Edge Learning for Surveillance Video Uploading Sharing in Public Transport Systems

Laizhong Cui, *Senior Member, IEEE*, Dongyuan Su, Yipeng Zhou, *Member, IEEE*, Lei Zhang, *Member, IEEE*, Yulei Wu, *Senior Member, IEEE*, and Shiping Chen, *Senior Member, IEEE*

***Abstract*—Nowadays, surveillance cameras have been pervasively equipped with vehicles in public transport systems. For the sake of public security, it is crucial to upload recorded surveillance videos to remote servers timely for backup and necessary video analytics. However, continuously uploading video content generated by tens of thousands of vehicles can be extremely bandwidth consuming. In this work, we investigate the video uploading problem for moving buses by proposing to deploy dedicated access points (AP) at bus stops to facilitate video uploading. We define the harmonic objective for our problem, which includes minimizing the video uploading delay and minimizing the AP deployment cost. This problem is with two fundamental challenges. Firstly, it is difficult to balance the bandwidth capacity allocated to many buses because a bus obtains bandwidth resource from a series of APs deployed at stops along its route. Secondly, due to the randomness of bus movement and the complexity of bus routes, it is hard to predict the workload of an AP. Hence, it is challenging to estimate the delay of uploading video content through an AP. To cope with these challenges, we propose a water filling placement (WFP) algorithm, aiming to balance the aggregated bandwidth allocated to each bus. A queuing model is established to analyze the uploading delay of video content. We further resort to machine learning models to factor the influence of bus routes into our queuing model. Finally, a convex problem is formulated to optimize the harmonic objective, which can be optimally solved with the gradient descent (GD) based algorithm. We validate the correctness of our theoretical analysis and demonstrate the effectiveness of our method by carrying out extensive experiments using bus traces collected in Shenzhen city of China. In comparison with benchmark algorithms, our solution can always achieve the best performance.**

***Index Terms*—Video Uploading, Bus, AP Placement, Edge Learning, Transportation**

I. Introduction

Thanks to the unprecedented development of smart cities in the last decade, it is feasible to pervasively deploy video surveillance devices on public transport systems, such as buses and taxis [1]. The public security can be drastically improved if the recorded videos can be transmitted to remote cloud servers, in a timely manner, for video analytics and backup [2], [3]. Nevertheless, the massive video content uploading can introduce tremendous traffic to the current Internet, resulting in significant bandwidth consumption [4], [5]. Existing related works mainly focus on content downloading for vehicles [6], [7], [8]. Efficient video uploading for vehicles is still an open but important research issue. To bridge this gap, we propose to deploy dedicated access points (AP) at bus stops to facilitate surveillance video uploading for bus transportation systems. In practice, this is a challenging problem to minimize both the video uploading delay and the AP deployment cost.

We consider a scenario that buses continuously generate surveillance video content with a fixed rate when they are carrying passengers. The video content is uploaded to remote servers via APs placed at bus stops. It is worth noting that we consider the case where APs are only deployed at stops, so that buses can interact with them with a stable connection and close distance so as to achieve a high uploading speed. Both APs and buses maintain uploading buffers. Before a bus reaches an AP, the generated content will be cached in the bus's buffer. Here, we assume that the transmission speed between a bus and a connected AP is considerably high, because of a very short distance between them [9], [10], [11]. Once the bus builds a stable connection with an AP, all fresh content in the buffer will be uploaded to the AP. The content that may come from multiple buses, will be temporarily cached in the AP's buffer before uploading to remote servers with a constant rate.

In this scenario, we need to consider two objectives. First, surveillance video content should be uploaded to remote servers as early as possible for timely analysis [2], [12], avoiding caching buffers of APs to be overflowed. Second, there exists deployment cost including hardware cost and maintenance cost for each deployed AP, which should not be ignored. Thereby, we define a harmonic objective including two sub-objectives: minimizing video uploading delay and minimizing deployment cost. Although it is trivial to optimize either one of these two sub-objectives alone, it is rather

This work was supported in part by the National Key Research and Development Plan of China under Grants 2018YFB1800302 and 2018YFB1800805, the National Natural Science Foundation of China under Grants 61772345, 61902257 and 61872420, Major Fundamental Research Project in the Science and Technology Plan of Shenzhen under Grants JCYJ20190808142207420 and GJHZ20190822095416463, the Pearl River Young Scholars funding of Shenzhen University, Tencent "Rhinoceros Birds" - Scientific Research Foundation for Young Teachers of Shenzhen University, in part by the project of "FANet: PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications" under Grant LZC0019, and in part by Australia Research Council (ARC) under Grant DE180100950.

Laizhong Cui, Dongyuan Su and Lei Zhang are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518000, China, emails: cuilz@szu.edu.cn, 1900271016@email.szu.edu.cn, leizhang@szu.edu.cn;

Yipeng Zhou is with the Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2109, Australia, and also with the Peng Cheng Laboratory, Shenzhen 518055, China, email: ypzhou@mq.edu.au;

Yulei Wu is with the College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, U.K., email: Y.L.Wu@exeter.ac.uk;

Shiping Chen is with CSIRO Data61, Sydney, NSW 2015, Australia, email: shiping.chen@data61.csiro.au

Corresponding author: Yipeng Zhou.

complicated to consider both of them simultaneously, because the two sub-objectives conflict with each other. For instance, we can deploy APs at all bus stops to minimize the uploading delay. However, this solution will incur significant deployment cost.

The problem investigated in this paper is challenging, which can be illustrated from two aspects. Firstly, we consider the videos generated by all buses are of an identical bitrate, and thereby, it is necessary to balance the bandwidth allocated to each bus evenly. However, a moving bus can obtain bandwidth resources from a series of APs deployed at bus stops along its route. It implies that we need to balance the aggregated bandwidth resources allocated for each bus. Secondly, it is difficult to estimate the waiting time of video content in an AP's buffer, due to the randomness of bus movement and the influence of complex bus routes.

To deal with above challenges and deduce an optimal AP placement, we propose a solution with three steps. Firstly, we design a water filling placement (WFP) algorithm to balance the aggregation of bandwidth allocated to each bus for a given number of APs. Secondly, we develop a queuing model to estimate the uploading delay of video content. Machine learning models are then leveraged to factor the effect of bus routes in the queuing model. Finally, a convex optimization problem is formulated to decide the optimal number of APs to be deployed, which can be efficiently solved by gradient descent (GD) based methods [13].

To the best of the authors' knowledge, our work is the first to focus on the uploading issues of constantly generated surveillance videos for moving buses. Our contributions can be summarized as below.

- We propose a WFP algorithm to balance the aggregation of bandwidth allocated to each bus.
- A queuing model is developed to analyze the uploading delay of surveillance video content.
- A convex optimization problem is formulated to obtain the optimal number of APs with an harmonic objective.
- Extensive experiments are conducted based on the bus traces collected in Shenzhen city of China. The results not only validate the correctness of our analysis, but also demonstrate the superiority of our solution in comparison with other competitive baselines.

The rest of the paper is organized as follows. The preliminaries are introduced in Section II-A, and the problem formulation is presented in Section II-B. The AP placement algorithm is designed and shown in Section III. The theoretical analysis of the uploading delay is presented in Section IV. Optimizing the harmonic objective is illustrated in Section V. The related works are discussed in Section VII, and finally, our paper is concluded in Section VIII.

II. Model Formulation

A. Preliminaries

We consider the surveillance video uploading problem, in which buses can upload their videos via the APs deployed at different bus stops. When a bus approaches a particular AP, it will build up a connection with the AP and transmit video content to the AP. Note that the bus only needs to upload new video content that has not been uploaded before to the AP. Due to the fact that the AP is in the very close proximity to the bus, the transmission speed should be considerably high so that all the new video content can be properly transmitted to the AP.

Both buses and APs are equipped with caching buffers. The surveillance video content is cached in buses' buffers before being uploaded to an AP. In addition, the video content will be temporarily stored in AP's buffers before being uploaded to remote servers with a constant uploading rate and an first-in-first-out (FIFO) manner.[1] The video content uploaded to remote servers will be discarded from APs to refresh caching space for new content. For simplicity, we assume that the buffer size is large enough for both APs and buses so that buffers will be never overflowed [14].[2]

An example is depicted in Fig. 1 to illustrate the video uploading process considered in this paper.

