

## Article

# Deadline-Aware Scheduling for Transmitted RSU Packets in Cooperative Vehicle-Infrastructure Systems

Beipo Su, Yongfeng Ju and Liang Dai \*

School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, China

\* Correspondence: ldai@chd.edu.cn; Tel.: +86-136-0929-3915

**Abstract:** In Cooperative Vehicle Infrastructure System (CVIS), the roadside unit (RSU) obtains many kinds of monitoring data through observation equipment carried by the RSU. The monitoring data from RSUs are transmitted to an RSU that is connected to the backbone network using the “store-carry-forward” scheme through the mobile vehicle. The monitoring data obtained by RSUs are timely, and different types of monitoring data have corresponding timelines. Reducing end-to-end delays to ensure more packets can be transmitted before deadlines is challenging. In this paper, we propose a Distributed Packet Scheduling Scheme for Delay-Packets Queue Length Tradeoff System (DDPS) in CVIS to solve the multi-RSU-distributed packet transmission problem. We also establish the vehicle speed state, vehicle communication quantity prediction, data arrival, and end-to-end delay minimization models. After Lyapunov’s optimization theory transformed the optimization model, a knapsack problem was described. The simulation results verified that DDPS reduced the end-to-end average delay and ensured the data queue’s stability under packet deadline conditions.

**Keywords:** cooperative vehicle infrastructure system; roadside unit; deadline; store-carry-forward; Lyapunov



**Citation:** Su, B.; Ju, Y.; Dai, L. Deadline-Aware Scheduling for Transmitted RSU Packets in Cooperative Vehicle-Infrastructure Systems. *Appl. Sci.* **2023**, *13*, 4329. <https://doi.org/10.3390/app13074329>

Academic Editors: Enrico Vezzetti, Subhas Mukhopadhyay and Yoshiyasu Takefuji

Received: 9 February 2023

Revised: 16 March 2023

Accepted: 27 March 2023

Published: 29 March 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Highways are important to the research and application of CVIS in intelligent highways [1,2]. Deploying multiple RSUs along the highway ensures vehicles can access the Internet more easily. RSUs can also collect and forward observation data within their range. Some RSUs are built in remote areas, such as mountain forests or the Gobi, which makes Internet connectivity difficult and costly [3]. Therefore, isolated RSUs must carry data through mobile vehicles to reach the RSU access range to the backbone network and then upload the data. RSU observation data include traffic conditions, environmental monitoring, natural disasters, and animal activity information [4–6]. RSU collects monitoring data in a timely manner, and different data types have different delay deadline requirements [7]. High-speed moving vehicles cause the time-varying network topology of vehicles. An unnecessary impact is caused when the data cannot be fed back to the relevant departments before the delay deadline. Therefore, the main goal of this paper is to determine the optimal scheduling strategy considering that monitoring data have different deadlines and data queues are stable, minimizing the end-to-end average delay.

## 2. Related Work and Contributions

The high mobility of vehicles in CVIS creates a time-varying network topology. Its communication has intermittent connectivity, which increases not only opportunities for data distribution and network capacity, but also data transmission delays. It is more suitable for delay-tolerant services.

Kyoungsoo et al. [8] proposed an RSU scheduling that provides V2I-based data services through multiple RSUs. The proposed scheme connects the RSU through a wired backbone network. The scheme adopts a cooperation strategy of multiple RSUs to transmit secure and non-secure data, reducing the deadline failure rate and average response

time. Guo et al. [9] proposed a V2R data access scheme based on data integrity and importance. Based on the vehicle request's initial information, they first estimated whether it could avoid invalid data access requests. On this basis, the author proposed a multi-request priority scheduling scheme considering the importance of requests and deadlines. Finally, the author used a priority-based strategy to satisfy these requests.

TAN et al. [10] proposed studying the problem of efficient content broadcasting through vehicle networks. They expressed it as a joint optimization problem of data prefetching and broadcast transmission scheduling with buffer size constraints and transmission rate adaptive constraints. This article considers a highway scenario covered by multiple RSUs and the deadline requirements of interactive applications. The author used a heuristic algorithm to solve the integer linear programming problem.

The block must be delivered within the playback period to achieve the desired video quality during playback. Bethanabhotla et al. [11] proposed a network utility maximization (NUM) problem where the network utility function is a concave, component-by-component, non-decreasing function of the video quality index requested by time-averaged users. Maximization depends on the stability of all queues in the system.

Due to the lack of an average delay performance guarantee in existing research, Yang et al. [12] focused on the key goal of delivering next-generation real-time services before the corresponding deadline of each packet while minimizing the overall cloud network resource cost, introducing a novel queueing system, which can track the life cycle of data packets, and formalizing the optimal cloud network control problem with strict deadline constraints. Having explained the main challenge of sending packets to their destination before their life cycle expires, the author developed an equivalent formula. The relaxed flow conservation allowed the use of Lyapunov optimization to derive a provable, near-optimal, and fully distributed algorithm for the original problem.

Hu et al. [13] proposed an auction bidding scheme to determine the RSU in response to the computing request. In other words, auctioning the computing request to obtain bids from the least energy consuming RSUs. The scheme works in a decentralized model, which effectively reduces the complexity of its implementation. In order to process computing requests simulated as DAG applications, the DAG is divided into individual tasks using upper-order values. They also proposed a deadline-aware queue jump algorithm assigned to server queues in specific RSUs.

