Article

Cost-Effective Resource Sharing in an Internet of Vehicles-Employed Mobile Edge Computing Environment

在车载移动边缘计算环境网络中的成本有效的资源共享

Tri D. T. Nguyen , **Tien-Dung Nguyen, Van Dung Nguyen, Xuan-Qui Pham and Eui-Nam Huh** *

Department of Computer Science and Engineering, Kyung Hee University, Yogin-si 17104, Korea; tringuyendt@khu.ac.kr (T.D.T.N.); ntiendung@khu.ac.kr (T.-D.N.); ngvandung85@khu.ac.kr (V.D.N.); pxuanqui@khu.ac.kr (X.-Q.P.)

* Correspondence: johnhuh@khu.ac.kr; Tel.: +82-031-201-2454

Abstract: By bringing the computation and storage resources close proximity [接近] to the mobile network edge, mobile edge computing (MEC) is a key enabling technology for satisfying the Internet of Vehicles (IoV) infotainment applications' requirements, e.g., video streaming service (VSA). However, the explosive [爆炸性的] growth of mobile video traffic brings challenges for video streaming providers (VSPs). One known issue is that a huge traffic burden on the vehicular network leads to increasing VSP costs for providing VSA to mobile users (i.e., autonomous vehicles). To address this issue, an efficient resource sharing scheme between underutilized vehicular resources is a promising solution to reduce the cost of serving VSA in the vehicular network. Therefore, we propose a new VSA model based on the lower cost of obtaining data from vehicles and then minimize the VSP's cost. By using existing data resources from nearby vehicles, our proposal can reduce the cost of providing video service to mobile users. Specifically, we formulate our problem as mixed integer nonlinear programming (MINP) in order to calculate the total payment of the VSP. In addition, we introduce an incentive mechanism to encourage users to rent its resources. Our solution represents a strategy to optimize the VSP serving cost under the quality of service (QoS) requirements. Simulation results demonstrate that our proposed mechanism is possible to achieve up to 21% and 11% cost-savings in terms of the request arrival rate and vehicle speed, in comparison with other existing schemes, respectively.

Keywords: cost-effective; Internet of Vehicles; volunteer computing; mobile edge computing; vehicular network

(问题的出现) (解决方案)

1. Introduction

With the rapid development of the Internet of Vehicles (IoV) and automated driving technologies, autonomous cars are one of the most advanced technologies in recent years. It is supposed that smart vehicles will provide safe driving experiences and content-rich multimedia services [1–3]. According to a recent report [4], the global passenger cars market will have a total of 97 million vehicles being produced. Most vehicles will be equipped with wireless transceivers to exchange information wirelessly by 2020. With this ability, the next generation of connected cars can collaborate to acquire surrounding data information.Therefore, it can adapt in a smart environment (SE) [5,6]. Moreover, developing applications for autonomous vehicles have emerged in recent years. They can be classified into safety applications (i.e., collision avoidance, accident prevention), and non-safety applications (i.e., map, infotainment), according to a mobile traffic. Each application is managed in different channels depending on the types of vehicular networking communication [7]. Some applications developing into safe operation must ensure the accuracy of obstacle detection or object recognition [8,9]. Other applications focus on how to enhance the driving experiences to passengers on the road.

For example, passengers can easily use mobile devices or in-vehicle equipment to access services such as 3D navigation map or in-vehicle infotainment in a convenient way than before [10–12]. Moreover, video streaming is the top launched application for in-vehicle infotainment [13,14], especially during long trips. For instance, AT&T is working with Ford to develop services for connected cars, including video streaming services. They emphasize that such services will play a significant role in future connected cars [15]. Meanwhile, Volvo, one of the largest car manufacturers, has revealed a partnership with Ericsson to develop intelligent media streaming for smart cars with the aid of fifth-generation (5G) communications [16]. This enables drivers to relax and enjoy free time watching their favorite TV shows in their fully autonomous car on their daily commute. The rapid growth of such mobile video traffic brings new challenges for video streaming providers (VSPs). Specifically, it causes a tremendous traffic burden on the vehicular network. Consequently, this leads to the increasing VSP's cost of delivering video content to vehicles and also degrades the quality of service (QoS) requirement. Therefore, efficiently delivering video content to mobile users while guaranteeing revenue for service providers becomes an important issue.

问题的出现：
爆发式的移动视频业务的增长给车辆网络造成严重负担 VSP增加了成本还降低了QoS要求
问题：保证VSP收入的同时 提高服务质量

Fortunately, mobile edge computing (MEC) is proposed to address this issue [17,18]. The key concept of MEC is to bring the computing and storage resources close to the end-users by pairing small servers with base stations (BSs) at the mobile network edge. As such, application services can be deployed in these edge servers in a distributed manner across the network. Consequently, the VSP can bring popular video content into the MEC servers on edge networks in advance and deliver this content to mobile users directly through wireless communications. This alleviates the traffic burden of congested vehicular networks and delivers video content at a lower cost. Moreover, MEC is supposed to be more efficient than the traditional centralized cloud computing paradigm [19]. Since MEC is a distributed paradigm, it mitigates the effect of failures at a centralized point. Due to the proximity to end-users, MEC can provide low latency services. Additionally, it can also determine the location of connected devices, thus improving the performance of location-based services [20]. Hence, employing MEC paradigm opens the way for better delivery of service to end-users.

MEC的核心概念

MEC带来的好处

Some research efforts have been made in recent years to design efficient service delivery mechanisms in vehicular networks employed in the MEC environment [21,22]. Most of the efforts have focused on allocating the MEC computation resource optimally to minimize the latency and improve the use of resources. Alternatively, some efforts have been made to find an optimal strategy to maximize the profits gained by the MNO (mobile network operation) and the VSP from resource leasing and renting. Such approaches only consider obtaining data stored on the MEC server nearby requests, which sometimes cannot be accomplished due to the limit of the MEC's storage resource. Hence, the quality of VSA service decreases. To tackle this, the VSP needs higher hardware and network capacity requirements, so the cost of delivering video content increases. Therefore, the research presented in [23,24] have studied solutions that allow parked vehicles to combine with MEC servers on edge networks, thus enlarging the storage resource for improving the performance of delivery services in vehicular networks. However, the main limitation of this approach is that it cannot fully use a cluster of parked cars as a supplemental resource when their power is turned off.

Volunteer computing has emerged as a solution to facilitate the above issue [25]. In fact, smart vehicles are equipped with a variety of resources (e.g., storage, computing) [26] and can be regarded as "computers-on-wheels," allowing them to benefit from various services including video streaming service (VSA). Moreover, according to [27], vehicles consume a large amount of content while simultaneously providing content. Hence, vehicles act as rich data volunteers. However, these resources are not exploited efficiently. For example, many vehicular computing resources are kept idle for long periods (e.g., in congested areas), and some vehicles store a large number of videos (i.e., TV series, frequently used content) while other vehicles are requesting similar content. In this context, volunteer computing allows the MEC framework to exploit vehicle resources nearby to alleviate congested wireless communications in vehicular networks, and hence minimizing the VSP's cost for delivering the content to mobile users.

Hence, in this paper, we propose a video streaming mechanism in the vehicular network for a volunteering computing data service so that it can carefully obtain data from vehicles to optimize the VSP's payment. We consider a simple representative model where one VSP offers video on demand (VoD) service to smart vehicles. Specifically, a user requests video content following Zipf popularity, and the requested video content can be transmitted to the target user in three different ways: (i) from the MEC cache in the case of similar requests having been processed before in the same BS, (ii) from the MEC cache but content was received from moving vehicles, and (iii) from the remote content server in the cloud. Overall, our main contributions are summarized as follows:

- We formulate a cost-effective resource sharing for optimizing the VSP's cost when providing the VSA to mobile users (i.e., smart vehicles) under an MINP optimization problem. The solutions show that the VSP will optimally obtain content from the moving vehicle or cloud to deliver video to the target user while guaranteeing the QoS requirement.
- An incentive method is investigated to encourage moving vehicles to share its data with BSs. This leads to a win-win solution for the VSP and users.
- We also analyze the conditions for achieving the optimal value under the QoS requirements using geometry stochastic theory.
- We evaluate the proposed mechanism on the real-world taxi and limousine trips data traces.