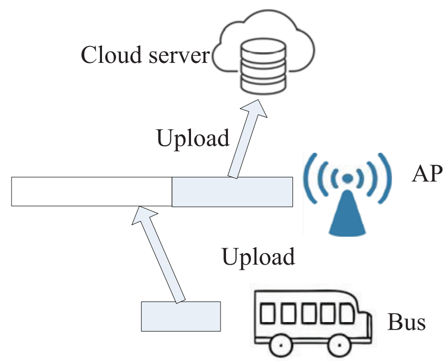


Fig. 1: An example to illustrate the video uploading process given a bus and a deployed AP at some stop.

B. Problem Formulation

There are two aspects that should be taken into account to design our objective.

As we have discussed in Section I, the video content should be uploaded to remote servers as early as possible. It implies that the first objective should be to minimize the video content's uploading delay. In principle, the uploading delay can always be reduced by deploying more APs in transportation systems. The uploading delay will be minimized by deploying APs at all bus stops, which however is a waste of AP resources. Because deploying each additional AP incurs extra hardware cost and maintenance cost, the second objective should be to minimize the number of deployed APs in transportation systems.

Note that it is not effective to deploy multiple APs at the same bus stop due to the reason that interference between multiple APs will dramatically reduce the uploading throughput

[1] We assume that APs are standardized products and their uploading capacity is fixed to simplify analysis. The feasibility to deploy heterogeneous APs with different upload capacity will be briefly discussed in Section V.

[2] More details and justifications will be explained in the experimental section.

TABLE I: Frequently used notations

Notation	Meaning
$\mathcal{N}, n$	the set of buses and the cardinality
$\mathcal{M}, m$	the set of bus stops and the cardinality
$\mathcal{S}_i$	the set of bus stops covered by bus i
T_i	the uploading delay of bus i
α	the weight corresponding to the AP deployment cost
w	the total number of APs to be deployed
Res_i	the number of APs allocated to serve bus i
D, d	the size of newly generated video and the expected value
$T_0, t_0, V[T_0]$	the time to travel between two adjacent stops, the expected value, and the variance
T_1, t_1	the delay before the video content is uploaded to an AP and the expected value
T_2, t_2	the transmission delay via an AP and the expected value
r	the bitrate of generated video content
u	the AP's uploading capacity
λ	the bus arrival rate of APs
ρ	the average uploading load of APs

[15]. Thus, we consider the condition that only a single AP can be deployed at each bus stop.

We assume that $\mathcal{N}$ is the set of buses with cardinality n, and $\mathcal{M}$ is the set of bus stops with cardinality m. The frequently used notations are shown in Table I. The route of each bus has been pre-determined. For a bus i, the set $\mathcal{S}_i$ denotes the route of the bus containing all stops bus i will go through.

We perceive the video uploading process from a stochastic perspective. Let T_i denote the uploading delay of the video content generated by bus i, which is a random variable affected by many factors, such as the AP placement algorithm, the competition from other buses, etc. Let I_j be an indicator that denotes whether bus stop j has been equipped with an AP. To involve both objectives into our analysis, we design the harmonic objective that assigns the weight α to the deployment cost and the weight $1-\alpha$ to the uploading delay. α is a parameter in the range $(0, 1)$, which can be tuned according to the importance of each objective. Then, the harmonic objective is defined as

$$obj = \alpha \sum_{j=1}^{m} I_j + (1-\alpha) \sum_{i=1}^{n} E[T_i], \tag{1}$$

which should be minimized. Here, I_j is with value 1 if an AP is placed at stop j, and 0 otherwise. $E[T_i]$ is the expected uploading delay suffered by the video content generated by bus i.

To minimize the harmonic objective defined in Eq. (1), we need to derive the expression of $E[T_i]$. Let u denote the uploading capacity of each AP. $E[T_i]$ is determined by the number of APs, the AP placement and u. The AP placement is a very complicated problem [16]. It is already an NP-complete problem by merely considering a simplified special case.

For the special case, we set $\alpha = 1$ and $u = +\infty$, which indicates that we only consider how to minimize the number of deployed APs that can cover all buses. It is sufficient to cover each bus with a single AP since its uploading capacity is infinity. Hence, the problem is converted to:

$$\min \quad \sum_{j=1}^{m} I_j, \tag{2a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{S}_i} I_j \geq 1, \tag{2b}$$

$$I_j \in 0, 1. \tag{2c}$$

The objective expressed in Eq. (2) is the simplified harmonic objective. The constraint in Eq. (2b) implies that all buses must be covered by at least once. This is a classical minimum set cover problem which is NP-hard [17]. It is common to devise heuristic algorithms to determine an AP placement.

Other than designing the AP placement algorithm, we still need to derive $E[T_i]$ to eventually solve this problem, which however is another complicated problem. In this paper, we propose a novel approach by leveraging a queuing model and machine learning models together to derive the expression of $E[T_i]$ once the AP placement is fixed.

To ease our discussion, the harmonic objective can be simplified as

$$obj = \alpha w + (1-\alpha) \sum_{i=1}^{n} E[T_i]. \tag{3}$$

Here, we consider w APs in total, where $w > 1$. The uploading delay $E[T_i]$ of each bus will be identical if the AP placement algorithm can balance the aggregated bandwidth allocated to each bus. Meanwhile, n (the number of buses) is omitted, because it can be regarded as a constant and can be absorbed into the term $1-\alpha$.

To summary, the harmonic objective defined in Eq. (3) can be minimized with three steps. Firstly, we heuristically design the AP placement algorithm that manages to balance the aggregation of bandwidth allocated to each bus once the total number of APs is fixed. Secondly, we employ the queuing theory to analyze the process of uploading video content through an AP. Machine learning models are utilized to determine two crucial parameters in the queuing model. Finally, we prove that $E[T_i]$ is a convex function, and hence, the harmonic objective defined in Eq. (3) is also a convex function with w. It can be minimized by searching the optimal w with GD based methods.

III. AP Placement Algorithm

As we have illustrated in Section II-B, the AP placement problem in principle is NP-complete. Inspired by previous works [16], [18], a feasible solution is to design heuristic algorithms. In this section, we propose a heuristic AP placement algorithm given the total number of APs is w. It is worth to emphasize that the speciality of our algorithm lies in balancing the aggregated bandwidth capacity allocated to each bus.

Let Res_i denote the number of APs allocated to serve bus i. The aggregated bandwidth capacity allocated to bus i is the aggregated capacity obtained from APs deployed at bus stops in the set $|\mathcal{S}_i|$. By considering the size of $|\mathcal{S}_i|$, our target is to balance the normalized bandwidth capacity, i.e., $\frac{Res_i}{|\mathcal{S}_i|}$ where $i = 1, \ldots, n$, for all buses, and thereby, our

AP placement algorithm is named as water filling placement (WFP) algorithm.

The design principle of the WFP algorithm is briefly described as follows. We define the popularity p_j for stop j as the number of buses that will go through it. Stops are ranked by decreasing order of p_j, and it is reasonable to choose stops of larger popularity with higher priority for AP placement. Thus, we begin the AP placement with the stop of the highest popularity. Once an AP is deployed, the aggregated bandwidth capacity of each bus is updated. If a bus has obtained sufficient bandwidth capacity, it should be removed and the stop popularity should be adjusted accordingly. The details of the WFP algorithm are presented in Algorithm 1.