When tasks require different delay deadlines, computing resource allocation becomes more challenging. Mukherjee et al. [14] proposed a scheduling strategy to maximize the number of tasks completed within their deadlines and stabilize the network. The Lyapunov drift plus penalty function on the queue length was used to schedule tasks. Sethi et al. [15] proposed a scheduling architecture that minimizes RSUs' energy consumption and achieves uniform energy consumption between adjacent RSUs. In turn, this scheme increases the request fulfillment percentage for RSUs. The proposed architecture classifies incoming requests as traditional (with less computation) or smart requests (with more computation). They proposed a hard-deadline low-computing-requirement method (HLCA) and a soft-deadline high-computing-requirement method (SHCA) aimed at traditional and intelligent data requests, respectively.

Wu et al. [16] proposed the random unloading problem of minimizing long-term energy consumption. They used the perturbed Lyapunov optimization technique to construct a virtual queue and transformed the task deadline guarantee problem into a virtual queue stability control problem. They proposed an energy-saving online unloading algorithm for DelayAware, which can adaptively unload more tasks under good network quality conditions. Additionally, the transmission is delayed when the connection is poor, but the deadline is not compromised.

Xiang et al. [17] considered the delay sensitivity of different vehicles to receive early warning information and studied the release of emergency information using V2X communication. We divided the vehicles covered by the roadside unit (RSU) into primary and secondary priority groups according to the distance between the vehicle and the emergency

point. A joint optimization problem of content division, user groupings, and channel allocation was proposed to improve resource utilization and emergency message delivery efficiency. The goal was to ensure that all vehicles in the primary priority group could reliably receive warning messages within a fixed deadline. In addition, RSUs could send countless warning messages to vehicles in the secondary priority group.

The main contributions of this paper are as follows:

We proposed an optimization strategy based on a vehicle carrier relay to minimize end-to-end delays of RSU packets. This strategy decides packet scheduling using the high- and low-priority data cache queue and vehicle speed status of multiple RSUs. According to the queue and speed states of multiple RSUs, this strategy dynamically selects the data cache queue and the number of packets sent to ensure system performance and minimize end-to-end delays.

The packet scheduling problem is transformed into a knapsack problem, and simulation results verify the algorithm's effectiveness.

The remaining organizational structure of this paper is as follows. Section 3 describes the packet scheduling strategy and system model. In Section 4, Lyapunov optimization theory transforms the model, and the knapsack problem is solved. In Section 5, the performance of the proposed optimization strategy is evaluated and compared to existing algorithms. The superiority of the proposed scheme is renamed here. Section 6 contains the conclusion of this paper.

### 3. System Model

On the highway section of  $L$ -length, transferring data packets to  $RSU'$  through a single RSU may cause the following problems:

1. Considering the maximum capacity of the data cache queue of the RSU, when the monitoring data of the road section is stored in the data cache queue of the RSU, the data cache queue of the RSU will be full quickly, and the data packets that arrive later will overflow, resulting in the loss of available data packets.
2. When the arrival rate of vehicles on the road section is high, a single RSU only sends data packets to one vehicle in each time slot, which may cause multiple vehicles to have no data packets to carry, so the stability of the data cache queue can't be guaranteed.
3. When the data packet arrival rate is high, considering that there are fewer data packets that can be sent in each time slot of a single RSU, the data cache queue length of the RSU will remain in a higher state, and the higher data queue length will lead to larger average queuing delay, which will affect the overall performance of the system.

In this paper, we consider that  $N$  RSUs are evenly distributed on the highway section of  $L$ -length. each RSU observes its coverage and collects data and stores it in its data cache queue. DDPS makes data packet scheduling decisions according to the corresponding states of  $N$  RSU in each time slot to avoid the above problems.

As shown in Figure 1, RSUs are deployed on highways.  $N$  RSUs in remote areas, such as mountains and forests, which consider economic benefits and other factors, cannot access the backbone network. Therefore, many RSUs deployed in remote areas use mobile vehicles as transmission media to forward data packets from mobile vehicle carriers to backbone networks. The data cache queue of RSUs is divided into emergency and non-emergency data cache queues. Emergency data mainly include extreme weather, forest fires, and static obstacles on the road. Non-emergency data are mainly daily traffic, environmental data, and other monitoring data.

Based on the speed and queue status, scheduling determines whether data packets are sent, how many, and what type of data packets are sent to vehicles. The data should be transmitted to the backbone before the deadline to ensure the relevant departments can handle the consequences of emergency and non-emergency data. In remote areas,  $l$  is the distance between two RSUs, and  $L'$  is the shortest distance between the destination node  $RSU'$  and the isolated RSUs. Since  $L'$  is much larger than the communication

range of RSU, the latter can be ignored. RSU indexes of  $n \in \{1, 2, \dots, N\}$  are deployed in the scenario of L-length highway sections. The system time is discretized, the index is  $t \in \{0, 1, \dots, T\}$ , and the length of each slot is represented by  $\tau$ . When the vehicle enters the coverage area of  $RSU_n$ ,  $RSU_n$  unloads the collected data packets onto it. Using the data scheduling controller,  $RSU_n$  determines a scheduling strategy for processing data packets in the data queues. Figure 2 shows that the data transmission control model comprises vehicle speed state  $y_n(t)$ , data queues length  $Q_n^H(t), Q_n^L(t)$ , packet arrival number  $h_n(t)$ , queue selection decision  $b_n(t)$ , and data packet transmission decision  $d_n(t)$ .