The remaining content of this paper is categorized as follows. Section 2 discusses some recent studies for cost-effective resource sharing in vehicular networks employed in the MEC environment. The overall system architecture and VSA employing vehicles as data volunteers are described in Section 3. The average transmission rate analysis is provided in Section 4. Cost-effective video transmission based on volunteer nodes is presented in Section 5, followed by the performance analysis in Section 6. Finally, conclusions and suggestions for future work are discussed in the concluding Section 7.

2. Related Work

Some studies have been performed to leverage vehicle resources as major providers of roadside services in various domains and interests. In [28], the authors considered how to coordinate computation resources in both the cloud and smart vehicles to solve the task scheduling problem in the Vehicular Cloud. The objective is aimed at optimizing the schedule cost, which is the cost of renting the resources. Vehicle resources are also exploited for off-loading schemes, such as in [29]. The authors studied two off-loading schemes based on V2I and V2V communications with the aid of the MEC framework, then compared the total savings in terms of the latency and transmission cost between them. The results from their experiment showed that combining two modes achieved better performance, in which tasks are flexibility off-loaded through direct uploading V2I or relay mode V2V. The work in [22] replicated the video chunks in vehicle storage space, and from that, used V2V communication to fetch streaming content before it finished playing, thus minimizing the amount of data downloaded from backhaul links. This was motivated by the fact that the more contents spread over the vehicular network, the higher possibility that these contents are downloaded from such vehicles on time.

Some studies employed parked vehicles to improve the performance of vehicular networks. In [23], the authors proposed a new method, namely, ParkCast. This enlarges the total available resources of the vehicular network by employing the idle resources in parked vehicles. In [24], the authors designed an algorithm to enhance the performance of content distribution in the vehicular network. Specifically, they developed a caching mechanism based on the mobility of vehicles and the analysis of access pattern in a parking area. In [30], an extension of the work in [24], a novel content delivery scheme was developed, where both RSU and parked vehicles can cooperatively provide content for moving vehicles to earn a profit. Moving vehicles can make decisions optimally to jointly receive the requested content from cars in a parking area and RSU. To model the interactions among those entities, a Stackelberg game is used. An equilibrium solution shows maximum utility with

the optimal price during content delivery. However, one of the main limitations is that it depends on clusters of parked vehicles to assist with content delivery. On a highway or in a rural area, these methods might not be an efficient solution.

Incentive mechanisms to motivate users to share resources with each other also have been studied in some previous works. In [31], the authors presented a credit-based incentive mechanism for MP2P (mobile peer-to-peer) streaming, with the aim of maximizing both the helper's revenue and the help receiver's utility. The authors in [31] considered a bandwidth allocation problem for uploading and downloading, while multi-bit rate-encoded streaming has not been mentioned. In contrast, in [32], the authors proposed a truthful auction theory, with the aim of maximizing social welfare by joint allocating radio resource and controlling the bit rate. Additionally, an incentive strategy has been also mentioned for improving the truthfulness and efficiency of an auction mechanism.

However, none of these works considers how to minimize the payment cost of providing a video streaming service using mobile users as volunteer nodes. In addition, the effect of wireless transmission links on the successful transmission of data between mobile users and infrastructure must be analyzed to ensure the QoS. Furthermore, incentive methods, which are offered to the mobile users to encourage renting their resources to the VSP, also was not mentioned in those works. Hence, all of these issues are solved in this paper.

3. System Model

3.1. Network Model

We consider a vehicular network for a unidirectional road, which consists of multi-edge computing nodes deployed at the edge cloud layer. Each edge computing node includes a BS connect to the MEC server. The list of BSs are denoted as $\mathbb{S} = \{1, 2, ..., S\}$. The regional cloud at the middle-tier coordinates the information exchange between edge computing nodes. Additionally, for better support vehicle mobility, the control and data planes are separated, whereby the regional cloud monitors the message transmission in centralized management. Meanwhile, the edge layer primarily provides data services. On the top is a centralized data center (core cloud), where the VSP deployed its service, as shown in Figure 1. Generally, a core cloud has the biggest scale, a regional cloud is smaller than a core cloud, and an edge cloud has the lowest scale [33]. While a core cloud has undisputed advantages of computing capabilities compared to the edge cloud, the networking latency accessing edge services is remarkably decreased due to the shorter distance. By this, the VSP tends to negotiate with IPs (infrastructure providers) to deploy their service at the BS in the edge cloud layer, bringing their service in close proximity to mobile vehicles. The distance between any two adjacent BSs is denoted as L. The transmission range of each BS is the coverage radius, denoted as $L/2$. The road can be divided into several segments with length L. A moving vehicle in a given segment can only access the BS deployed in the corresponding segment through V2I communication. It acts not only as a service's user but also a volunteer node, which provides data resources for the VSA.

The BSs in the same region are able to communicate with each other through regional cloud node. When overloading occurs at the edge cloud, a part of the workload is able to transfer to the upper layer (core cloud) [29,34]. Vehicles are distributed based on a Poisson model on the road [35]. The number of vehicles per unit distance (km^2) is denoted as λ, namely, the traffic density. The distance between vehicles on the road follows an exponential distribution. Table 1 outlines a list of the main notations to be used throughout this paper.
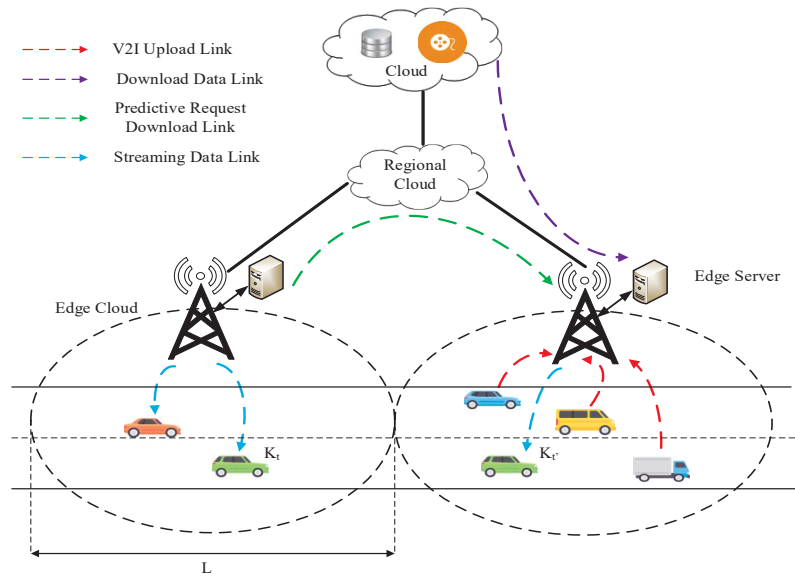
Figure 1. Video streaming service in a distributed cloud-based vehicular network.

Table 1. Table of notations.