Algorithm 1: Water Filling Placement (WFP) Algorithm

```
Data: The set of buses 𝒩; The set of stops 𝒮_i covered
      by bus i; The set of stops ℳ; The number of APs
      w to be deployed.
Result: The set of bus stops 𝒬 which should be
        deployed with APs.
Initialization: Res_i = 0 for n buses;
Compute p_j based on the route information 𝒮_i for all i's;
Rank all stops by the decreasing order of p_j ;
while w > 0 do
    Select stop j ∈ ℳ of the highest popularity p_j;
    Add j to the set 𝒬;
    for bus i covered by stop j do
        Increase Res_i by 1 ;
    Rank all buses by decreasing order of Res_i/|𝒮_i|;
    Remove top β percent of buses with the highest Res_i/|𝒮_i|
      from the set 𝒩;
    Re-compute the popularity of each stop;
    Update the rank of stops by the decreasing order of
    popularity;
    Add all removed buses back to the set 𝒩;
    Remove stop j from ℳ, and update w = w − 1;
Output: All stops in 𝒬.
```

In the WFP algorithm, β is a parameter used to adjust the balance of the bandwidth allocation. Before placing an AP, all buses will be ranked by the decreasing order of allocated bandwidth capacity. Then, top βn buses will be removed so that the AP is placed to enhance the bandwidth capacity for the rest $(1-\beta)n$ buses. If $\beta = 0$, the most popular and vacant stop is always chosen as the next one for AP placement. The value of β can be determined through experiments. In our study, we set $\beta = 0.1$ and empirically demonstrate that the video uploading performance is not sensitive when the value of β is around 0.1.

By placing an AP at stop j, we need to update the bandwidth capacity for βn buses, and the complexity of updating stop popularity is $O(\beta n)$. The complexities of ranking stops and buses are $O(m \log m)$ and $O(n \log n)$, respectively. Overall, the time complexity of the WFP algorithm is $O(\beta w n + w m \log m + w n \log n)$. This algorithm can be executed efficiently for a typical-size city with the scale of thousands of stops and tens of thousands of buses.

IV. Analysis of Uploading Delay

Once the AP placement is determined by the WFP algorithm, we establish a queuing model to analyze the uploading delay, $E[T_i]$. The proposed queuing analysis initially does not consider the information of bus routes. Machine learning models are then developed to factor the effect of bus routes in the derivation of $E[T_i]$.

A. Stochastic Analysis

By ignoring the influence of bus routes, we assume that the bandwidth capacity allocated to serve each bus is identical. In fact, bus routes are heavily affected by the road topology of a city, which makes the AP placement complicated. If the road topology is a straight line, it is easier to balance the bandwidth capacity allocated to each bus. However, if the road topology is a tree, balancing the bandwidth capacity becomes very challenging. We first establish a queuing model without considering such influence.

For the sake of clarity of the illustration, our analysis focuses on a particular bus. As we have described, when a bus approaches a stop that has been equipped with an AP, it will upload all newly generated video content with size D to the AP. D is regarded as a random variable, since it is uncertain how much time it will take for a bus to encounter the next AP since its departure from the previous AP. The probability that an AP has been deployed at a particular stop is $\frac{\gamma w}{m}$. Here, γ is a parameter determined by bus routes. If all stops are of the same popularity, we have $\gamma = 1$. Otherwise, if each of the stops that are ranked at top is deployed an AP, we should have $\gamma > 1$. γ will be determined later by a machine learning model. Note that we do not discriminate buses using different values of γ, since the AP placement algorithm aims to balance the number of APs to cover each bus, and each bus should have a roughly identical chance to encounter an AP though they may travel along different routes.

The uploading delay measures the time period since the birth of video content, with size D, to the time when the content is uploaded to remote servers. The uploading delay can be further decomposed into two parts: T_1, which is the delay before the video content is uploaded to an AP, and T_2, which is the transmission delay via an AP. Here, both T_1 and T_2 are perceived as random variables. The expected value of D, T_1 and T_2 are represented by d, t_1 and t_2, respectively.

According to the definition of t_1, it is the expected time for a bus to travel from one AP to the next. Since we have assumed that a bus meets an AP at a particular stop with probability $\frac{\gamma w}{m}$, it is expected that a bus will encounter the next AP after visiting $\frac{m}{\gamma w}$ stops on average.

Based on the above discussion, the expectation of T_1, i.e., t_1, can be given by

$$t_1 = E\left[\frac{m}{\gamma w} T_0\right] = \frac{m}{\gamma w} t_0, \tag{4}$$

where T_0 is defined as the variable to indicate the time a bus spends on traveling between two adjacent stops, with expected value t_0 and variance $V[T_0]$. t_0 and $V[T_0]$ can be estimated from the historical trace records of a bus.

B. Queuing Model

We derive the expression of t_2 by establishing a queuing model. Let r denote the bitrate of generated video content by a bus. By taking each AP as a server and video content to be uploaded as customers, we can employ the queuing theory to analyze our problem.

If an AP's buffer is perceived as a queue, each AP can be regarded as a server processing video content uploading tasks. Each task is a segment of video content uploaded by a bus. The task size is D and the server capacity is the AP's uploading capacity that can process tasks with a constant uploading rate u. The time to completely processing a task is then $\frac{D}{u}$. All tasks will be processed with an FIFO manner.

The bus arrival to accessing an AP approximately follows a Poisson process [19], [20]. The arrival rate is denoted by λ which is heavily affected by w. For popular stops serving many buses, λ should be larger, otherwise λ should be smaller. According to the WFP algorithm, a small number of popular stops will be selected for AP placement if w is small. Otherwise, if w is large, we have to place APs on some unpopular stops, and hence, λ is smaller. Thus, λ is affected by the AP placement as well, and it is complicated to precisely derive the expression of λ. To simplify our problem, we assume that λ is known here, and we can resort to developing a machine learning model to estimate λ later.

An uploading task with size D is always temporarily cached at the AP's buffer first, i.e., waiting in the queue. Thus, t_2 is the sum of the uploading processing time and the waiting time in the queue. If the distribution of D is unknown, this task uploading process can be modeled as an M/G/1 queue [21], and hence, the expression of t_2 can be given by:

$$t_2 = E[T_s] + \frac{\lambda E[T_s^2]}{2(1-\rho)}, \tag{5}$$

where $\rho \approx \frac{nr}{wu}$ is the average uploading load for each AP.[3] It is required that $\rho < 1$ to maintain the stability of an AP's buffer, and thus, we have $nr < wu$. We will further discuss the stability issue later. T_s is the service time (or uploading processing time) of a particular bus, i.e., the time to complete the uploading of a video segment with size D. In accordance, we have

$$\begin{aligned} E[T_s] &= \frac{E[D]}{u}, \\ E[T_s^2] &= E\left[\frac{D^2}{u^2}\right] = \frac{1}{u^2}\left(V[D] + E[D]^2\right). \end{aligned}$$

Here $V[D]$ is the variance of D. According to Eq. (4), $E[D] = t_1 r = \frac{m}{\gamma w} t_0 r$ and $V[D] = \frac{m^2 r^2}{\gamma^2 w^2} V[T_0]$. $E[T_s]$ and $E[T_s^2]$ are further derived as

$$E[T_s] = \frac{m t_0 r}{\gamma w u}, \tag{6}$$

$$E[T_s^2] = \frac{m^2 r^2}{\gamma^2 w^2 u^2}\left(V[T_0] + t_0^2\right). \tag{7}$$

Recall that T_0 is the time a bus takes to travel between any two adjacent stops. t_0 and $V[T_0]$ are the expected value and variance of T_0, which can be learned from bus trace records.

By substituting Eq. (6) and Eq. (7) back into Eq. (5), we have:

$$t_2 = \frac{m t_0 r}{\gamma w u} + \frac{\lambda m^2 r^2}{2(1-\rho)\gamma^2 w^2 u^2}\left(V[T_0] + t_0^2\right). \tag{8}$$

C. Learning Parameters

1) Learning γ: Recall that a bus can access an AP at the next stop with probability $\frac{\gamma w}{m}$. γ is dependent on the value of w, the AP placement and bus routes. Thus, it is rather complicated to explicitly derive the expression of γ. With a given w and the WFP algorithm, we propose to learn γ with a machine learning model, which can automatically factor in the influence of bus routes.

As we have discussed, γ should be greater than 1 if the bus popularity is unevenly distributed, and WFP prefers to place APs on more popular stops. Meanwhile, γ should be a decreasing function with w, which can be intuitively explained as follows. As w is small, APs can be placed at a small number of popular stops, so as to more efficiently serve buses. However, as w increases, it becomes more difficult to identify appropriate popular stops to place APs, in order to balance the aggregated bandwidth allocation. In view of that, γw should be a concave increasing function, and γ is a convex decreasing function with w.