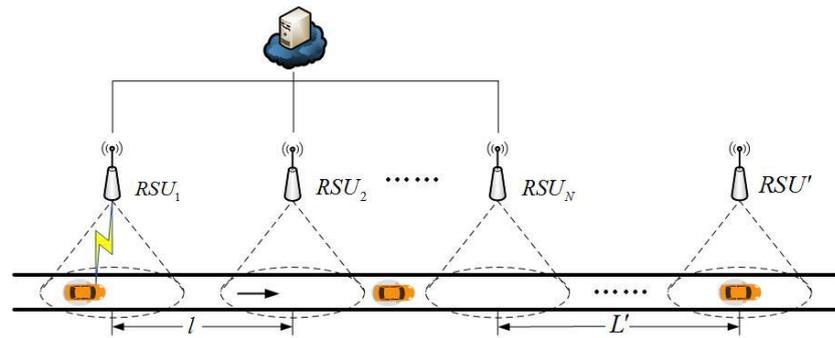


Figure 1. RSU packet scheduling for CVIS.

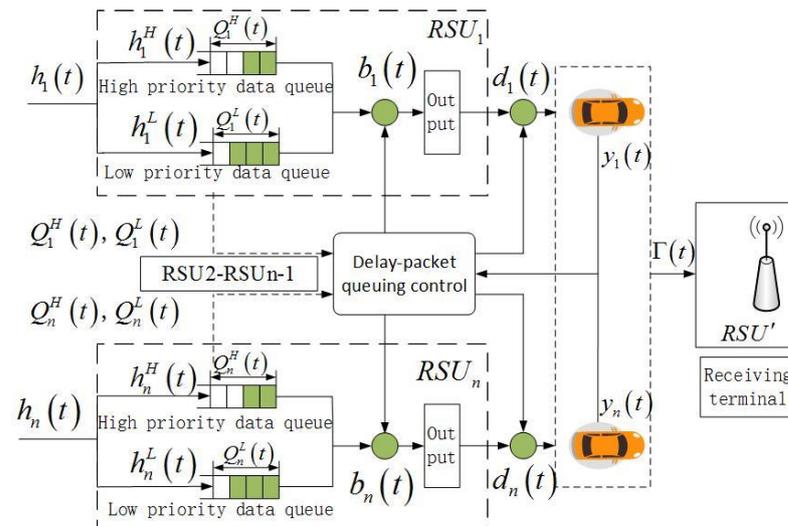


Figure 2. Packet scheduling model.

### 3.1. Vehicle Speed State Model

We only considered the one-lane scenario of the highway, where the vehicle is in a free-flow traffic date and its arrival obeys the Poisson distribution with parameter  $\lambda$ . Therefore, we can conclude that the arrival time interval  $\Delta t$  of any two cars obeys a negative exponential distribution. Its probability density function is as follows:

$$f(t) = \lambda e^{-\lambda t} (t > 0) \tag{1}$$

The probability distribution function is derived as

$$F(t) = P(\Delta t \leq t) = 1 - e^{-\lambda t} (t > 0) \tag{2}$$

Therefore, the probability of at least one car arriving at this section in the  $\tau$  time slot is

$$P_v = P(\Delta t \leq \tau) = 1 - e^{-\lambda \tau} (\tau > 0) \tag{3}$$

Then, the probability of no vehicles arriving in the  $\tau$  time slot is  $1 - P_v$ .

In this paper, we considered  $v(t)$  as the speed of the vehicle when it arrives in the  $t$ -slot and assumed that the vehicle's speed in the road section does not change. In the free flow traffic state, the speed  $v(t)$  obeys the normal distribution with an average value of  $\bar{V}$  and a standard deviation of  $\sigma$ . Its probability density function is

$$f_v(v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{v-\bar{V}}{\sigma\sqrt{2}}\right)^2} \tag{4}$$

If the speed limit interval of the highway is expressed as  $[V_{\min}, V_{\max}]$ , then the truncation probability density function of the speed distribution is

$$f_v^*(v) = \frac{2f_v(v)}{\operatorname{erf}\left(\frac{V_{\max}-\bar{V}}{\sigma\sqrt{2}}\right) - \operatorname{erf}\left(\frac{V_{\min}-\bar{V}}{\sigma\sqrt{2}}\right)} \tag{5}$$

The continuous speed is divided into  $X + 1$  speed states [18]. The state vector is expressed as  $V = [V_1, \dots, V_{X+1}]$ , in which  $V_1 = V_{\min}$  and  $V_X = V_{\max}$  represent the minimum and the maximum speed limits of the highway, respectively, and satisfy  $V_x > V_{x+1}$  ( $x = 1, 2, \dots, X$ ).  $y_n(t)$  indicates the speed the vehicle reaches  $RSU_n$  in the  $t$ -slot.  $y_n(t) = x$  ( $1 \leq x \leq M$ ) indicates  $v(t) \in [V_{x+1}, V_x]$  and  $y_n(t) = X + 1$  indicates that no vehicle will arrive at  $RSU_n$  in the  $t$ -slot.

Let  $\delta_x$  represents the probability that the vehicle is in the speed state  $x$ , and its probability density function expression is

$$\delta_x = P\{y_n(t) = x\} = \begin{cases} P_v \int_{V_{x+1}}^{V_x} f_v^*(v) dv, & 1 \leq x \leq X \\ 1 - P, & x = X + 1 \end{cases} \tag{6}$$

### 3.2. Prediction Model of Vehicle Communication Quantity

In the research scenario of this paper, the connection between vehicles and RSU is not continuous. Therefore, we must determine how many vehicles establish communication links with N RSU. When a vehicle enters the RSU coverage area, its communication state is defined as ON. When the vehicle drives out of the RSU coverage area, its communication state is defined as OFF [19]. In this paper, we analyzed the free flow traffic model to predict the number of vehicles establishing communication links with RSU on the road as  $\kappa(t)$ . We assumed that the probability of the vehicle and establishing communication links obey the discontinuous Bernoulli distribution (IBP) with parameter  $(\lambda, \alpha, \beta)$  [20]. The vehicle's travel time in the ON/OFF state shows a geometric distribution, and the expected values are  $\alpha$  and  $\beta$ , respectively.

