q$ can be requested by VC with the request probability $p_q$ and then be cached at VC. Therefore, the effective VC offloading time, $E_{T_f}$, can of CSV can be given by $p_q E_{[t_{off}]}$.

### 3.2. VC Offloading Efficiency

The expected amounts of downloaded content during the effective VC offloading time (i.e., $E_{T_f}$) can be calculated as $E_{T_f} d_{b}$ for both CSV and non-migration systems. On the other hand, the expected total amounts of downloaded content during the session duration can be obtained by $E_{T_f} d_{b} + (E[t_d] - E_{T_f})d_b$. Therefore, the VC offloading efficiency can be given by

$$ E_E = \frac{E_{T_f} d_{b}}{E_{T_f} d_{b} + (E[t_d] - E_{T_f})d_b}. \quad (8) $$

Particularly, if $E_{E} = 0$, it means that the content of the user's ongoing session does not exist in VC, or the session starts and ends without VC offloading. On the other hand, if $E_{E} = 1$, it indicates that the session maintains with VC through the entire session duration.

### 3.3. VC Offloading Failure

As described in Section 3.1, when the user tries to offload the service from FN to VC considering two cases (b) and (c) in Figure 2, it can be failed due to the long migration time. Let $F_Y$ be the probability of the VC offloading failure in the case $Y \in \{b, c\}$ for CSV. Consequently, $F_Y$ can be derived as follows.

#### 3.3.1. $F_b$

In case (b), the VC offloading failure can occur when $t_d - t_{rf} < t_m$. Consequently, $F_b$ can be obtained by

$$ F_b = \int_{t_c=0}^{\infty} \int_{t_{rf}=0}^{t_d} \int_{t_d=1}^{t_d} f_d(t_d) f_{rf}(t_{rf}) f_b(t_b) dt_d dt_{rf} dt_b. \quad (9) $$

#### 3.3.2. $F_c$

In case (c), if $t_v \leq t_m$, the VC offloading failure can occur. Therefore, $F_c$ can be defined as

$$ F_c = \int_{t_v=0}^{t_m} \int_{t_{rf}=0}^{\infty} \int_{t_d=1}^{t_d} f_d(t_d) f_{rf}(t_{rf}) f_v(t_v) dt_d dt_{rf} dt_v. \quad (10) $$

Because the cases (b) and (c) are disjoint events, the total probability of the VC offloading failure for CSV, $P_F$, can be obtained by $F_b + F_c$.