Based on the above discussion, we select $w^{\frac{9}{10}}/w, w^{\frac{8}{10}}/w, \ldots,$, $w^{\frac{1}{10}}/w$ as the basis functions of our learning model.[4] Specifically,

$$\gamma(w|\theta_\gamma) = \frac{1}{w}\left(c_0 + \sum_{l=1}^{9} c_l w^{\frac{10-l}{10}}\right). \tag{9}$$

Here, $\theta_\gamma = (c_0, c_l, \ldots c_9)$ represents the set of the parameters of the learning model. We can easily verify that $r(w|\theta)$ is a convex monotonic decreasing function with w.

θ_γ can be learned as follows. Once APs are placed by the WFP algorithm, we can compute γ easily. The WFP algorithm will be executed for multiple times by setting with different values of w, so that we can collect a number of samples with different w and the corresponding γ. By feeding these samples back to Eq. (9), we can train θ_γ by minimizing the fitting error.

2) Learning $\frac{\lambda}{1-\rho}$: Recall that λ is the average bus arriving rate to a particular AP and ρ is the average service load of each AP. If the popularity of bus stops is unevenly distributed, and WFP prefers to select more popular stops with higher priority, we anticipate that λ will decrease with w. When w is smaller, APs will be placed on a fewer number of popular stops, and hence, the average bus arriving rate will be higher. In contrast, if w is larger, less popular stops have to be chosen for AP placement, and hence, λ will be smaller. Meanwhile, the term $\frac{1}{1-\rho}$ decreases with the increase of w as well since ρ

[3] In reality, the load ρ is heavier for more popular stops and lighter for less popular stops. For the sake of clarity of illustration, we simply use a fixed average load here. The influence is negligible, which will be validated through trace-based experiments.

[4] The fitting error will be smaller if more basis functions are selected, but overfitting risk will be higher. According to experiment, using 9 basis functions is appropriate.

is expected to decrease with w. Then, we design the function $\frac{\lambda}{1-\rho}(w|\theta_\lambda)$ as a convex decreasing function as w approaches m. Based on the above discussion and by using the same set of basis functions, $\frac{\lambda}{1-\rho}(w|\theta_\lambda)$ is expressed as

$$\frac{\lambda}{1-\rho}(w|\theta_\lambda) = \sum_{l=0}^{9} b_l w^{\frac{-l}{10}}. \quad (10)$$

Here $\theta_\lambda = (b_0, b_1, \ldots, b_9)$ is the set of parameters to be fit. Similar to the training of θ_γ, a sample with a pair of w and λ can be collected if WFP is executed once. We learn θ_λ through a number of samples collected by executing WFP multiple times with different values of w.

V. Optimizing the Harmonic Objective

We embark optimizing the harmonic objective with the knowledge of $E[T_i]$.

A. Optimizing Objective

Recall that the harmonic objective includes two sub-objectives: minimizing deployment cost and minimizing uploading delay. Until now, all our discussions assume that w is a fixed value. Yet, this is not our final goal, and we need to find w_* that can minimize the harmonic objective.

By wrapping up all aforementioned arguments, our problem can be formulated as

$$\min_{w} \quad obj = \alpha w + (1-\alpha)(t_1 + t_2), \quad (11a)$$

$$\text{s.t.} \quad t_1 = \frac{m}{\gamma w} t_0, \quad (11b)$$

$$t_2 = \frac{m t_0 r}{\gamma w u} + \frac{\lambda m^2 r^2}{2(1-\rho)\gamma^2 w^2 u^2}\left(V[T_0] + t_0^2\right), \quad (11c)$$

$$\gamma = \frac{1}{w}\left(\hat{c}_0 + \sum_{l=1}^{9} \hat{c}_l w^{\frac{10-l}{10}}\right), \quad (11d)$$

$$\frac{\lambda}{1-\rho} = \sum_{l=0}^{9} \hat{b}_l w^{\frac{-l}{10}}, \quad (11e)$$

$$w \le m, \quad (11f)$$

$$wu > nr. \quad (11g)$$

Here $\hat{c}_l$ and $\hat{b}_l$ are parameters learned in Section IV. The condition, $wu > nr$, ensures that the overall uploading capacity is greater than the overall video generating rate, i.e., $\rho < 1$. Based on the current form of the harmonic objective, we prove:

Proposition *1:* The harmonic objective $obj = \alpha w + (1-\alpha)(t_1 + t_2)$ is a convex monotonic decreasing function with w.

The detailed proof can be found in the Appendix.

The range of w is $\left[\frac{nr}{u}, m\right]$, which is a convex set. Thus, optimizing the harmonic objective is a convex optimization problem, which can be efficiently solved by GD based algorithms [22]. Let η denote the learning rate, and we can iteratively search the value of w_* as below.

$$w \quad \leftarrow \quad w - \eta \frac{d\left(\alpha w + (1-\alpha)(t_1+t_2)\right)}{dw}.$$

In our experiments, we set $\eta = 0.1$, and the convergence condition is that the gradient is less than 10^{-4}.

B. Discussion of Stability

It is vital to maintain the condition $\rho < 1$ for each AP, so that the AP's buffer will not be overflowed. We discuss this problem from two perspectives.

Firstly, w should be large enough to ensure stability if we assume that u is a fixed value that cannot be adjusted arbitrarily. It implies that ρ is mainly determined by w. If w is too small, it is very risky that some APs placed at popular stops will suffer from exploding buffer increase. Therefore, to ensure the stability of APs' buffers, it is necessary to deploy a plentiful number of APs in the system.

Unfortunately, it is difficult to explicitly derive the threshold of w, over which the system will be stable and no AP's buffer will be overflowed. We turn to solving this problem by revising w_* searched by the GD algorithm. Given the AP placement and w_*, it is necessary to check the workload of each AP by computing $\rho_j = \frac{\lambda_j D}{u}$, where $j \in \mathcal{Q}$ and λ_j is the bus arrival rate to the stop j. If there exists any j such that $\rho_j > 1$, w_* should be increased to a value between $(w_*, m]$. To reduce searching time, the revised w_* can be determined through a binary search algorithm.

Secondly, the stability can be ensured better if an AP's upload capacity can be configured according to the load of the AP. Intuitively, it can be realized by deploying more powerful APs on more popular stops with heavier traffic loads. However, deploying heterogeneous APs unavoidly increases the deployment cost and complicates theoretical analysis. In addition, the deployment will be infeasible if bus routes were adjusted. due to the analysis of AP placement will be much more complicated.

In fact, the first solution is more practical for deployment. We will empirically prove the stability as long as w is not very small in the next section. By deploying standardized APs, if a very popular AP is overloaded, buses can carry video content cached in their buffers to the next APs to balance AP loads.

VI. Experimental Results and Analysis

In this section, we carry out trace-based experiments to validate the correctness of our theoretical analysis and the efficacy of our video uploading solution by comparing with baselines.

A. Experimental Settings

In our experiments, we use the dataset generated by the GPS terminals of buses in Shenzhen city of China. The dataset records the history trajectory of each bus from 1st Nov. to 7th Nov. 2018 denoted by trace-20181101 to trace-20181107, respectively. The record of each bus contains timestamp, bus line, bus ID, and bus location. The statistics of representative trace files are presented in Table II. As we can see, there are more than 14K buses on each day driving along more than 700 routes, and more than 5K available stops for AP placement.

Based on the collected bus traces in Shenzhen city, we have implemented a trace-driven video uploading simulator, consisting of two modules:

TABLE II: GPS trace file statistics.

	# bus IDs	# of bus routes	# bus stop
trace-20181101	14346	766	5146
trace-20181102	14351	765	5147

TABLE III: Traffic information in different time periods.

	t_0	$V[T_0]$	n
RH	146.94	15198.96	14281
NRH	132.28	11108.03	13976
Overall	140.19	13368.26	14321

1) Bus Module: This module simulates the driving of a bus, the generating of surveillance video, and the uploading of video content to APs;
2) AP Module: This module makes the AP deployment decisions and simulates the video uploading processes through APs.