The cumulative distribution function of vehicle residence time  $t_r$  on this road section can be expressed as

$$F_R(t_r) = 1 - F_V\left(\frac{L}{t_r}\right) = 1 - \frac{1 + \operatorname{erf}\left(\left(\frac{L}{t_r} - \bar{V}\right) / \sigma\sqrt{2}\right)}{\operatorname{erf}\left(\left(V_{\max} - \bar{V}\right) / \sigma\sqrt{2}\right) - \operatorname{erf}\left(\left(V_{\min} - \bar{V}\right) / \sigma\sqrt{2}\right)} \tag{7}$$

$F_V(L/t_r)$  is the probability distribution function corresponding to the truncated probability density function in the previous section.

The arrival of vehicles obeys the Poisson distribution with parameter  $\lambda$ ; therefore, the probability expression of  $k$  vehicles arriving in the road section within the interval of  $(0, t)$  is as follows:

$$A_k(t) = \frac{(\lambda t)^k e^{-(\lambda t)}}{k!} \tag{8}$$

The  $t$ -slot,  $1 - F_R(t - t_i)$ , represents the probability of any vehicle arriving in the  $t_i$ -slot. The vehicle's arrival obeys the Poisson distribution. Therefore, the vehicle arrival time

distribution depends on whether  $k$  vehicles arrive within the time interval of  $(0, t)$ . This is the same as the uniform distribution of  $k$  points within  $(0, t)$ . The probability expression for the existence of any  $k$  vehicles on the road section within the  $t$ -slot is as follows:

$$p_k(t) = \int_0^t [1 - F_R(t - t_i)] \frac{dt_i}{t} = \frac{1}{t} \int_0^t [1 - F_R(t_i)] dt_i \tag{9}$$

The probability of the vehicle leaving the road section in the  $t$ -slot during the time interval of  $(0, t)$  is as follows:

$$1 - p_k(t) = \frac{1}{t} \int_0^t F_R(t_i) dt_i \tag{10}$$

Depending on whether  $k$  vehicles arrive within the time interval of  $(0, t)$ , the probability of  $y$  vehicles driving on the road section within the  $t$ -slot is as follows:

$$P_{y|k}(t) = \begin{cases} C_y^k [p_k(t)]^y [1 - p_k(t)]^{k-y} & , y \leq k \\ 0 & , y > k \end{cases} \tag{11}$$

The probability of establishing a communication link between a vehicle and  $RSU_n$  follows a discontinuous Bernoulli distribution with a parameter of  $(\lambda, \alpha, \beta)$ . Therefore, the probability of  $y$  vehicles driving on the road section in  $t$ -slot and  $r$  vehicles with  $RSU$  coverage is as follows:

$$P\{\kappa(t) = r\} = \frac{\alpha}{\alpha + \beta} \sum_{k=y}^{\infty} C_y^k [p_k(t)]^y [1 - p_k(t)]^{k-y} \cdot \frac{(\lambda t)^k e^{-\lambda t}}{k!} = \frac{\alpha [\lambda t \cdot p_k(t)]^y e^{-(\lambda t) \cdot p_k(t)}}{(\alpha + \beta) y!} \tag{12}$$

### 3.3. Queue Model

The data types are divided into emergency and non-emergency data. According to the type of data arrival, the data cache queue of the vehicle is divided into high- and low-priority data queues. Let  $b_n(t)$  denote the data cache queue selection decision within the range of  $RSU_n$  in the  $t$ -slot, of which  $b_n(t) \in \{0, 1\}$ .  $b_n(t) = 1$  indicates that the data packets in the high-priority data queue will be transmitted by  $RSU_n$  in the  $t$ -slot.  $b_n(t) = 0$  indicates that the data packets in the low-priority data queue will be transmitted by  $RSU_n$  in the  $t$ -slot.

Considering that the data queue capacity of  $RSU$  is limited, let  $Q_n^H(t + 1)$  denote the high-priority data cache queue length of  $RSU_n$  in the  $t + 1$  slot. Then, the updated data queue expression would be as follows:

$$Q_n^H(t + 1) = \max\{Q_n^H(t) - B_n^H(t) + h_n^H(t), 0\} \tag{13}$$

$h_n^H(t)$  represents the number of emergency data packets arriving at the high-priority data queues of  $RSU_n$  in the time slot  $t$ , which is expressed as  $h_n^H(t) = \vartheta \cdot h_n(t)$ .

$\vartheta$  indicates the probability that the data type arriving is an emergency data packet.  $1 - \vartheta$  indicates the probability that the data type arriving is a non-emergency data packet.

$h_n(t)$  represents the number of data packets that reach  $RSU_n$  in time slot  $t$ . Its probability density function is

$$P\{h_n(t) = u\} = \theta_u, u \in \{0, 1, \dots, U\} \tag{14}$$

where  $\theta_u \in [0, 1]$  represents the probability of  $u$  data packets arriving at  $RSUs$  in the time slot  $t$ . The data arrival distribution satisfies  $\sum_{u=0}^U \theta_u = 1$ , and  $\theta_u = 0, \forall u > U$ . The average

arrival rate of the group is  $\bar{h} = \sum_{u=0}^U u\theta_u$ . When the data queue is full, the newly arrived data packet overflows.