Symbol	Definition
$\mathbb{S}$	Number of BSs on the road
L	Distance between two adjacent BSs
H_f	Number of vehicles containing requested file f in its storage resource
$\mathbb{F}$	Set of files f that VUs can request
λ	Number of vehicles per km^2
q_f	Popularity of a file f
T_h^i	Total remaining time a vehicle h travels through BS-i
$T_{f,max}$	Total time a BS needs to obtain file f before the requester arrives at this BS
Q_C	Content hit rate at a specific BS-i
ρ	Proportion of VUs in the ongoing VSA session
$P_{f,i}$	Probability the requested file f is sent to BS-i
$C(t)$	Cost function, which calculates the total fee the VSP has to pay for a VU-h to obtain its data within the delay t
D_f	Total size of requested file f
t	Time delay for uploading data from a VU-h to a BS-i
R_h	Data rate received by a VU-h
W_h	Bandwidth allocates to a VU-h
v_h	Received SINR
W_S	Total bandwidth of a BS-i
N_{VR}	Random variable indicating the number of resident VUs associated with BS-i
P_t	Power transmission of BS-i
d_{hi}	Distance from VU-h to BS-i
σ^2	Gaussian noise
I	Inter-cell interference
$x_{h,f}$	Decision variable of the BS indicating the selection vehicle h to obtain file f
$d_{h,f}$	Portion of data f if a VU-h shares with the BS

3.2. Traffic Model

We consider the achievable data rate for communication between a moving vehicle and its connected BS-i. A vehicle can share its contents with BS-i through V2I communication. The collection of video content that vehicles can access in the system is denoted as $\mathbb{F} = \{1, 2, ..., f, ..., F\}$, where F is the number of files. The different popularity of each file is considered to follow Zipf distribution [36], given by

$$q_f = \frac{\frac{1}{f^\psi}}{\sum_{h=1}^{F} \frac{1}{h^\psi}} \tag{1}$$

where $\psi \geq 0$ indicates the skew parameter. Generally, a larger ψ implies more concentrated file requests. Moreover, the sequence of the files are assumed to be arranged in descending order of popularity in general, which means $q_i \geq q_j, \forall i, j \in \mathbb{F}$, and $i \leq j$. Each vehicle kept the full content f in its storage if f has been requested before.

3.3. Video Transmission Process

The BS can directly stream video to a vehicle if the video content has already cached in the MEC's storage. However, because of the vehicle's mobility, full video transmission might not be accomplished in only one BS. It may need several BSs along the vehicle's trajectory, depending on the data size, vehicle velocity, and data rate of the wireless network. Each edge cloud layer has to collect a vehicle's information and transmit the processed information (e.g., proper route) to moving vehicles. For example, when the vehicle starts to request the desired video, it will send information, including speed, location, video ID, etc., to its connected BS. The BS collects this information to transfer to the upper layer regional cloud. Because the regional cloud has large computing resources, it can execute various route selection algorithms based on real-time traffic acquisition as discussed in [37], or historical vehicles trajectories as discussed in [38]. Since the number of BSs along the vehicle road is known by the regional cloud, it can plan which part of the requested video is handled by which BS along the road. This processed information then is delivered to each BSs along the vehicle's road. Furthermore, considering public transportations (i.e., airport taxi, bus) or commuter transportation, the schedule timetable might be fixed, thus the total remaining time travelling in each BS of a vehicle is also estimated efficiently by the regional cloud (denoted as T_h^i). To manage the connectivity between moving vehicles and BSs, LTE standard for vehicular networking is considered in our system. This is one of the most viable vehicular communication technology [7].

After receiving control messages from the region cloud, each BS starts finding demand content for serving requests. The BS can fetch certain replicas of the content (with size denoted as D_f) from the remote content server, or broadcast a message to its resident vehicles to check whether it can obtain the demanded content from these vehicles. Otherwise, if a copy of the requested data is perfected in the BS cache in advanced, then the BS is able to respond to the requested vehicle directly. Broadcast messages usually contain small bytes, so they will not cause congestions in cellular networks. No matter how the request file f is obtained, with the data size set as D_f, the BS has to ensure that f must be stored in the MEC server before the requester arrives at its coverage so as to guarantee QoS of the VSA.

The MEC server computation capacity can perform transcoding of a video into different bit rate versions to satisfy a user request [39]. However, we do not consider this ability in our current work. Instead, we leverage the MEC storage capacity to cache popular videos to reduce the video transmission delay. Therefore, we consider each BS as keeping the most popular files in its cache. Denoting C as the cache size of each BS, the content hit rate can be given as

$$Q_C = \sum_{f=1}^{f=C} q_f \tag{2}$$

Given λ as the traffic density in terms of vehicles per unit distance (km^2), suppose that there is ρ portion of vehicle users (VUs) per unit distance use the VSA in the system. The remaining VUs can share their cached content with the BS if requested. Based on two possible cases, cached-hit or cached-missed, requests to the BS are divided into two types. One type is the requested contents that are available in the cache called cache-hit VUs. The remaining type is the requested contents that are not satisfied by

the BS cache called cache-miss VUs. Hence, the density of content-hit VUs and content-miss VUs are defined as $\rho Q_C \lambda$ and $\rho (1 - Q_C) \lambda$, respectively, so $\rho Q_C + \rho (1 - Q_C) \lambda = \rho \lambda$.

Due to the distribution of vehicles on the road following the Poisson distribution, the arrival of workloads (i.e., video requests) to a BS is the Poison process [40,41]. We model the arrival and processing of tasks on the BS as an M/M/1 queue. In this queueing model, the average time a request spends in the system is defined as $\frac{1}{\varpi - \bar{\lambda}}$. This comprises the average waiting time in a queue and the processing time of that request. We define ϖ as the number of requests per minute that can be served by BS-*i*, namely, service rate. Meanwhile, $\bar{\lambda}$ is the average rate of requests arriving at a BS and can be given by

$$\bar{\lambda} = \sum_{f=1}^{f=F} P_{f,i} \rho \lambda \tag{3}$$

where $P_{f,i}$ is defined as the probability of VUs requesting file *f* that arrives at the BS-*i*. Please note that the service rate ϖ is greater than the arrival rate $\bar{\lambda}$ for stability [42]. From that, the number of arrival requests that the BS should consider in order to optimize the service cost in the content-miss case is given as

$$\left(1 - Q_{C_i}\right) \sum_{f=1}^{f=F} P_{f,i} \rho \lambda \tag{4}$$

3.4. Cost Model

Our proposed method cannot succeed without participating VUs sharing their vehicular resources. Hence, an incentive method must be provided, for example, monetary returns, to encourage the willingness of VUs to participate. In this way, a VU-*h* can earn benefits from sharing its resources, while VSPs can reduce its service cost by acquiring available data in nearby vehicles rather than in the core cloud far away. We define a cost function $C(t)$ as a monotonically non-increasing function of data uploading time *t*. This function reveals the price that VSP rewards VUs if they obtain the required data within the delay *t* from VUs. Inspired by the work in [43], the cost function is defined as:

$$C(t) = p - bt^{\gamma} \tag{5}$$

where *p* is the original charge for obtaining data in the core cloud and *b* is a positive constant integer, while $t = \frac{D_f}{R_h}$ is defined as the total delay to upload the file *f* from the vehicle to the BS, and R_h denotes the data rate. The curve represents the price the BS pays for VUs when the delay time keeps increasing. For example, as shown in Figure 2, with delay t_1, the VSP pays p_1 dollars for a VU. Without receiving the same content from nearby vehicles, the VSP must pay *p* dollars for downloading data in the remote server. This shows that the VSP is saving $(p - p_1)$ dollars if they acquire data from VUs. However, the VSP will not obtain data from VUs if the delay exceed the bound t_{bound}. This result is reasonable because the cost of using data in the origin server (core cloud) is the sum of the costs of storage and processing, which is strictly higher than the cost of the smallest amount of capacity that might be accessed through a volunteer node [44].

Otherwise, we can see that the vehicle can upload a file successfully depending on the rate that it is received from a BS. However, this variable is changed over time depending on the network status. Hence, the VSP needs to approximate this value. This analysis process is discussed in the following section.
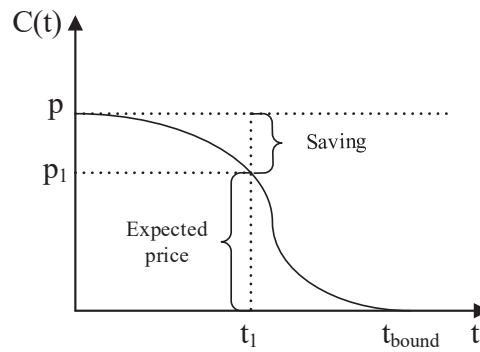
Figure 2. Cost function of the VSP.