B. Road Traffic Conditions

Traffic conditions in a city can vary significantly during a day. To present the fluctuating traffic information in different time periods, we measure the average arrival rate to each bus stop in each hour and present the results in Figure 2. From the figure, we can see that the time periods from 8 AM to 10 AM and from 5 PM to 7 PM are the rush hours (RH), the time periods from 11 AM to 4 PM are the non-rush hours (NRH), and the rest time periods are the free hours (FH). More detailed statistics of the training set are listed in Table III.

To further show the traffic conditions in the three time periods, we plot the cumulative distribution function (CDF) of the average duration for each bus traveling from one stop to the next stop, shown as Figure 3. From the CDF graph, we can see that about 24.25%, 42.04%, and 53.19% of the buses have duration of less than 125 seconds for RH, NRH, and FH, respectively, which means that traffic jams are more common in RH compared to FH and NRH.

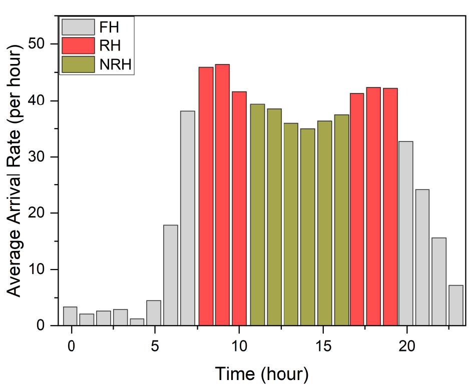


Fig. 2: Average bus arrival rates in different time periods.

We assume that an AP's communication range is 100 meters. By verifying the GPS trace files, we find that 80.40% of the buses stay within an AP's communication range for more than 42.5 seconds on average, shown as Figure 4. With higher-order modulation and other uplink transmission enhancements

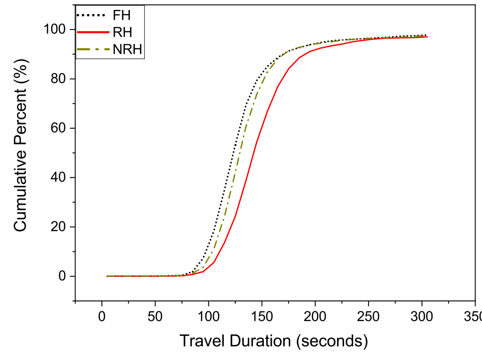


Fig. 3: The CDF of average duration for each bus traveling between two adjacent stops in different periods.

developed in 802.11 family, a very high transmission speed among buses and a connected AP is achievable [9], [10], [11]. Based on these observations, we assume that all fresh content in a bus's buffer have enough time to be uploaded to the neighboring AP.

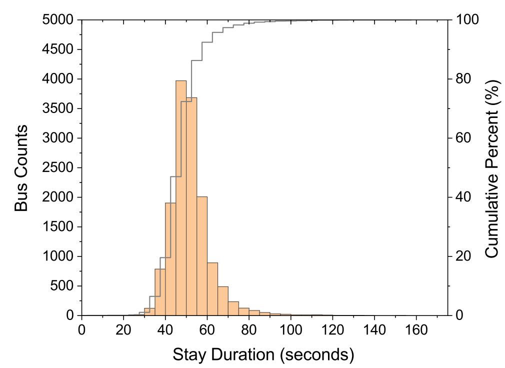


Fig. 4: The CDF of the average duration for each bus staying within an AP's communication range.

In our experiments, the video generating rate (i.e., r) is 8 Mbps, and the video uploading rate (i.e., u) of each AP is 1000 Mbps. We observe that a bus equipped with only 10GB hard disk can buffer video for more than 2.7 hours without uploading to any AP when the video generating rate is 8Mbps, i.e., 1MBps. For an AP, we intuitively set the buffer limit as 50GB for each of them to make the simulation more robust.

C. Model Verification

We first conduct experiments to verify the correctness of our theoretical analysis. Since we adopt machine learning models to determine the values of γ and $\frac{\lambda}{1-\rho}$, we use trace-20181101 as the training set and trace-20181102 to trace-20181107 as the testing set.

As the definitions of γ and $\frac{\lambda}{1-\rho}$ have been illustrated in Section IV, we can learn γ and $\frac{\lambda}{1-\rho}$ accordingly, which can be further used to calculate t_1 and t_2, respectively. The parameters θ_γ and $\theta_{\frac{\lambda}{1-\rho}}$ learned based on Eqs. 9 and 10 are listed in

TABLE IV: Parameters for fitting γ and $\frac{\lambda}{1-\rho}$

c_0-c_3	-4.798	-7.223	21.781	-2.715
c_4-c_7	14.849	11.366	-22.055	-13.549
c_8-c_9	-10.419	-7.421		
b_0-b_3	-0.048	0.212	0.103	0.038
b_4-b_7	-0.231	-0.823	-1.045	-1.265
b_8-b_9	3.118	17.029		

Table IV. The regularization parameters for learning γ and λ are set as 1e-3 and 1e-7, respectively.

We depict the learned parameters against with w in Figures 5 and 6, where w increases from 1000 to 3000. We further plot the measured parameters obtained from the testing set to verify the correctness of our machine learning models. As we can observe from Figures 5 and 6, γ and $\frac{\lambda}{1-\rho}$ are indeed convex monotonic decreasing with the increase in w. The gap between the parameters calculated with the learned models and the parameters measured from the training set is very small, which indicates the low fitting errors and the good applicability of our machine learning models.

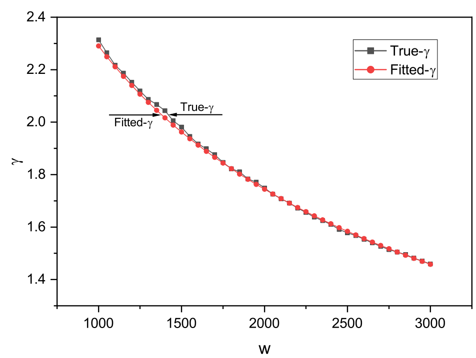


Fig. 5: Comparing fitted γ obtained through the training set with the measured parameter γ obtained through the testing set.

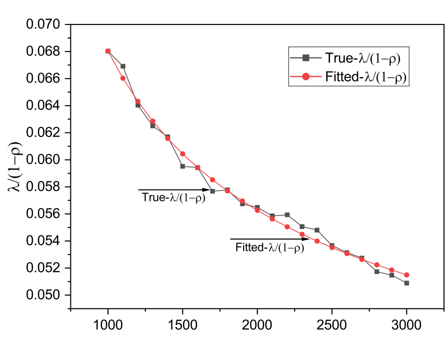


Fig. 6: Comparing fitted $\frac{\lambda}{1-\rho}$ obtained through the training set with the measured parameter $\frac{\lambda}{1-\rho}$ obtained through the testing set.

It is essential to further validate the correctness of t_1 and t_2. With the learned parameters γ and $\frac{\lambda}{1-\rho}$ from the training set, we can calculate the theoretical t_1 and t_2 for the testing set, which are used to compare with the simulated t_1 and t_2. Here, w is varied from 1000 to 3000 at an interval of 50. Both the theoretical and the simulation results are plotted in Figures 7 and 8 for the Overall hours.

From Figure 7, we can see that the difference between the theoretical t_1 and the simulated t_1 is negligible. It manifests the effectiveness of estimating t_1 with our model. Similarly, we can compare the theoretical t_2 with the simulated t_2 in Figure 8. The small gap between the two curves also indicates that it is accurate to estimate t_2 with our developed queuing model.

It is interesting to note that both t_1 and t_2 monotonically decrease with the increase in w. It implies that placing more APs with larger w is an effective way of reducing the uploading delay. However, the convex decreasing curves of the delay suggest that the marginal effect of each incremental AP deployment is decreasing. Thus, it is necessary to find an appropriate w to avoid wasting AP resource.