### 4. Simulation Results

To verify the analytical results, event-driven simulations based on MATLAB R2018b are performed in Windows 10 with 16 GB RAM. Moreover, arrival times of events are drawn by generating 20,000 random numbers of $t_f$ and $t_d$ and the average values are utilized. As described in Section 3, the simulation results of CSV are compared to those of the non-migration scheme. The default values of $\mu_f$, $\lambda_d$, $d_{b}$, $d_{v}$, $Q$, $t_m$, $r_q$, and $C_v$ are 10, 10, 10 Mbps, 5 Mbps, 50, 0.01, 1, and 10 GB respectively [9,25] as shown in Table 1. In addition, the content size is within the range of [500 MB, 1000 MB] and the exponent of the Zipf distribution (i.e., $\epsilon$) is 0.7 [25]. Figure 3 shows the effect of the average contact time between VC and the user $1/\mu_v$ which is normalized by the non-contact time $1/\mu_f$ for the VC offloading efficiency $E_E$ and effective VC offloading time $E_{T_f}$. In Figure 3a,b, as the VC contact time increases, VC offloading efficiency becomes higher because the expected VC offloading time can increase. From Figure 3a, it can be found that CSV can achieve higher offloading efficiency compared to the non-migration system. This is because CSV efficiently exploits the VC to offload the service when it becomes available thanks to the service migration. In addition, VC offloading efficiency at $1/\lambda_d = 0.1$ is higher than that at
$1/\lambda_d = 0.2$. This is because the increasing amount of downloaded content through the FN is larger than that through the VC when $1/\lambda_d$ increases. This means that the denominator rapidly increases than the numerator in Equation (8) according to the increasing $1/\lambda_d$. In addition, at the same $1/\lambda_d$, the differences in the offloading efficiency between the two schemes increase as $1/\mu_v$ increases. This is because the effect of cases (b) and (c) in Figure 2 on the offloading efficiency is marginal at smaller $1/\mu_v$ and becomes significant at bigger $1/\mu_v$. On the other hand, from Figure 3b, it can be found that the effective VC offloading time becomes higher when $1/\lambda_d$ increases for both systems because the opportunity to offload the service to VC also increases. However, the increasing rate of the non-migration system is much smaller than that of CSV. This means that the cases (d) and (e) in Figure 2 do not have an important effect on the expected VC offloading time according to the change of $1/\lambda_d$. Specifically, Table 2 summarizes the ratio of the partial offloading time for each case (i.e., $E_{b,C}$, $E_{c,C}$, $E_{d,C}$, and $E_{e,C}$ as defined in Section 3) to the total expected offloading time (i.e., $E[\text{off}]$) for CSV. From Table 2, it can be noted that as $1/\lambda_d$ increases, the effect of $E_{d,C}$ and $E_{e,C}$ decreases while that of $E_{a,C}$ and $E_{b,C}$ increases. For example, when $1/\mu_v = 0.6/\mu_f$, the ratio of $E_{a,C}$ to the total expected offloading time accounts for 11.7% and 21.7% for $1/\lambda_d = 0.1$ and $1/\lambda_d = 0.2$, respectively (i.e., approximately 10% increase). On the other hand, the ratio of $E_{d,C}$ to the total expected offloading time becomes 26.5% from 35.3% when $1/\lambda_d$ changes into 0.2 from 0.1 (i.e., approximately 8.8% decrease).

Table 1. Simulation parameters.

| Parameter | Default Value |
|-----------|---------------|
| $\mu_f$   | 10            |
| $\lambda_d$ | 10           |
| $d_b$     | 10 Mbps       |
| $d_v$     | 5 Mbps        |
| $Q$       | 50            |
| $t_m$     | 0.01          |
| $r_q$     | 1             |
| $C_v$     | 10 GB         |
| Content size | [500 MB, 1000 MB] |
| $\epsilon$ | 0.7          |

Table 2. Ratio of the partial offloading time.

| 1/\lambda_d | 1/\mu_v | $E_{b,C}$ | $E_{c,C}$ | $E_{d,C}$ | $E_{e,C}$ |
|-------------|---------|-----------|-----------|-----------|-----------|
| 0.1         | 0.3/\mu_f | 10.1%     | 22.6%     | 34.9%     | 32.4%     |
|             | 0.6/\mu_f | 11.7%     | 22.6%     | 35.3%     | 30.3%     |
|             | 0.9/\mu_f | 12.5%     | 20.4%     | 37.4%     | 29.7%     |
| 0.2         | 0.3/\mu_f | 19.1%     | 25.9%     | 28.0%     | 26.9%     |
|             | 0.6/\mu_f | 21.7%     | 27.3%     | 26.5%     | 24.5%     |
|             | 0.9/\mu_f | 22.8%     | 26.4%     | 26.8%     | 23.9%     |

Table 3 shows the effect of the service migration time $t_m$ to the VC offloading efficiency $E_E$ when $1/\mu_v$ changes from $0.5/\mu_f$ to $1/\mu_f$. From Table 3, it can be found that the VC offloading efficiency decreases according to the increasing migration time because the service cannot be provided during the migration time. In addition, Table 4 shows the VC offloading failure probability according to the service migration time. The VC offloading failure probability increases as the migration time increases, which can be considered a risk to try to offload the service that exists. However, as the service migration has become efficient and fast [12], it can be expected that the effect of the migration time will become
trivial. In addition, since the service operator can predict the migration time based on the network status [12], service migration can be exploited at low offloading failure probability.

![Graph](image_url)

**Figure 3.** Effect of the average contact time between VC and the user. (a): VC Offloading Efficiency, (b): Effective VC Offloading Time.