3.5. Wireless Transmission Model

For each moving vehicles VU-h in the system, the data rate is achieved from its nearest BS-i as in [45].

$$R_h = W_h \log_2 (1 + v_h) \tag{6}$$

where W_h is defined as the bandwidth BS-i assigns for a specific VU-h. Meanwhile, v_h indicates the SINR value. The BS shared the bandwidth equally with its associated VUs, as discussed in [45], and is defined by

$$W_h = \frac{W_S}{1 + N_{VR}} \tag{7}$$

where N_{VR} is a random variable indicating the number of resident VUs communicating with BS-i except VU-h. The received SINR is given by

$$v_h = 1 + \frac{P_t d_{hi}^{-\alpha}}{\sigma^2 + I} \tag{8}$$

where P_t is the power transmission of BS-i, d_{hi} is the distance from VU-h to BS-i, and supposing that it follows a uniform distribution, a power α stands for the path loss factor. Meanwhile, σ^2 is the Gaussian noise and I represents interference among cells on the road.

4. Average Transmission Rate Analysis

The cost of a VSP when downloading data from the vehicle's storage also depends on the transmission rate R_h. We analyze the average transmission rate based on the result of [45]. The average achievable rate is given by

$$
\begin{aligned}
\bar{R}_h &= E_{\{N_{VR}, d_{hi}\}} \left[\frac{W_S}{1 + N_{VR}} \log_2 \left(1 + \frac{P_t d_{hi}^{-\alpha}}{\sigma^2 + I} \right) \right] \\
&= \int_{d=0}^{\infty} \sum_{n=0}^{\infty} \left[\frac{W_S}{1 + N_{VR}} \log_2 \left(1 + \frac{P_t d_{hi}^{-\alpha}}{\sigma^2 + I} \right) \right] \\
&\qquad . Pr \{N_{VR} = n, d_{hi} = d\} \, dd
\end{aligned}
\tag{9}
$$

As N_{VR} and d_{hi} are independent random variables, we have

$$Pr \{N_{VR} = n, d_{hi} = d\} = p_{N_{VR}} (n) \, p_{d_{hi}} (d) \tag{10}$$

where $p_{N_{VR}}(n)$ and $p_{d_{hi}}(d)$ are the probability mass function of N_{VR} and d_{hi}, respectively. Based our traffic model, the geographical distribution of vehicles follows Poisson distribution model. Hence, N_{VR} is the Poisson distribution of parameter λL under one segment on the road

$$p_{N_{VR}}(n) = \frac{(\lambda L)^n}{n!} e^{-\lambda L} \tag{11}$$

Additionally, d_{hi} follows uniform distribution in a range $\left[0, \frac{L}{2}\right]$ based on above traffic model, thus its probability mass function is given by

$$p_{d_{hi}}(d) = \frac{2}{L} \tag{12}$$

Based on geometry stochastic theory, we can derive the lower bound of $\bar{R}_h$ as follows:

$$\tilde{R}_h = \frac{W_S}{\lambda L} \left(1 - e^{-\lambda L}\right) \left[log_2 \left(\frac{P_t}{\sigma^2 + I} \left(\frac{L}{2}\right)^{-\alpha} \right) + \frac{\alpha}{\ln 2} \right] \leq \bar{R}_h \tag{13}$$

This lower bound $\tilde{R}_h$ is asymptotic to the average achievable rate $\bar{R}_h$ in the high SINR region when

$$\lim_{\frac{\sigma^2 + I}{P_t} \to 0} \tilde{R}_h = \bar{R}_h \tag{14}$$

Proof. Refer to a Section III in [45].

5. Cost-Effective Video Transmission Based on Vehicles as Resources

When a request $f \subset F$ arrives at BS-*i*, the BS decides how to provide streaming service as follows:

1. If f exists in the BS-*i* cache, it can be served directly from the cache. Then, the VSP's serving cost is defined as 0.
2. If there is no VUs inside BS-*i* for obtaining f, or if the number of VUs does not satisfy condition (22), the BS downloads f from the remote server with cost p (per unit).
3. Otherwise, the BS downloads f from VUs in its coverage.

To formulate the optimization problem in case BS-*i* obtains file f from its volunteer nodes, we introduce a binary variable $x_{h,f} \in \{0, 1\}$, which denotes the decision variable of BS-*i*. $x_{h,f} = 1$ when BS-*i* chooses VU-*h* off-loaded data, otherwise $x_{h,f} = 0$. Meanwhile, $d_{h,f}$ is a portion of data that VU-*h* shares with BS-*i*, and $t_{h,f}$ is the total time to successfully upload $d_{h,f}$ from a VU-*h* to BS-*i*, defined as

$$t_{h,f} = \frac{d_{h,f}}{\bar{R}_h} \tag{15}$$

In addition, $T_{f,max}$ is the time BS-*i* must obtain the requested content f before the requester arrives at its coverage.

Cost-effective video transmission aims to minimize the VSP's cost while meeting the quality of service of users, which can be formulated as follows:

$$\min_{x,d} \quad \sum_{f=1}^{F} \sum_{h=1}^{H_f} x_{h,f} d_{h,f} C(t_{h,f}) \tag{16}$$

$$\text{subject to} \quad \max(t_{h,f}) \leq T_{f,max} \quad \forall f \in \mathbb{F}, h \in H_f \tag{17a}$$

$$t_{h,f} \leq T_h \quad \forall f \in \mathbb{F}, h \in H_f \tag{17b}$$

$$\sum_{h=1}^{H_f} x_{h,f} d_{h,f} = D_f \tag{17c}$$

$$x_{h,f} \in \{0,1\}, d_{h,f} \geq 0. \tag{17d}$$

where each request f has a total of H_f vehicles ready to share their content with BS-i, and $\bigcup_f^F H_f \subset H_i$, where H_i denotes the number of VUs currently associated with BS-i.

The constraint in Equation (17a) indicates the condition for guaranteeing the quality of the VSA. This means that off-loading file f from VUs to BS-i has to be complete before the requester VU actually arrives at BS-i. Constraint (17b) means that the selected vehicle has to upload data completely within its remaining time traveling T_h in BS-i. The total upload size is represented by constraint (17c). In the practical environment, there is a delay when calculating the value of $T_{f,max}$, T_h due to the variance of network or traffic on the road. To reflect this issue, we introduce two parameters as follows:

$$T'_{f,max} = T_{f,max} - \beta \tag{18}$$

$$T'_h = T_h - \theta, \tag{19}$$

where $\beta, \theta > 0$. Hence, objective (16) can be re-written as

$$\min_{x,d} \quad \sum_{f=1}^{F} \sum_{h=1}^{H_f} x_{h,f} d_{h,f} C(t_{h,f}) \tag{20}$$

$$\text{subject to} \quad \max(t_{h,f}) \leq T'_{f,max} \quad \forall f \in \mathbb{F}, h \in H_f \tag{21a}$$

$$t_{h,f} \leq T'_h \quad \forall f \in \mathbb{F}, h \in H_f \tag{21b}$$

$$(17c), (17d).$$

The formulation of cost-effective video transmission based on volunteer nodes (i.e., vehicles) in (20) is a mixed nonlinear integer programming (MNIP) problem. The binary variables $x_{h,f}$ ($\forall f \in \mathbb{F}, h \in H_f$) represent how vehicles are properly selected for obtaining the requested content. Meanwhile, non-integer variables indicating the amount of data off-loaded to the BS from each vehicle. Thus, finding a global solution of objective function (20) is an NP-hard problem [46]. Alternatively, this MINLP problem can be solved by the COIN-OR toolbox in the Julia language, as it is a well-known commercial optimization tool in solving the near-optimal for non-linear programming. Additionally, the computational complexity is also efficient. In the COIN-OR toolbox, the BONMIN solver [47] uses IPOPT to find the solutions for relaxed problems. Meanwhile, CBC package is used to solve mixed integer problem [48,49]. It combines several algorithms such as B-OA outer-approximation, which generates and successively improves the outer approximation of MIP, and a branch-and-bound algorithm is used to reduce the search space [49]. When the solution in (20) is infeasible, the algorithm will download data from the remote server.