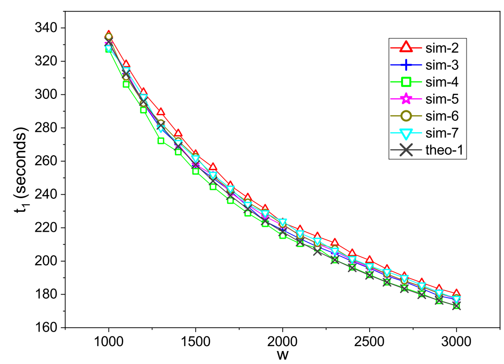


Fig. 7: The accuracy of theoretic t_1 compared to the simulation results.

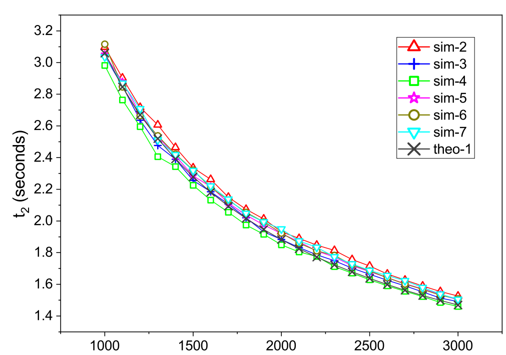


Fig. 8: The accuracy of theoretic t_2 compared to the simulation results.

D. Performance Evaluation

1) Baseline Algorithms: We compare the performance of our solution to that of two baselines: 1) the Uniform algorithm; 2) the Evolutionary algorithm.

- The Uniform algorithm is adopted from [17]. For this algorithm, we first divide each bus route into several sub-routes, which have the same number of bus stops. Then, we apply the algorithm proposed in [17] to solve the set-cover problem for all sub-routes. We named the algorithm Uniform-x if each sub-route contains x bus stops. Uniform-x guarantees that there is at least one AP to cover every x bus stops.
- the Evolutionary algorithm is revised based on [23]. For this algorithm, we set the fitness function the same as our objective function, i.e., $\alpha w + (1-\alpha)(t_1 + t_2)$, which is defined in Eq. 11.

2) Experimental Results: In the first experiment, we compare the performance of our solution to that of the Uniform algorithm. As the Uniform algorithm can only make AP placement with a given number of APs, we fix the number of APs as 605, 900, and 1819 to evaluate the algorithms Uniform-9, Uniform-6, and Uniform-3, respectively. We use the same number of APs for each test case in our solution to keep a fair comparison of the uploading delay with the Uniform algorithm. We evaluate the performance of the two algorithms with trace files from trace-20181102 to trace-20181107, and the experimental results are the average values of the 6 tested days.

The experimental results are presented in Figure 9. In addition to comparing both algorithms during the overall period, we also compare their performance in RH and NRH separately to fully evaluate each algorithm. Fig. 9 indicates that our solution can always achieve low uploading delays for all evaluation cases with an equal number of APs. It is worth mentioning that the uploading delay of both algorithms will be larger during RH and smaller during NRH. The performance of our solution during RH is even better than that of Uniform in NRH, which indicates the superiority of our solution under severe traffic conditions.

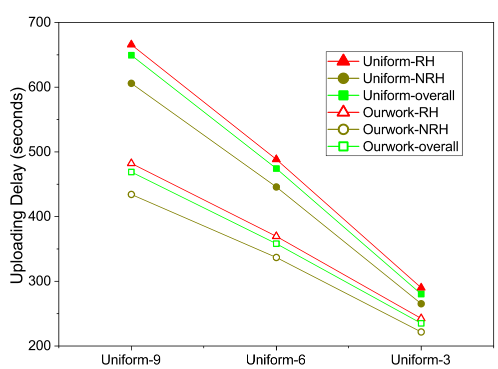


Fig. 9: The performance comparison between our method and the Uniform algorithm in terms of upload delay.

We further compare our solution with the Evolutionary algorithm by varying the parameter α in the harmonic objective function. For the Evolutionary algorithm, we set the mutation rate, the elite rate, the population size, and the number of generations as 0.01, 0.05, 50, and 50, respectively, according to the recommendation in [23].

We compare the obj of our solution to that of the evolutionary algorithm with the same α in the RH of the testing traces, as shown in Figure 10. From the figure, we can see that our solution outperforms the evolutionary algorithm for all testing traces, especially when α becomes larger. When α is 0.21, the obj of our solution is 29.90% lower than that of the evolutionary algorithm on average. Recall that higher α indicates less importance for the objective of minimizing uploading delay, which will cause both our solution and the evolutionary algorithm to deploy less APs. Thus, as α increases, the number of APs and their deployment positions become more critical, where our solution performs better. However, as α decreases, the bus stops will be widely equipped with APs, and thus the problem becomes more trivial.

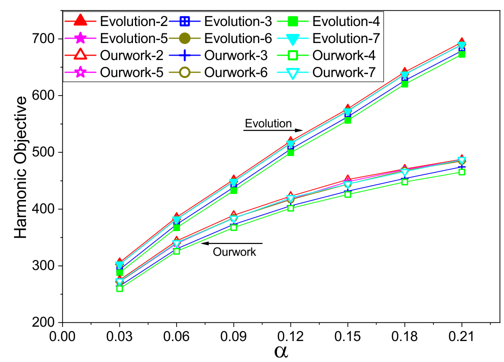


Fig. 10: The performance comparison between our method and the evolutionary algorithm in terms of obj.

VII. Related Work

There have been many works utilizing the power of edge facilities to alleviate the burdens of cellular networks and increase the communication and computation capabilities [24], [25], [26], [27]. Effective RSU deployment strategies have been widely explored by researchers.

A. Linear Programming Based Solutions

Linear Programming (LP) has been widely used for the modeling of the RSU deployment problem, which was further solved by tools such as CPLEX or by more efficient heuristic solutions.

In [28], Wang et al. deployed RSUs in a 2-D VANET and conducted experiments in both urban and suburban road scenarios. They proposed a LP model to maximize the total centrality of candidate positions and solved the problem with the 0-1 Knapsack algorithm. However, they only considered road topology information with degree centrality and closeness centrality without considering the traffic conditions, and simply assumed a constant vehicle speed.

Eftekhari et al. [29] proposed a Binary Programming (BP) model to deploy RSUs for a 1-D road scenario, which incorporated information such as accident rates and was solved by the CPLEX tool. Works, such as [30], [31], [32], leveraged LP to model the RSU deployment problem, aiming to make a trade off between the cost of RSU placement and the coverage ratio.

B. Genetic Algorithm Based Solutions

There are works solving the RSU deployment problem based on Genetic Algorithm (GA).

In [23], Moura et al. modeled the RSU deployment problem in VANETs as Maximum Coverage with Time Threshold Problem (MCTTP). They performed a preprocessing based on the betweenness centrality and the community detection method to improve the convergence time, and solved the problem based on GA. Mehar et al. [33] improved the end-to-end application delay and reduced the deployment costs by placing RSUs in vehicular networks. They filtered out the RSU candidates based on the connectivity information, and utilized GA and Dijkstra algorithm to reduce the number of RSUs.

C. Heuristic Algorithm Based Solutions

In [16], Trullols et al. deployed Dissemination Points (DP) in a Vehicle-to-Infrastructure (V2I) communication scenario, where the movement of vehicles was simulated with VanetMobiSim, aiming to maximize the number of vehicles connected to DPs. They first maximized the number of contacted vehicles assuming small message exchange, and then considered the contacted duration without the assumption. A greedy algorithm and a time-subzone algorithm were used to improve the coverage ratio and the coverage time. Zhao et al. [34] proposed a novel three-phase deployment approach to provide real-time processing service and reduce the required number of edge nodes (ENs). They constructed a utility metric for candidate positions and proposed a heuristic algorithm to select positions with high utilities.

D. Clustering in Deployment Problem

Clustering has been used to reduce the search space or simplify model construction in facility deployment problem [35]. In [36], Ni et al. proposed a LP-based clustering algorithm to jointly improve the deployment of RSUs and assignment of service tasks for the 2-D Internet-of-Vehicle (IoV) networks. They used the LP-based clustering algorithm to group the RSUs into different clusters, then greedily solved sub-problems for each cluster, and finally combined the solutions of all clusters. Cao et al. [37] optimized the deployment of RSUs by considering the deployment cost and the latency. They firstly chose the RSU candidates based on the information of road topology and traffic data, such as traffic flow in intersections. Then, they used the branch-and-bound algorithm to reduce the number of RSUs based on K-nearest neighbor algorithm (KNN).