$B_n^H(t)$  is the number of emergency data packets transmitted by  $RSU_n$  in the  $t$ -slot, which is expressed as

$$B_n^H(t) = b_n(t)d_n(t) \tag{15}$$

$d_n(t) \in \{0, 1, \dots, S\}$  represents the number of data packets that  $RSU_n$  sent to the arriving vehicle in the time slot  $t$ , where  $S$  is the maximum number of data packets that  $RSU_n$  can transmit to the vehicle in each time slot,  $d_n(t) = 0$  is when the vehicle speed state is  $X + 1$ , indicating that no vehicle arrived in the current time slot.  $\sum_{n=1}^N d_n(t) \leq S \cdot \kappa(t)$  is satisfied  $\forall n, t$ .

In order to ensure the data cache queue's stability and reduce the data transmission delays, the system designates  $\omega$  as the upper limit of the long-term average data cache queue length. The expression is as follows:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{Q_n^H(t)\} \leq \omega \tag{16}$$

Let  $Q_n^L(t + 1)$  denote the low-priority data cache queue length in the  $t + 1$  slot. The data queue update expression would be as follows:

$$Q_n^L(t + 1) = \max\{Q_n^L(t) - B_n^L(t) + h_n^L(t), 0\} \tag{17}$$

$h_n^L(t)$  represents the number of non-emergency data packets arriving at the low-priority data queues of  $RSU_n$  in the time slot  $t$ , which is expressed as

$$h_n^L(t) = (1 - \vartheta)h_n(t) \tag{18}$$

The newly arrived data packet overflows when the data queue is full.

$B_n^L(t)$  is the number of non-emergency data packets transmitted by  $RSU_n$  in the  $t$ -slot, which is expressed as

$$B_n^L(t) = (1 - b_n(t))d_n(t) \tag{19}$$

To ensure the data cache queue's stability and reduce data transmission delays, the system gives  $\zeta$  as the upper limit of the long-term average data cache queue length. The expression is as follows:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{Q_n^L(t)\} \leq \zeta \tag{20}$$

### 3.4. Average Delay Model

Emergencies such as extreme weather, fires, traffic accidents, etc., are timely and may have a negative impact when not handled properly or within an effective time. Although less serious than the emergency itself, non-emergency events such as animals and daily traffic still reduce the work efficiency of the relevant departments when not treated quickly. Therefore, data packet deadlines must be considered.

In this paper, different data deadlines are divided into emergency  $t_n^H$  and non-emergency data packet deadlines  $t_n^L$  based on the severity of the event. The data packet is discarded when the transmission time is exceeded.

The average end-to-end delays of emergency data packet transmission in the time slot mainly comprise the following two parts:

The average propagation delay  $\bar{T}_{n,c}^H(t)$  is the average time  $RSU_n$  requires sending a packet to the vehicle and carry it to  $RSU'$  in the  $t$ -slot. Its expression is as follows:

$$\bar{T}_{n,c}^H(t) = \frac{L_n + L'}{v_n(t)} \tag{21}$$

$v_n(t)$  represents the average speed at which the vehicle reaches  $RSU_n$  at time slot,  $t$ . The expression is  $v_n(t) = (V_{x+1} + V_x)/2$ .

$L_n$  represents the distance between  $RSU_n$  and  $RSU_N$ , and its expression is  $L_n = (N - n)L$ .

The average queue delay is  $\bar{T}_{n,q}^H(t)$ . According to the emergency data deadline  $t_n^H$ , the relationship can be deduced as  $0 \leq T_{n,q}^H(t) \leq t_n^H - \bar{T}_{n,c}^H(t)$ .

Its expression is as follows:

$$\bar{T}_{n,q}^H(t) = \frac{t_n^H - \bar{T}_{n,c}^H(t)}{2} \tag{22}$$

Therefore, the average end-to-end delay expression for  $RSU_n$  to complete emergency data packet transmission in the  $t$ -slot is as follows:

$$\tau_n^H(t) = \bar{T}_{n,c}^H(t) + \bar{T}_{n,q}^H(t) \tag{23}$$

It can be obtained in the same way that the average end-to-end delay expression for  $RSU_n$  to complete non-emergency data packet transmission in the  $t$ -slot:

$$\tau_n^L(t) = \bar{T}_{n,c}^L(t) + \bar{T}_{n,q}^L(t) \tag{24}$$

Furthermore, the expression of the weighted end-to-end average delay of each data packet from all RSUs in the time slot is as follows:

$$\Gamma(t) = \frac{\sum_{n=1}^N d_n(t) \cdot (b_n(t) \cdot \tau_n^H(t)/t_n^H + (1 - b_n(t)) \cdot \tau_n^L(t)/t_n^L)}{\sum_{n=1}^N d_n(t)} \tag{25}$$

The system's long-term average weighted delay expression is as follows:

$$\Gamma = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{\Gamma(t)\} \tag{26}$$

The optimization problem is as follows:

$$\begin{aligned} \min_{b_n(t), d_n(t)} \Gamma &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{\Gamma(t)\} \\ \text{s.t.} \left\{ \begin{array}{ll} b_n(t) \in \{0, 1\}, 1 \leq n \leq N & a \\ d_n(t) \in \{0, 1, \dots, S\}, 1 \leq n \leq N & b \\ \sum_{n=1}^N d_n(t) \leq S \cdot \kappa(t), \forall n, t & c \\ \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{Q_n^H(t)\} < \omega & d \\ \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{Q_n^L(t)\} < \zeta & e \end{array} \right. \end{aligned} \tag{27}$$

#### 4. Lyapunov Optimization Model

Lyapunov optimization refers to the optimal control of a dynamic system using the Lyapunov function. The Lyapunov function is widely used in control theory to ensure system stability. Multidimensional vectors usually describe a system's state at specific times. The Lyapunov function is a non-negative scalar measure of this multidimensional

state. According to Lyapunov optimization technology, network stability is achieved by initiating the control action that causes Lyapunov drift from negative to zero. Moreover, the weighted (positive) network utility term is subtracted from the Lyapunov drift and minimizes the sum. As a result, the drift plus penalty algorithm for joint network stability and utility maximization is produced.