Corollary 1. *Given that a request arrives at BS-i, denoted as $Req = \{D_f, T_{f,max}\}$, BS-i can guarantee the QoS for this request if the number of VUs inside BS-i, which can provide the content f to BS-i, satisfying this condition.*

$$\left| H_f \right| \geq \frac{D_f}{d_{f,max}} \geq \frac{D_f}{\min \left\{ \tilde{R}_h T_{f,max}, \sqrt[\gamma]{\frac{p\tilde{R}_h^\gamma}{b}} \right\}} \tag{22}$$

Proof. From the QoS constraint in Equation (17a), we have:

$$\max(t_{h,f}) \leq T_{f,max} \quad \forall f \in \mathbb{F}, h \in H_f \tag{23}$$

Let $t_{max} = max(t_{h,f})$, where $t_{max} = \frac{d_{f,max}}{\tilde{R}_h}$. Herein, $d_{f,max}$ is the maximum data size that a VU-h can share with its connected BS-i during time $T_{f,max}$. Substitute this into the above Equation (23), we have:

$$d_{f,max} \leq \tilde{R}_h T_{f,max} \tag{24}$$

Furthermore, from the fact that the value of the cost function in Equation (5) is always positive, we can infer that

$$d_{f,max} \leq \sqrt[\gamma]{\frac{p\tilde{R}_h^\gamma}{b}} \tag{25}$$

From Equations (24) and (25), we have:

$$d_{f,max} \leq \min \left\{ \tilde{R}_h T_{f,max}, \sqrt[\gamma]{\frac{p\tilde{R}_h^\gamma}{b}} \right\} \tag{26}$$

Let $\left| H_f \right|$ indicates the number of VUs, which can provide the content f to BS-i. To upload successfully the content f before the requester arrives at BS-i, the total data size uploading to BS-i must satisfy this equation

$$\left| H_f \right| d_{f,max} \geq D_f \tag{27}$$

Hence, from Equations (27) and (26), we can deduce that the lower bound of the number of VUs required for satisfying a requested file f is given as

$$\left| H_f \right| \geq \frac{D_f}{d_{f,max}} \geq \frac{D_f}{\min \left\{ \tilde{R}_h T_{f,max}, \sqrt[\gamma]{\frac{p\tilde{R}_h^\gamma}{b}} \right\}} \tag{28}$$

Furthermore, if Equation (28) is not satisfied, then the video will be fetched from the core cloud to guarantee the QoS.

6. Simulation Results

6.1. Dataset and Parameter Setup

6.1.1. Taxi Trajectory Data Trace

We use trajectory data containing all green taxi and limousine trips in New York City (NYC) for real vehicle traffic [50]. The dataset collected approximately one and a half millions taxi trips in NYC within one month in September 2015. Each record represents an individual trip, which consists of GPS pick-up/drop-off location information, the date and time of pick-up and drop-off, rate code, trip distance, etc. From these data, we extract all of the trips that traveled between Manhattan and JFK (John F. Kennedy) airport with a trip distance over ten miles. We consider the time of request and time of pick-up to be the same, for which the pick-up time is sampled from 12 PM to 6 PM. Meanwhile,

the travel time on each road segment of the taxi trips is calculated by the daily mean travel time and daily mean distance estimate. The summary of distance distribution and hourly distribution of trips are shown in Figures 3 and 4.
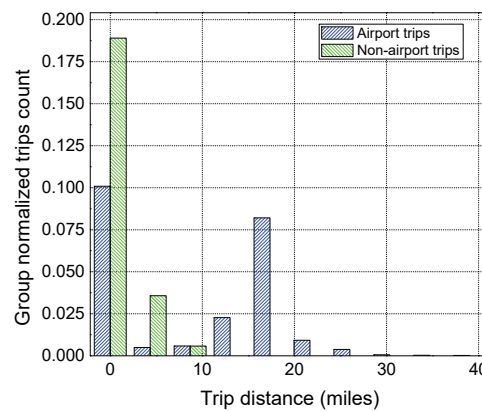

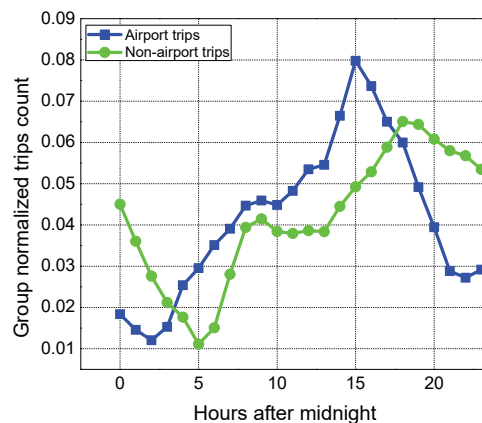
Figure 3. Trip distance distribution.

Figure 4. Hourly distribution of trips.

6.1.2. Base Station Data Trace

We also collect the locations of BSs in New York City via the free and publicly available OpenCellid database [51]. To do this, we filter out all BSs located in NYC based on longitude and latitude from OpenCellid database. Then, we extract some interesting information such as cell ID, GPS location of the BSs, etc. Finally, we get a dataset with more than four million unique cell IDs. From these data, we extract all BS positions that were located in the same area as the taxi trips.

6.1.3. Mapping of the Two Datasets

In our experiment, we match the vehicle locations to their corresponding BS to explore the BSs that are available to VUs when they use the VSA. Suppose that the signal coverage of each BS L is 1000 m, within which VUs can connect to the BS. Each VU is mapped to the nearest BS if the distance between the VU and the BS is less than 1000 m, otherwise, we will remove the VUs that do not map to any BSs.

6.1.4. Parameters Setup

Based on the results from combining two datasets, we can infer the number of vehicles connected to their nearest BS, and then the density of vehicles on the road is set as $\lambda = 0.15$ (150/km/lane). For the lower bound of the average data rate, all the parameters are set up similar to that in [45]. Note that, based on their results, the lower bound in Equation (13) is asymptotic with the actual achievable data

rate under different traffic and BSs densities (Experiment A in [45]), so the average data rate $\bar{R}_h$ of each VU-*h* in the BS-*i* from Equation (13) is approximately 2 Mbps. Based on the real vehicle traffic dataset analysis above, we also know the mean travel time on each BS-*i* of every VU-*h*, specifically in a range of [20, 50] s. The size of each request arriving at BS-*i* is generated randomly in a range of [20, 30] MB. The number of vehicles inside BS-*i* that can share video data with their connected BS-*i* is generated so as to satisfy Equation (28). The hyper-parameters p, b, γ in cost function (5) are set with values of 10 cents/MB, 0.02, and 1.3, respectively.

6.2. Performance Analysis

We first conduct experiments for the scenario when BS-*i* does not store the requested files in its cache. BS-*i* has to make the decision to download such files from VUs or from the core cloud in order to optimize the VSA cost while guaranteeing the time delay constraint. We then compare the performance of our proposed model with two other strategies as follows:

- Content Delivery Networks (CDN) scheme [52]: The copies of content are distributed to edge servers that are geographically closer to users. A user request is directed to the nearest edge server with the capacity to respond. If the content is not in the edge server cache, this scheme attempts to download it from the origin server. When the cache receives the content, the cache streams data to the user. The VSP's cost is defined as the cost for downloading data from origin server to serve user requests.

- Peer-Assisted Content Delivery Networks centralized (PA-CDN) scheme [53]: In this scheme, users (peers) not only consume data but also provide their storage resources for other peers in the same group. This enlarges data availability for this content delivery scheme. If the user fails to connect to the group of dedicated peers, or if the data cannot be received within the QoS requirement, the edge server serves the request directly. Please note that a video is divided into several small chunks of the same size and transferred through the P2P network in this scheme. The VSP's cost is defined as the cost the VSP has to pay to encourage peers to contribute their data.