E. Summary

Previous works attempted to increase the coverage ratio of RSUs or ensure timely contacts between vehicles and RSUs, while limiting the deployment cost. These works justified the important role of RSUs in enabling inter-connection of vehicles. However, they focused on the small content transmission, such as small text messages, safety messages, or control signals [38], [39], ignoring the sizes of the content. On the contrary, we consider the uploading of continuously generated surveillance video of moving buses, which is extremely bandwidth consuming and cannot be effectively solved by existing approaches. We propose WFP to balance the aggregation bandwidth of different buses, and combine a queuing model with machine learning models to analyze the uploading delay so as to optimize the AP budget.

VIII. Conclusion

In this work, we investigated the surveillance video uploading problem for buses in public transportation systems. A novel strategy that uploads video content via APs deployed at stops was designed. Firstly, the deployment cost and the uploading delay were analyzed respectively. A WFP algorithm was proposed to balance bandwidth allocation, and a queuing model was established to analyze the uploading delay. To factor in the effect of bus routes, machine learning models were leveraged to learn two key parameters in the queuing model. Then, a convex optimization problem was formulated to minimize the harmonic objective that takes both the uploading delay and the placement cost into account. Real traces collected from Shenzhen city of China were used for carrying out experiments. The experimental results not only validated the correctness of our analysis but also showed the effectiveness of our strategy. In the future, we will extend our strategy to make it applicable for other public transportation systems such as taxis.

References

[1] Z. Che, G. Li, T. Li, B. Jiang, X. Shi, X. Zhang, Y. Lu, G. Wu, Y. Liu, and J. Ye, "D^2-city: A large-scale dashcam video dataset of diverse traffic scenarios," *arXiv preprint arXiv:1904.01975*, 2019.

[2] H. Salmane, L. Khoudour, and Y. Ruichek, "A video-analysis-based railwayroad safety system for detecting hazard situations at level crossings," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 596–609, April 2015.

[3] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, "Driver behavior analysis for safe driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3017–3032, Dec 2015.

[4] Z. Xu, Y. Liu, L. Mei, C. Hu, and L. Chen, "Semantic based representing and organizing surveillance big data using video structural description technology," *Journal of Systems and Software*, vol. 102, pp. 217 – 225, 2015.

[5] Y. Wu, F. Hu, G. Min, and A. Y. Zomaya, *Big data and computational intelligence in networking*. CRC Press, 2017.

[6] E. Evdokimova, A. Vinel, N. Lyamin, and D. Fiems, "Internet provisioning in vanets: Performance modeling of drive-thru scenarios," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2019.

[7] L. Yang, L. Zhang, Z. He, J. Cao, and W. Wu, "Efficient hybrid data dissemination for edge-assisted automated driving," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 148–159, Jan 2020.

[8] J. He, Y. Ni, L. Cai, J. Pan, and C. Chen, "Optimal dropbox deployment algorithm for data dissemination in vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 632–645, March 2018.

[9] B. Bellalta and K. Kosek-Szott, "Ap-initiated multi-user transmissions in ieee 802.11 ax wlans," *Ad Hoc Networks*, vol. 85, pp. 145–159, 2019.

[10] B. Bellalta, "Ieee 802.11 ax: High-efficiency wlans," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.

[11] Z. El Khaled, H. Mcheick, and F. Petrillo, "Wifi coverage range characterization for smart space applications," in *2019 IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT)*. IEEE, 2019, pp. 61–68.

[12] A. M. M. Ali, N. M. Ahmad, and A. H. M. Amin, "Cloudlet-based cyber foraging framework for distributed video surveillance provisioning," in *2014 4th World Congress on Information and Communication Technologies (WICT 2014)*, Dec 2014, pp. 199–204.

[13] P. H. Calamai and J. J. Moré, "Projected gradient methods for linearly constrained problems," *Mathematical programming*, vol. 39, no. 1, pp. 93–116, 1987.

[14] M. Rashidi, I. Batros, T. K. Madsen, M. T. Riaz, and T. Paulin, “Placement of road side units for floating car data collection in highway scenario,” in *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, Oct 2012, pp. 114–118.

[15] M. A. Ergin, K. Ramachandran, and M. Gruteser, “Understanding the effect of access point density on wireless lan performance,” in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*. New York, NY, USA: Association for Computing Machinery, 2007, pp. 350–353.

[16] O. Trullols, M. Fiore, C. Casetti, C.-F. Chiasserini, and J. B. Ordinas, “Planning roadside infrastructure for information dissemination in intelligent transportation systems,” *Computer Communications*, vol. 33, no. 4, pp. 432–442, 2010.

[17] R. S. Garfinkel, “Integer programming,” New York, NY, USA: Wiley, Tech. Rep., 1972.

[18] V. Chvatal, “A greedy heuristic for the set-covering problem,” *Mathematics of operations research*, vol. 4, no. 3, pp. 233–235, 1979.

[19] H. Kim, Y. Cheng, and G. Chang, “Variable signal progression bands for transit vehicles under dwell time uncertainty and traffic queues,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 109–122, 2019.

[20] K. Zhang, S. Leng, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, “Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1987–1997, April 2019.

[21] M. Zukerman, “Introduction to queueing theory and stochastic teletraffic models,” *arXiv preprint arXiv:1307.2968*, 2013.

[22] P. H. Calamai and J. J. Moré, “Projected gradient methods for linearly constrained problems,” *Mathematical programming*, vol. 39, no. 1, pp. 93–116, 1987.

[23] D. L. Moura, R. S. Cabral, T. Sales, and A. L. Aquino, “An evolutionary algorithm for roadside unit deployment with betweenness centrality preprocessing,” *Future Generation Computer Systems*, vol. 88, pp. 776–784, 2018.

[24] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, “Edge intelligence and blockchain empowered 5g beyond for the industrial internet of things,” *IEEE Network*, vol. 33, no. 5, pp. 12–19, Sep. 2019.

[25] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, “Deep learning empowered task offloading for mobile edge computing in urban informatics,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7635–7647, Oct 2019.

[26] Z. Wang, L. Sun, M. Zhang, H. Pang, E. Tian, and W. Zhu, “Propagation- and mobility-aware d2d social content replication,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 1107–1120, 2016.

[27] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, “Understanding performance of edge content caching for mobile video streaming,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1076–1089, 2017.

[28] Z. Wang, J. Zheng, Y. Wu, and N. Mitton, “A centrality-based rsu deployment approach for vehicular ad hoc networks,” in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–5.

[29] H. R. Eftekhari, A. J. Bashirzadeh, and M. Ghatee, “Binary programming model to optimize rsu placement for information dissemination,” in *2015 International Conference on Connected Vehicles and Expo (ICCVE)*, Oct 2015, pp. 112–115.

[30] N. M. Balouchzahi, M. Fathy, and A. Akbari, “Optimal road side units placement model based on binary integer programming for efficient traffic information advertisement and discovery in vehicular environment,” *IET Intelligent Transport Systems*, vol. 9, no. 9, pp. 851–861, 2015.

[31] Y. Liang, H. Liu, and D. Rajan, “Optimal placement and configuration of roadside units in vehicular networks,” in *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, May 2012, pp. 1–6.

[32] N. Nikookaran, G. Karakostas, and T. D. Todd, “Combining capital and operating expenditure costs in vehicular roadside unit placement,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7317–7331, Aug 2017.

[33] S. Mehar, S. M. Senouci, A. Kies, and M. M. Zoulikha, “An optimized roadside units (rsu) placement for delay-sensitive applications in vehicular networks,” in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2015, pp. 121–127.

[34] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, and Q. Ni, “Deploying edge computing nodes for large-scale iot: A diversity aware approach,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3606–3614, Oct 2018.