**Table 3.** VC offloading efficiency ($E_E$) according to $t_m$.

| $1/\mu_v$ | $t_m$ | 0.01   | 0.02   | 0.03   | 0.04   | 0.05   |
|-----------|-------|--------|--------|--------|--------|--------|
| 0.5/\mu_f |       | 0.159  | 0.141  | 0.127  | 0.114  | 0.104  |
| 1/\mu_f   |       | 0.257  | 0.240  | 0.225  | 0.213  | 0.202  |

Figure 4 illustrates the effect of the popularity rank $r_q$ of the content on the VC offloading efficiency $E_E$. We assume that the first rank is the lowest rank. For both systems,
the VC offloading efficiency decreases as the popularity rank increases. This is because the probability that the VC and user have the same content decreases when the popularity rank increases. In addition, the VC offloading efficiency increases according to the increasing exponent of the Zipf distribution (i.e., $\epsilon$). The reason is that the request probability of high-rank contents becomes higher as the exponent increases. Specifically, the sum of the request probabilities of the top ten contents becomes approximately 13% higher from $\epsilon = 0.7$ to $\epsilon = 1$.

Table 4. VC offloading failure probability ($P_F$) according to $t_m$.

| $1/\mu_v$ | $t_m$ | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|-----------|-------|------|------|------|------|------|
| $0.5/\mu_f$ | 0.33% | 1.64% | 2.93% | 4.21% | 5.46% |
| $1/\mu_f$ | 0.05% | 0.64% | 1.45% | 2.25% | 3.05% |

Figure 4. Effect of the popularity rank of the content on the VC offloading efficiency.

Table 5 shows the effect of the total number of contents $Q$ to the VC offloading efficiency $E_E$. For both systems, the VC offloading efficiency decreases as the total number of contents increases. The reason is that the probability that VC has the same content with user’s request becomes reduced when the total number of contents increases. Furthermore, the difference of the VC offloading efficiency between both schemes decreases according to the increasing total number of contents. This is because the decreasing rate of the Zipf distribution becomes reduced according to the total number of contents.

Table 5. VC offloading efficiency ($E_E$) according to the total number of contents ($Q$).

| $Q$ | CSV | 10 | 30 | 50 | 70 | 90 |
|-----|-----|----|----|----|----|----|
| CSV | 0.244 | 0.179 | 0.159 | 0.148 | 0.141 |
| None | 0.160 | 0.117 | 0.104 | 0.097 | 0.092 |

Figure 5 illustrates the effect of the caching capacity of VC $C_v$ on the VC offloading efficiency $E_E$. For both systems, the VC offloading efficiency increases as the caching capacity of VC increases. This is because the probability that the VC has the same content with user’s request increases when the caching capacity increases. In addition, the increasing rate of CSV is higher than that of the non-migration scheme because CSV has more opportunity to
utilize VC compared to the non-migration scheme. Specifically, the increasing rates of CSV and non-migration schemes are about 0.016 and 0.005, respectively, when $1/\lambda_d$ is 0.1 and $C_v$ changes from 5 to 14.

Figure 5. Effect of the caching capacity of VC ($C_v$) on the VC offloading efficiency ($E_E$).

5. Conclusions

In this paper, we introduced a CSV where the content service is offloaded to the VC to reduce the load of the FN and provide service continuity. To analyze the performance of CSV, we developed the analytical model of the VC offloading efficiency. Simulation results show that CSV can improve the VC offloading efficiency compared to the non-migration system and the offloading efficiency is affected by the session duration, VC contact time, migration time, and the popularity rank of the content. In our future work, we will validate the CSV by means of the emulation-based feasibility test.

Author Contributions: Conceptualization, Y.K.; methodology, Y.K.; validation, Y.K.; formal analysis, Y.K. and T.S.; investigation, T.S.; resources, Y.K. and T.S.; data curation, Y.K. and T.S.; writing—original draft preparation, Y.K.; writing—review and editing, Y.K. and T.S.; visualization, Y.K.; supervision, T.S.; funding acquisition, T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Soonchunhyang University Research Fund (No. 20210447).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work was supported in part by the Soonchunhyang University Research Fund (No. 20210447) and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No.2020R1G1A1100493).