Figure 5 presents the VSP's cost under three different strategies. In particular, we evaluate the total cost that a BS spends for acquiring data to serve the requests in terms of the arrival rate of requests. This experiment is conducted by modeling the sequence of ten requests following the Poisson distribution with different rates per minute $\bar{\lambda} = 1, 2, ..., 20$. For a clear comparison, the total data size that a BS needs to obtain is equal for each value of $\bar{\lambda}$. In addition, we generate ten different sequences of the same $\bar{\lambda}$ value to derive the average result of the total cost as shown in Figure 5. We also set the value of $\beta = \theta = 0$. It can be seen that our proposed scheme greatly reduces the cost for a BS compared with the two baseline methods when the arrival rate of the requests is low. However, the cost-saving effectiveness is decreasing when the arrival rate is high. When $\bar{\lambda}$ is small, the interval time between two consecutive requests is longer, so a large portion of the requests in a BS queue can be optimally accomplished by obtaining data from VUs inside the BS range. As a result, significant cost-saving is achieved. On the contrary, when the arrival rate is high, due to the interval time of two successive tasks being shorter, some tasks are fulfilled by downloading data from the core cloud instead of from VUs to satisfy the time delay constraint. Hence, the total cost-saving is slightly decreasing. Moreover, the difference between the costs of our proposed scheme and the CDN scheme is higher than the difference of PA-CDN against the CDN mechanism.

Figure 6 presents the VSP's cost gap when the traffic/network is changed over time. We compare the difference of the solution of problem (20) when $\beta = \theta = \left\{ 3, 4, 5, 8 \right\}$ seconds with the solution of problem (16) when $\beta = \theta = 0$. According to the results, the solution gap increases when the variance of traffic/network is larger (from 3 s to 8 s). For a specific value of β (or θ), the solution gap decreases when the arrival rate $\bar{\lambda}$ increases. When $\bar{\lambda}$ increases, VSP tends to download data from the remote server rather than from volunteer nodes so as to satisfy the QoS requirement. Hence, the difference between solutions in (20) when $\beta = \theta = 2$ to 8 and solutions in (16) is reduced.
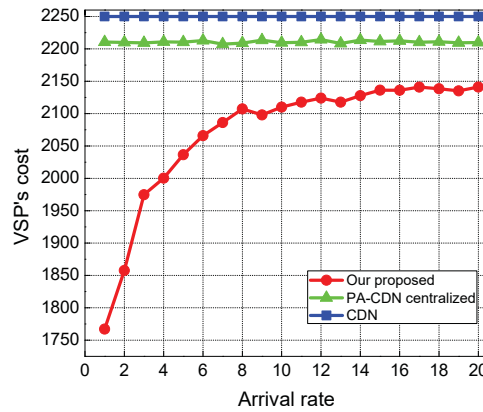
Figure 5. Spending cost in terms of the arrival rate.
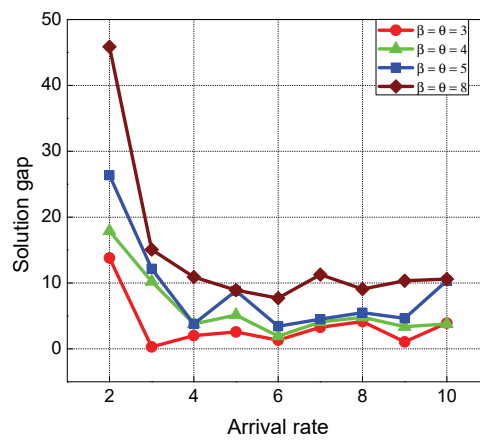

Figure 6. The solution gap in terms of the variance of traffic.

Figure 7 presents the VSP's cost under different vehicle speeds. For this experiment, we also generate a sequence of ten requests arriving at the BS, but they are generated with the same arrival rate $\bar{\lambda} = 3$ for each speed. As shown, when the vehicle has a faster velocity, the total data it can receive from the BS's streaming process is lower during traveling time. Hence, we can see that all three mechanisms have lower spending costs with faster vehicle speed. Moreover, the difference between our proposed mechanism and the two baseline mechanisms is higher when the vehicle speed is lower and vice versa. The reason for this result is that when the speed is low, the BS's cost-saving is efficient because it obtains data from VUs. However, when the speed is high, the time delay constraint of each task is more strictly bound. Therefore, the BS tends to download data from the higher layer cloud rather than from VUs.

Figure 8 shows the VSP's cost under different average data rates. In this experiment, we change the value of $\tilde{R}_h$ among $\left\{0.5, 1, 2, 3, 5\right\}$ Mbps to investigate the performance of our proposed mechanism. The horizontal axis is represented by the different arrival rate of requests. As shown, when $\tilde{R}_h = 0.5$ Mbps, the VSP's cost spending is frequently higher than the others. This is because if the average transmission rate of each VU is lower, then the time needed to completely upload data into the BS is longer. Hence, to guarantee QoS for the VSA, the BS tends to download data from the remote content server. Otherwise, the remaining results induce that the paying cost is inversely proportional to the data rate, based on Equation (5).
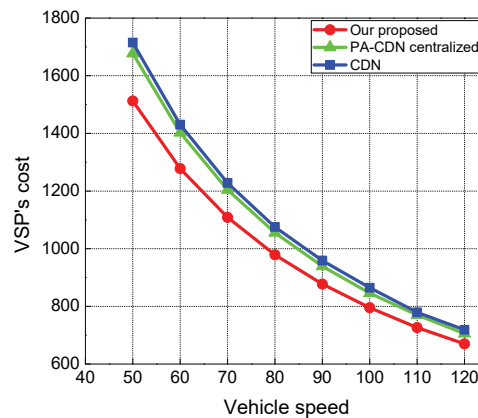
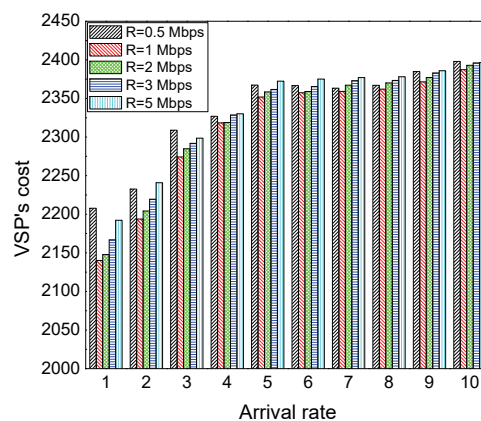**Figure 7.** The spending cost in terms of the vehicle speed.

Figure 8. The spending cost in terms of the data rate variation.

We now conduct an experiment for the scenario when the BS has already cached a part of the requests. The purpose is to investigate the relationship between cache size and the VSP's cost spending. Figure 9 shows the VSP's cost under different BS cache capacities. The cache capacity is defined as the number of files that can be stored at the BS. With this experiment, we set the size of the BS cache as $\left\{50, 100, 150, 200, 250, 300, 350, 400\right\}$ files out of a total of $F = 1000$ files. For the VSA, the Zipf exponent ψ is set as 0.56 based on real data statistics from YouTube [54]. The cache miss ratio is identified by averaging of 100 generated requested files to the BS corresponding to each cache size. It can be seen that the optimal VSP cost-saving increases with cache capacity since a larger cache capacity can increase the cache hit ratio when fetching content. In particular, the VSP's cost-saving can achieve approximately 52% when increasing the cache size from 50 to 400.
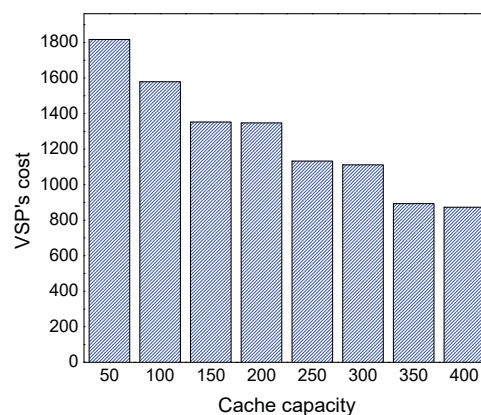

Figure 9. The spending cost in terms of the cache capacity.