It is necessary to transform the optimization model and reduce computational complexity because of the large state space of the problem and the high complexity of online optimization. We set the vector matrix  $G(t) = [Q_n^H(t), Q_n^L(t)]$ . The Lyapunov function is defined by this matrix:

$$L(G(t)) = \frac{1}{2} \sum_{n=1}^N [Q_n^H(t)^2 + Q_n^L(t)^2] \tag{28}$$

According to Formula (28), the data queue length is positively correlated with the value of  $L(G(t))$ . Data queue dynamics can be described using the Lyapunov function for differences between adjacent time slots, that is, the Lyapunov drift  $\Delta(G(t))$ , which is expressed as follows:

$$\Delta(G(t)) = E\{L(G(t+1)) - L(G(t)) | G(t)\} \tag{29}$$

System optimization aims to keep the RSU's data queue length low while minimizing end-to-end average delays. Therefore, the end-to-end average delay becomes the drift component in Formula (29), which can be rewritten as

$$\Delta(G(t)) + VE\{\Gamma(t) | G(t)\} \tag{30}$$

$V$  is the tradeoff parameter between data queue length and end-to-end average delays. If the  $V$  value increases, the weight of the end-to-end average delay in the Lyapunov drift parameter increases. In turn, the drift effect on the system increases. It makes the system focus on reducing the end-to-end average delay, but the length of the data queue increases accordingly. Therefore, the system can be optimized by adjusting the value of  $V$ . System optimization is possible by minimizing the upper bound of the Lyapunov drift.

According to the theorem,  $\{\max[(a - b + c), 0]\}^2 \leq (a - b + c)^2$  is true for  $\forall a, b, c > 0$ ; therefore, the following inequality is derived from Formula (13).

$$\frac{1}{2} Q_n^H(t+1)^2 \leq \frac{1}{2} Q_n^H(t)^2 + \frac{1}{2} (h_n^H(t) - B_n^H(t))^2 + Q_n^H(t) (h_n^H(t) - B_n^H(t)) \tag{31}$$

It can be obtained in the same way:

$$\frac{1}{2} Q_n^L(t+1)^2 \leq \frac{1}{2} Q_n^L(t)^2 + \frac{1}{2} (h_n^L(t) - B_n^L(t))^2 + Q_n^L(t) (h_n^L(t) - B_n^L(t)) \tag{32}$$

According to Formula (28), the following relational expression can be derived:

$$\begin{aligned} \Delta(G(t)) \leq & \frac{1}{2} \sum_{n=1}^N E\{(h_n^H(t) - B_n^H(t))^2 | G(t)\} + \sum_{n=1}^N Q_n^H(t) E\{(h_n^H(t) - B_n^H(t)) | G(t)\} \\ & + \frac{1}{2} \sum_{n=1}^N E\{(h_n^L(t) - B_n^L(t))^2 | G(t)\} + \sum_{n=1}^N Q_n^L(t) E\{(h_n^L(t) - B_n^L(t)) | G(t)\} \end{aligned} \tag{33}$$

Through  $\alpha \cdot \bar{h} = E[h_n^H(t)]$ ,  $(1 - \alpha) \cdot \bar{h} = E[h_n^L(t)]$ , and  $B_n(t) = \{B_n^H(t), B_n^L(t)\} \leq S$ , we can obtain the following relations:

$$\begin{aligned} & \sum_{n=1}^N E\{(h_n^H(t) - B_n^H(t))^2 | G(t)\} + \sum_{n=1}^N E\{(h_n^L(t) - B_n^L(t))^2 | G(t)\} \\ & \leq \sum_{n=1}^N \max\left\{(\alpha \cdot \bar{h})^2, (S - \alpha \cdot \bar{h})^2\right\} + \sum_{n=1}^N \max\left\{[(1 - \alpha) \cdot \bar{h}]^2, [S - (1 - \alpha) \cdot \bar{h}]^2\right\} \end{aligned} \tag{34}$$

Through  $d_n(t) \in \{0, 1, \dots, S\}$ , we can obtain the following relation:

$$\begin{aligned} & \sum_{n=1}^N [d_n(t) \cdot (b_n(t) \cdot \tau_n^H(t)/t_n^H + (1 - b_n(t)) \cdot \tau_n^L(t)/t_n^L)] / \sum_{n=1}^N d_n(t) \\ & \leq \sum_{n=1}^N [d_n(t) \cdot (b_n(t) \cdot \tau_n^H(t)/t_n^H + (1 - b_n(t)) \cdot \tau_n^L(t)/t_n^L)] \end{aligned} \tag{35}$$