[35] G. Wang, Y. Zhao, J. Huang, and Y. Wu, “An effective approach to controller placement in software defined wide area networks,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 344–355, March 2018.

[36] Y. Ni, J. He, L. Cai, J. Pan, and Y. Bo, “Joint roadside unit deployment and service task assignment for internet of vehicles (iov),” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3271–3283, 2018.

[37] X. Cao, Q. Cui, S. Zhang, X. Jiang, and N. Wang, “Optimization deployment of roadside units with mobile vehicle data analytics,” in *2018 24th Asia-Pacific Conference on Communications (APCC)*, Nov 2018, pp. 358–363.

[38] T. Wen, C. Constantinou, L. Chen, Z. Tian, and C. Roberts, “Access point deployment optimization in cbtc data communication system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1985–1995, June 2018.

[39] L. Cheng, J. Liu, G. Xu, Z. Zhang, H. Wang, H.-N. Dai, Y. Wu, and W. Wang, “Sctsc: A semicentralized traffic signal control mode with attribute-based blockchain in iovs,” *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1373–1385, 2019.

APPENDIX

Proof of Proposition 1: It is trivial to show that the harmonic objective decreases with the increase of w because λ decreases while γ increases with the increase of w.

To prove the harmonic objective is a convex function with w, we turn to prove that t_1 and t_2 are convex functions with w respectively.

Define the composite function $f\left(g(w)\right)$ where $f = \frac{1}{g(w)}$ and $g = w\gamma(w|\theta)$. Apparently, the difference between $f\left(g(w)\right)$ and t_1 is only a constant. By differentiating $f\left(g(w)\right)$ with respect to w, it turns out that

$$\begin{aligned} f\left(g(w)\right)' &= f'\left(g(w)\right) g'(w), \\ f\left(g(w)\right)'' &= f''\left(g(w)\right) \left(g'(w)\right)^2 + f'\left(g(w)\right) g''(w). \end{aligned}$$

According to the definition of f and g, $f' < 0$, $f'' > 0$, $g'' < 0$, and thus $f\left(g(w)\right)'' > 0$, which indicates that t_1 is a convex function with w.

Assuming that we have two convex function $f_1(w) > 0$ and $f_2(w) > 0$ which are monotonically decreasing with w. Then, we prove that $f_1(w)f_2(w)$ is a convex monotonic decreasing function with w. The first and second order differentials of $f_1(w)f_2(w)$ with respect to w are

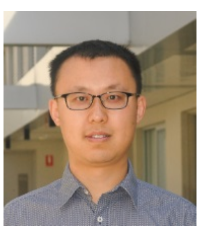
$$\begin{aligned} \left(f_1(w)f_2(w)\right)' &= f_1'(w)f_2(w) + f_1(w)f_2'(w), \\ \left(f_1(w)f_2(w)\right)'' &= f_1''(w)f_2(w) + 2f_1'(w)f_2'(w) \\ &\quad + f_1(w)f_2''(w). \end{aligned}$$

Since $f_1(w)$ and $f_2(w)$ are decreasing function with w, we have $f_1'(w) < 0$ and $f_2'(w) < 0$, and thus $\left(f_1(w)f_2(w)\right)' < 0$. It implies that $f_1(w)f_2(w)$ is a decreasing function. If $f_1(w)$ and $f_2(w)$ are convex, we have $f_1(w) \geq 0$ and $f_2''(w) \geq 0$, which implies that $\left(f_1(w)f_2(w)\right)'' \geq 0$ and thus $f_1(w)f_2(w)$ is a convex function.

t_2 can be decomposed as a product of a constant and $\frac{\lambda}{1-\rho}\frac{1}{\gamma}\frac{1}{\gamma}$. Due to the fact that $\frac{\lambda}{1-\rho}$ and $\frac{1}{\gamma}$ are convex decreasing functions of w, t_2 is a convex monotonic decreasing function with the increase of w. ■

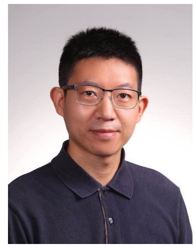
Laizhong Cui is currently a Professor in the College of Computer Science and Software Engineering at Shenzhen University, China. He received the B.S. degree from Jilin University, Changchun, China, in 2007 and Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2012. His research interests include Future Internet Architecture and protocols, Edge Computing, Multimedia Systems and Applications, Blockchain, Internet of Things, Cloud and Big Data Computing, Software-Defined Network, Social Network, Computational Intelligence and Machine Learning. He led more than 10 scientific research projects, including National Key Research and Development Plan of China, National Natural Science Foundation of China, Guangdong Natural Science Foundation of China and Shenzhen Basic Research Plan. He has published more than 70 papers, including IEEE Transactions on Multimedia, IEEE IoT Journal, IEEE Transactions on Industrial Informatics, IEEE Transactions on Vehicular Technology, IEEE Transactions on Network and Service Management, ACM Transactions on Internet Technology, IEEE Transactions on Computational Biology and Bioinformatics and IEEE Network. He serves as an Associate Editor or a Member of Editorial Board for several international journals, including International Journal of Machine Learning and Cybernetics, International Journal of Bio-Inspired Computation, Ad Hoc and Sensor Wireless Networks and Journal of Central South University. He is a Senior Member of the IEEE, and a Senior Member of the CCF.

Dongyuan Su received the B.E. degree in computer science from the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, in 2019, where he is currently working toward the Master's degree. His research interests include edge computing and machine learning.



Yipeng Zhou is a lecturer in computer science with Department of Computing at Macquarie University, and the recipient of ARC DECRA in 2018. From Aug. 2016 to Feb. 2018, he was a research fellow with Institute for Telecommunications Research (ITR) ofUniversity of South Australia. From 2013.9-2016.9, He was a lecturer with College of Computer Science and Software Engineering, Shenzhen University. He was a Postdoctoral Fellow with Institute of Network Coding (INC) of The Chinese University of Hong Kong (CUHK) from Aug. 2012 to Aug. 2013. He won his PhD degree and Mphil degree from Informatio Engineering (IE) Department of CUHK respectively. He got Bachelor degree in Computer Science from University of Science and Technology of China (USTC).

Lei Zhang (S'12-M'19) received the B.Eng. degree from the Advanced Class of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2011, and the M.S. degree and the Ph.D degree from Simon Fraser University, Burnaby, BC, Canada, in 2013 and 2019, respectively. He is a recipient of C.D. Nelson Memorial Graduate Scholarship (2013) and Best Paper Finalist at IEEE/ACM IWQoS (2016). He is currently an Assistant Professor at the College of Computer Science and Software Engineering, Shenzhen University. His research interests include multimedia systems and applications, mobile cloud computing, edge computing, social networking, and Internet of Things.



Yulei Wu is a Senior Lecturer with the Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, United Kingdom. He received the B.Sc. degree (First Class Honours) in Computer Science and the Ph.D. degree in Computing and Mathematics from the University of Bradford, United Kingdom, in 2006 and 2010, respectively. His expertise is on networking and his main research interests include computer networks, networked systems, software defined networks and systems, network management, and network security and privacy. Dr. Wu contributes to major conferences on networking and networked systems as various roles, including the Steering Committee Chair, the General Chair and the Program Chair. His research has been supported by Engineering and Physical Sciences Research Council of United Kingdom, National Natural Science Foundation of China, University's Innovation Platform and industry. He is an Editor of IEEE Transactions on Network and Service Management, Computer Networks (Elsevier) and IEEE Access. He is a Senior Member of the IEEE, and a Fellow of the HEA (Higher Education Academy).

Shiping Chen is a principal research scientist in CSIRO Data61. He also holds an conjoint A/Professor title with the University of New South Wales (UNSW) and the University of Sydney through teaching and supervising PhD students. He has been working on distributed systems for over 20 years with focus on performance and security. He has published 180+ research papers and technical reports in these areas. He has been actively participating in research communities through publishing papers, journal editorships and conference PC/Chair services, including WWW, EDOC, ICSOC and IEEE ICWS/SCC/CLOUD etc. His current research interests include: application security, blockchain and service-oriented trusted collaboration. He is a senior member of the IEEE.