We also analyze the effect of various parameters on the performance of our proposed model. As shown in Figure 10, we again evaluate the VSP's cost-saving by studying the effect of the number of VUs H_f that can share their data with the BS. The experiment is conducted with the fixed arrival rate $\lambda = 3$. H_f ranges from 1 to 15 vehicles. We found that with $H_f = 1$, the VSP's cost-saving is lower saving. This is caused by the condition of Equation (28). When the number of VUs is not enough to share data with the BS within the time delay constraint, the BS will download data directly from the remote server. The cost-spending decreases dramatically when H_f increases from 1 to 3. After that, it slightly reduces when the H_f value changes to 4, 5, and 6. However, when the H_f value equals 7, the cost-saving is likely to be stable.

Figure 11 presents the effect of hyper-parameter γ on the VSP's cost. From Equation (5), while p is a fixed cost between two values b and γ, we realize that the exponent factor will have a much greater impact on the total cost. Hence, we set up the experiment with a range of values of γ from 1.1 to 2 to analyze its effect. It can be seen that the VSP's cost decreases gradually from 1.1 to 1.6, reaching the lowest point of $\gamma = 1.7$. The reason is clearly shown in Equation (5), where the cost decreases with the γ value. However, when γ is greater than 1.7, the VSP's cost goes up noticeably. This is caused by the upper bound of the upload time in Equation (5), $t \leq \left(\frac{p}{b}\right)^{\frac{1}{\gamma}}$. The required upload time t is shorter when γ increases. In addition, the average data rate $\bar{R}_h$ does not change. Hence, the time constraint (21a) is violated. To guarantee the performance, the BS has to fetch content from the remote server, which then causes the increased cost.
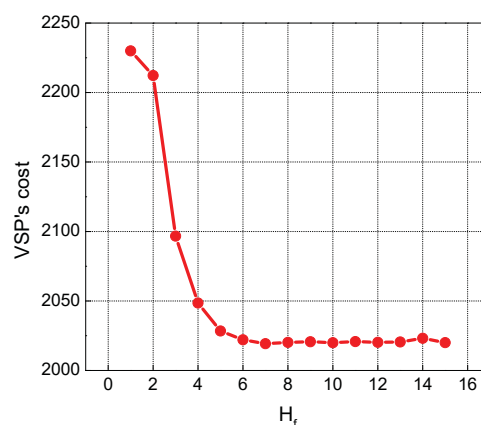

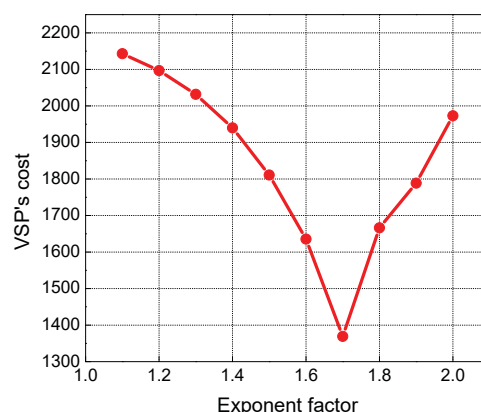**Figure 10.** Effect of the number of vehicles H_f.

Figure 11. Effect of the exponent factor γ.

7. Conclusions

In this paper, we introduced a new mechanism to reduce the VSP's cost of providing VSA for the emerging volunteer computing environment. We discuss the decision problem as mixed nonlinear integer programming to calculate the total payment of the VSP. The solution represents a win-win

strategy for both the VSP and mobile users. Therefore, the VSP can reduce the serving cost, while the mobile users gain profits from sharing their existing resources in our proposal. The simulation results indicate that our proposal significantly improves the VSP's cost under different data traffic conditions in vehicular networks. In future work, we will extend our model considering the various quality of service requirements.

Author Contributions: Software, T.D.T.N.; Supervision, E.-N.H.; Writing—Review and Editing, T.D.T.N., T.-D.N., V.D.N. and X.-Q.P.

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

References

1. Vegni, A.M.; Loscrí, V. A Survey on Vehicular Social Networks. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2397–2419. [CrossRef]
2. Watzenig, D.; Horn, M. *Automated Driving: Safer and More Efficient Future Driving*; Springer: Berlin/Heidelberg, Germany, 2016.
3. Yu, R.; Huang, X.; Kang, J.; Ding, J.; Maharjan, S.; Gjessing, S.; Zhang, Y. Cooperative Resource Management in Cloud-Enabled Vehicular Networks. *IEEE Trans. Ind. Electron.* **2015**, *62*, 7938–7951. [CrossRef]
4. The Statistics Portal. 2018. Available online: http://www.statista.com/ (accessed on 2 September 2018).
5. Curry, E.; Derguech, W.; Hasan, S.; Kouroupetroglou, C.; Hassan, U. A Real-time Linked Dataspace for the Internet of Things: Enabling "Pay-As-You-Go" Data Management in Smart Environments. *Future Gener. Comput. Syst.* **2019**, *90*, 405–422. [CrossRef]
6. Cicirelli, F.; Fortino, G.; Guerrieri, A.; Spezzano, G.; Vinci, A. Metamodeling of Smart Environments: From design to implementation. *Adv. Eng. Inform.* **2017**, *33*, 274–284. [CrossRef]
7. Zeeshan, H.M.; Fethi, F. LTE and IEEE 802.11p for vehicular networking: A performance evaluation. *EURASIP J. Wirel. Commun. Netw.* **2014**. [CrossRef]
8. Woźniak, M.; Połap, D. Object detection and recognition via clustered features. *Neurocomputing* **2018**, *320*, 76–84. [CrossRef]
9. Połap, D.; Kęsik, K.; Książek, K.; Woźniak, M. Obstacle Detection as a Safety Alert in Augmented Reality Models by the Use of Deep Learning Techniques. *Sensors* **2017**, *17*, 2803. [CrossRef] [PubMed]
10. Liu, J.; Kato, N.; Ma, J.; Kadowaki, N. Device-to-Device Communication in LTE-Advanced Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1923–1940. [CrossRef]
11. He, S.; Shin, D.H.; Zhang, J.; Chen, J.; Sun, Y. Full-View Area Coverage in Camera Sensor Networks: Dimension Reduction and Near-Optimal Solutions. *IEEE Trans. Veh. Technol.* **2016**, *65*, 7448–7461. [CrossRef]
12. Kang, J.; Yu, R.; Huang, X.; Jonsson, M.; Bogucka, H.; Gjessing, S.; Zhang, Y. Location privacy attacks and defenses in cloud-enabled internet of vehicles. *IEEE Wirel. Commun.* **2016**, *23*, 52–59. [CrossRef]
13. Li, J.; Chen, H.; Chen, Y.; Lin, Z.; Vucetic, B.; Hanzo, L. Pricing and Resource Allocation via Game Theory for a Small-Cell Video Caching System. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 2115–2129. [CrossRef]
14. Wu, H.; Liu, L.; Zhang, X.; Ma, H. Quality of video oriented pricing incentive for mobile video offloading. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
15. Zhu, W.; Li, D.; Saad, W. Multiple Vehicles Collaborative Data Download Protocol via Network Coding. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1607–1619. [CrossRef]
16. Volvo Concept 26. 2018. Available online: https://www.volvocars.com/my/cars/concept-cars/c26 (accessed on 5 September 2018).
17. ETSI, Sophia Antipolis, France. Mobile-Edge Computing-Introductory Technical White Paper. 2014. Available online: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1 (accessed on 9 September 2018).

18. Intel, Santa Clara, CA, USA. Real-World Impact of Mobile Edge Computing (MEC). 2016. Available online: https://builders.intel.com/docs/networkbuilders/Real-world-impact-of-mobile-edgecomputing-MEC.pdf (accessed on 11 September 2018).