If parameter  $Cons = N \left\{ \max \left\{ (\alpha \cdot \bar{h})^2, (S - \alpha \cdot \bar{h})^2 \right\} + \max \left\{ [(1 - \alpha) \cdot \bar{h}]^2, [S - (1 - \alpha) \cdot \bar{h}]^2 \right\} \right\} / 2$  is substituted with inequality (34), then the Lyapunov drift function satisfies the following inequality relation:

$$\begin{aligned} \Delta(G(t)) + VE\{\Gamma(t)|G(t)\} & \leq Cons + \sum_{n=1}^N Q_n^H(t) E\{ (h_n^H(t) - B_n^H(t)) | G(t) \} \\ & + \sum_{n=1}^N Q_n^L(t) E\{ (h_n^L(t) - B_n^L(t)) | G(t) \} \\ & + VE\{ [d_n(t) \cdot (b_n(t) \cdot \tau_n^H(t)/t_n^H + (1 - b_n(t)) \cdot \tau_n^L(t)/t_n^L)] | G(t) \} \end{aligned} \tag{36}$$

Among them, the right end of the inequality is the Lyapunov drift function’s upper bound. By minimizing the upper bound, the optimization model’s critical point can be obtained through data packet scheduling decisions  $d_n(t)$ . The system’s optimization model then transforms into the following expression:

$$\begin{aligned} \min_{b_n(t), d_n(t)} & Cons + \sum_{n=1}^N Q_n^H(t) (h_n^H(t) - B_n^H(t)) + \sum_{n=1}^N Q_n^L(t) (h_n^L(t) - B_n^L(t)) \\ & + V \sum_{n=1}^N d_n(t) \cdot [b_n(t) \cdot \tau_n^H(t)/t_n^H + (1 - b_n(t)) \cdot \tau_n^L(t)/t_n^L] \end{aligned} \tag{37}$$

$Cons$ ,  $h_n^H(t)$  and  $h_n^L(t)$  are constant in each time slot. Since the system results are not affected by them, they can be omitted. The Formula (37) is simplified as follows:

$$\min_{b_n(t), d_n(t)} \sum_{n=1}^N d_n(t) \cdot \left[ b_n(t) \cdot \left( V \frac{\tau_n^H(t)}{t_n^H} - Q_n^H(t) \right) + (1 - b_n(t)) \cdot \left( V \frac{\tau_n^L(t)}{t_n^L} - Q_n^L(t) \right) \right] \tag{38}$$

#### 4.1. Queue Selection Decision

Assuming a reasonable data packet scheduling decision vector  $D = \{d'_n(t)\}$ , we can describe the queue selection problem as follows:

$$\begin{aligned} \min_{b_n(t)} & \sum_{n=1}^N d'_n(t) \cdot \left[ b_n(t) \cdot \left( V \frac{\tau_n^H(t)}{t_n^H} - Q_n^H(t) \right) + (1 - b_n(t)) \cdot \left( V \frac{\tau_n^L(t)}{t_n^L} - Q_n^L(t) \right) \right] \\ \text{s.t.} & b_n(t) \in \{0, 1\}, 1 \leq n \leq N \end{aligned} \tag{39}$$

Since  $d'_n(t)$  is a constant, the optimal solution of the above equation can be expressed as

$$b_n^*(t) = \begin{cases} 1 & \left( V \frac{\tau_n^H(t)}{t_n^H} - Q_n^H(t) \right) \leq \left( V \frac{\tau_n^L(t)}{t_n^L} - Q_n^L(t) \right) \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

We can obtain the optimal decision vector of queue selection  $B = \{b_n^*(t)\}$ .

#### 4.2. Packet Scheduling Decision

Given the queue selection optimal decision vector  $B = \{b_n^*(t)\}$ , the packet scheduling problem is described as

$$\min_{d_n(t)} \sum_{n=1}^N d_n(t) \cdot \left[ b_n^*(t) \cdot \left( V \frac{\tau_n^H(t)}{t_n^H} - Q_n^H(t) \right) + (1 - b_n^*(t)) \cdot \left( V \frac{\tau_n^L(t)}{t_n^L} - Q_n^L(t) \right) \right]$$

$$s.t. \begin{cases} 0 \leq d_n(t) \leq S, 1 \leq n \leq N \\ \sum_{n=1}^N d_n(t) \leq S \cdot \kappa(t), \forall n, t \end{cases} \tag{41}$$

When we set  $\sigma_n(t) = b_n^*(t) \cdot \left( Q_n^H(t) - V \frac{\tau_n^H(t)}{t_n^H} \right) + (1 - b_n^*(t)) \cdot \left( Q_n^L(t) - V \frac{\tau_n^L(t)}{t_n^L} \right)$ , the system optimization model is described as

$$\max_{d_n(t)} \sum_{n=1}^N d_n(t) \cdot \sigma_n(t)$$

$$s.t. \begin{cases} 0 \leq d_n(t) \leq S, 1 \leq n \leq N \\ \sum_{n=1}^N d_n(t) \leq S \cdot \kappa(t), \forall n, t \end{cases} \tag{42}$$

The purpose of the optimization model was to maximize its value based on limited packet scheduling decisions (resource space). Therefore, we obtained optimal data scheduling by transforming the model into a knapsack problem [21].

In the knapsack problem, the system optimization model’s system value is expressed by  $\sigma_n(t)$ . The system first arranges the values of  $\sigma_n(t)$  in descending order, then inserts them into the “knapsack”. However, this process can be interrupted. The optimal solution to the system optimization problem depends on the system state at the time of interruption. Therefore, the key to the knapsack problem is to find the breakpoint. Typical interruption conditions in knapsack problems are as follows:

The remaining “weight” space is 0.

The value of the “item” inserted into the “backpack” is negative.