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

References

1. Liu, Y.; Peng, M.; Shou, G.; Chen, Y.; Chen, S. Toward edge intelligence: Multiaccess edge computing for 5G and internet of things. *IEEE Internet Things J.*, 2020, 7, 672–6747. [CrossRef]

2. Fan, Q.; Ansari, N. Towards workload balancing in fog computing empowered IoT. *IEEE Trans. Netw. Sci. Eng.* 2020, 7, 253–262. [CrossRef]
3. Wang, X.; Ning, Z.; Wang, L. Offloading in internet of vehicles: A fog-enabled real-time traffic management system. *IEEE Trans. Ind. Inform.* 2018, 14, 4568–4578. [CrossRef]

4. Ruan, Y.; Joe-Wong, C. On the Economic Value of Mobile Caching. In Proceedings of the IEEE INFOCOM, Toronto, ON, Canada, 6–9 July 2020.

5. Su, Z.; Hui, Y.; Xu, Q.; Yang, T.; Liu, J.; Jia, Y. An edge caching scheme to distribute content in vehicular networks. *IEEE Trans. Veh. Technol.* 2018, 67, 5346–5356. [CrossRef]

6. Wu, H.; Zhang, J.; Cai, Z.; Liu, F.; Li, Y.; Liu, A. Toward energy-aware caching for intelligent connected vehicles. *IEEE Internet Things J.* 2020, 7, 8157–8166. [CrossRef]

7. Huang, X.; Xu, K.; Chen, Q.; Zhang, J. Delay-aware caching in internet of vehicles networks. *IEEE Internet Things J.* 2021, 8, 10911–10921. [CrossRef]

8. Zhang, Y.; Li, C.; Luan, T. H.; Fu, Y.; Shi, W.; Zhu, L. A mobility-aware vehicular caching scheme in content centric networks: Model and optimization. *IEEE Trans. Veh. Technol.* 2019, 68, 3100–3112. [CrossRef]

9. Yao, L.; Wang, Y.; Wang, X.; Guowei, W.U. Cooperative caching in vehicular content centric network based on social attributes and mobility. *IEEE Trans. Mob. Comput.* 2021, 20, 391–402. [CrossRef]

10. Taleb, T.; Ksentini, A. Follow me cloud: Interworking federated Clouds and distributed mobile networks. *IEEE Netw.* 2013, 27, 12–19. [CrossRef]

11. Bao, W.; Yuan, D.; Yang, Z.; Wang, S.; Li, W.; Zhou, B.B.; Zomaya, A.Y. Follow me fog: Toward seamless handover timing schemes in a fog computing environment. *IEEE Commun. Mag.* 2017, 55, 72–78. [CrossRef]

12. Addad, R.A.; Dutra, D.L.C.; Bagaa, M.; Taleb, T.; Flinck, H. Fast service migration in 5G trends and scenarios. *IEEE Netw.* 2020, 34, 92–98. [CrossRef]

13. Huang, X.; Xu, K.; Chen, Q.; Zhang, J. Delay-aware caching in internet of vehicles networks. *IEEE Internet Things J.* 2021, 8, 10911–10921. [CrossRef]

14. Michalopoulou, M.; Kolios, P.; Panayiotou, C.; Ellinas, G. An Event-triggering Approach for Bus Tracking based on Multimodel Mobility Prediction. In Proceedings of the IEEE Intelligent Transportation System Conference, Auckland, New Zealand, 27–30 October 2019.

15. Lee, J. H.; Lee, G.; Pack, S. Performance optimization of delayed WiFi offloading in heterogeneous networks. *IEEE Trans. Veh. Technol.* 2017, 66, 9436–9447. [CrossRef]

16. Lim, H.T.; Kim, Y.; Pack, S.; Kang, C.H. Call admission control with heterogeneous mobile stations in cellular/WLAN interworking systems. *EURASIP J. Wirel. Commun. Netw.* 2011, 91. doi: 10.1186/1687-1499-2011-91 [CrossRef]

17. Ross, S.M. *Stochastic Processes*; John Wiley: New York, NY, USA, 1996.