19. Satyanarayanan, M.; Lewis, G.; Morris, E.; Simanta, S.; Boleng, J.; Ha, K. The Role of Cloudlets in Hostile Environments. *IEEE Pervas. Comput.* **2013**, *12*, 40–49. [CrossRef]

20. Mach, P.; Becvar, Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1628–1656. [CrossRef]

21. Zhang, K.; Mao, Y.; Leng, S.; Maharjan, S.; Zhang, Y. Optimal delay constrained offloading for vehicular edge computing networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.

22. Vigneri, L.; Pecoraro, S.; Spyropoulos, T.; Barakat, C. Per-Chunk Caching for Video Streaming from a Vehicular Cloud. In *Proceedings of the 12th Workshop on Challenged Networks (CHANTS '17)*; ACM: New York, NY, USA, 2017; pp. 57–62.

23. Liu, N.; Liu, M.; Chen, G.; Cao, J. The sharing at roadside: Vehicular content distribution using parked vehicles. In Proceedings of the 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012), Orlando, FL, USA, 25–30 March 2012; pp. 2641–2645.

24. Su, Z.; Ren, P.; Gan, X. A novel algorithm to cache vehicular content with parked vehicles applications. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 5665–5669.

25. Souza, V.B.; Masip-Bruin, X.; Marín-Tordera, E.; Sànchez-López, S.; Garcia, J.; Ren, G.J.; Jukan, A.; Ferrer, A.J. Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures. *Future Gener. Comput. Syst.* **2018**, *87*, 1–15. [CrossRef]

26. Abdelhamid, S.; Hassanein, H.S.; Takahara, G. Vehicle as a resource (VaaR). *IEEE Netw.* **2015**, *29*, 12–17. [CrossRef]

27. Lee, E.-K.; Gerla, M.; Pau, G.; Lee, U.; Lim, J.-H. Internet of vehicles: from intelligent grid to autonomous cars and vehicular fogs. *Int. J. Distr. Sens. Netw.* **2016**, *12*, 1–14. [CrossRef]

28. Nabi, M.; Benkoczi, R.; Abdelhamid, S.; Hassanein, H.S. Resource assignment in vehicular clouds. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.

29. Zhang, K.; Mao, Y.; Leng, S.; He, Y.; Zhang, Y. Mobile-Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Off-Loading. *IEEE Veh. Technol. Mag.* **2017**, *12*, 36–44. [CrossRef]

30. Su, Z.; Xu, Q.; Hui, Y.; Wen, M.; Guo, S. A Game Theoretic Approach to Parked Vehicle Assisted Content Delivery in Vehicular Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 6461–6474. [CrossRef]

31. Kang, X.; Wu, Y. Incentive Mechanism Design for Heterogeneous Peer-to-Peer Networks: A Stackelberg Game Approach. *IEEE Trans. Mob. Comput.* **2015**, *14*, 1018–1030. [CrossRef]

32. Tang, M.; Gao, L.; Pang, H.; Huang, J.; Sun, L. A multi-dimensional auction mechanism for mobile crowdsourced video streaming. In Proceedings of the 2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Tempe, AZ, USA, 9–13 May 2016; pp. 1–8.

33. Nguyen, T.; Kim, Y.; Pham, X.; Nguyen, T.D.T.; Huh, E. *Mobile Services Meet Distributed Cloud: Benefits, Applications, and Challenges*; Mobile Computing Mutamed Khatib and Nael Salman; IntechOpen: London, UK, 2018.

34. Tong, L.; Li, Y.; Gao, W. A hierarchical edge cloud architecture for mobile computing. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.

35. Shao, C.; Leng, S.; Zhang, Y.; Vinel, A.; Jonsson, M. Performance Analysis of Connectivity Probability and Connectivity-Aware MAC Protocol Design for Platoon-Based VANETs. *IEEE Trans. Veh. Technol.* **2015**, *64*, 5596–5609. [CrossRef]

36. Li, W.; Oteafy, S.M.A.; Hassanein, H.S. Dynamic adaptive streaming over popularity-driven caching in Information-Centric Networks. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 5747–5752.

37. Yang, H.; Deng, Y.; Qiu, J.; Li, M.; Lai, M.; Dong, Z.Y. Electric Vehicle Route Selection and Charging Navigation Strategy Based on Crowd Sensing. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2214–2226. [CrossRef]

38. Zhu, Y.; Wu, Y.; Li, B. Trajectory Improves Data Delivery in Urban Vehicular Networks. *IEEE Trans. Parallel Distr. Syst.* **2014**, *25*, 1089–1100.

39. Tran, T.X.; Pandey, P.; Hajisami, A.; Pompili, D. Collaborative multi-bitrate video caching and processing in Mobile-Edge Computing networks. In Proceedings of the 2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Jackson, WY, USA, 21–24 February 2017; pp. 165–172.

40. Khabbaz, M.J.; Fawaz, W.F.; Assi, C.M. A Simple Free-Flow Traffic Model for Vehicular Intermittently Connected Networks. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1312–1326. [CrossRef]

41. Chaqfeh, M.; Lakas, A.; Jawhar, I. A survey on data dissemination in vehicular ad hoc networks. *Veh. Commun.* **2014**, *1*, 214–225. [CrossRef]

42. Schwarz, M.; Sauer, C.; Daduna, H.; Kulik, R.; Szekli, R. M/M/1 Queueing systems with inventory. *Queue. Syst.* **2006**, *54*, 55–78. [CrossRef]

43. Zhuo, X.; Gao, W.; Cao, G.; Dai, Y. Win-Coupon: An incentive framework for 3G traffic offloading. In Proceedings of the 2011 19th IEEE International Conference on Network Protocols, Vancouver, BC, Canada, 17–20 October 2011; pp. 206–215.

44. Imsook, H.; Steven, W.; Johannes, B. P2P, CDNs, and Hybrid Networks: The Economics of Internet Video Distribution. *Int. Telecommun. Policy Rev.* **2010**, *17*, 1–22.

45. Zhang, S.; Zhang, N.; Fang, X.; Yang, P.; Shen, X.S. Cost-effective vehicular network planning with cache-enabled green roadside units. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.

46. Burer, S.; Letchford, A.N. Non-convex mixed-integer nonlinear programming: A survey. *Surv. Oper. Res. Manag. Sci.* **2012**, *17*, 97–106. [CrossRef]

47. Bonami, P.; Biegler, L.T.; Conn, A.R.; Cornuéjols, G.; Grossmann, I.E.; Laird, C.D.; Lee, J.; Lodi, A.; Margot, F.; Sawaya, N.; et al. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optim.* **2008**, *5*, 186–204. [CrossRef]

48. Masteri, K.; Venkatesh, B.; Freitas, W. A Fuzzy Optimization Model for Distribution System Asset Planning With Energy Storage. *IEEE Trans. Power Syst.* **2018**, *33*, 5114–5123. [CrossRef]

49. Michael, B.; Stefan, V. *MINLP Solver Software*; Wiley: Hoboken, NJ, USA, 2011. [CrossRef]

50. Green Taxi NYC. 2015. Available online: https://s3.amazonaws.com/nyc-tlc/trip+data/green_tripdata_2015-09.csv (accessed on 12 September 2018).

51. Base Station Dataset. 2018. Available online: https://www.opencellid.org/ (accessed on 25 September 2018).

52. Pathan, A.M.K.; Rajkumar, B. *A Taxonomy and Survey of Content Delivery Networks*; University of Melbourne: Parkville, Australia, 2012.

53. Anjum, N.; Karamshuk, D.; Shikh-Bahaei, M.; Sastry, N. Survey on peer-assisted content delivery networks. *Comput. Netw.* **2017**, *116*, 79–95. [CrossRef]

54. Gill, P.; Arlitt, M.; Li, Z.; Mahanti, A. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07)*; ACM: New York, NY, USA, 2007; pp. 15–28.