Optimal packet scheduling decisions can be determined by the break point of the knapsack problem. Define  $\Omega$  as the interruption index parameter of the system knapsack problem, and  $\Omega = \{\zeta_1, \zeta_2\}$ , of which  $\zeta_1, \zeta_2$  satisfies the following relation:

$$\begin{cases} \zeta_1 = \operatorname{argmin}_n \sum_{j=1}^n d_j(t) \geq \kappa(t)S \\ \zeta_2 = \operatorname{argmin}_n \sigma_n(t) < 0 \end{cases} \tag{43}$$

Then, the optimal solution for packet scheduling decisions  $d_n^*(t)$  is supplied as follows:

$$d_n^*(t) = \begin{cases} S & n < \Omega \\ \min\{\kappa(t)S - (\Omega - 1)S, d_\Omega(t)\} & n = \Omega \\ 0 & \text{otherwise} \end{cases} \tag{44}$$

### 5. Performance Evaluation

The simulation experiment in this section follows two parts of the simulation experiment on DDPS proposed in this paper through MATLAB simulation software. The first part involves drawing curve images of the average system delay and packet queue length, which vary with the system’s weight coefficient  $V$  and packet arrival rate. The second part compares this strategy model’s performance to two commonly used packet scheduling strategies, Equal Allocation Strategy (EAS) and Queue-weighted Strategy (QS), regarding system average delay and packet queueing under constant simulation parameters. Table 1 shows the simulation parameters.

In addition, Section 3 mentions the discontinuous Bernoulli distribution parameter  $\alpha = \beta = 0.5$ . The time,  $T$ , is set to 1500 s, the number of emergency data packet deadlines  $t_n^H$  is set to 1000 s, and the number of non-emergency data packet deadlines  $t_n^L$  is set to 1500 s.

**Table 1.** The simulation parameters.

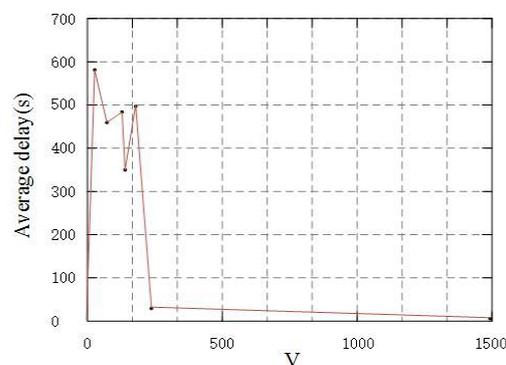
Parameter	Notation/Units of Measurement	Value
Highway mileage	L/m	10,000
RSU to Backbone RSU Spacing	$L'/m$	10,000
RSU Spacing	L/m	1000
Speed limit range	$[V_{min}, V_{max}]/(m/s)$	[16.67,33.33]
Speed expectation	$\bar{V}/(m/s)$	25
Speed standard deviation	$\sigma$	5.56
Vehicle arrival rate	$\lambda/(vehicle/s)$	5.5
Slot length	$\tau/s$	1
Maximum number of transmission packets	S	2

This paper considers that compared with the end-to-end delay, the end-to-end average delay fluctuates more stably, which can better measure the performance of the scheduling strategy. The lower the end-to-end average delay, makes the higher the transmission efficiency of data packets. The average end-to-end delay cannot exceed the data deadlines.

In the second part of the comparative experiment, the EAS indicates that the number of packet transmissions between RSUs and vehicles is equal. QS indicates that the system allocates the transmission weight between vehicles according to the packet queue length in RSUs: the longer the queue length of RSU packets, the more data RSUs send to vehicles.

Figures 3–5 show a DDPS simulation based on system performance. Figure 3 shows the curve trend of average delay in each time slot with the increase in weight coefficient  $V$ . The simulation results show that the average delay decreases with the increase in weight coefficient  $V$ . The results show that with the increase in weight coefficient  $V$ , the average delay’s impact on system performance increases gradually. Therefore, the system will adaptively reduce the average delay to ensure its overall performance. Figures 4 and 5 show that the high- and low-priority data queue lengths change with the increase in weight coefficient  $V$ , which reduces the impact of queue length on system performance. System decisions tend to adjust the factors that greatly impact system performance; therefore, the system reduces the average delay and increases the length of high- and low-priority queues.

The packet arrival rate will change accordingly when the system environment is different. Therefore, verifying whether the system’s packet scheduling strategy can adaptively optimize packet scheduling under different arrival rates ensures online optimization. The results of this experimental scheme are as follows: The experimental scheme provides the weight value  $w$ , set to 0.8, 1, 1.2, multiplied by the packet arrival state  $h_n(t)$  of the system. Set the weight to  $V = 2$ . Figures 6–8 show the experimental results. As shown in Figure 6, the increase in packet arrival rate increases the average queue length and system delays accordingly.



**Figure 3.** The weight-dependent trend of average delay.

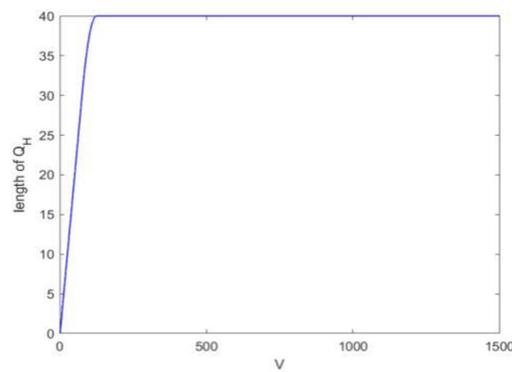


Figure 4. Curve of high-priority queue length changing with weight.

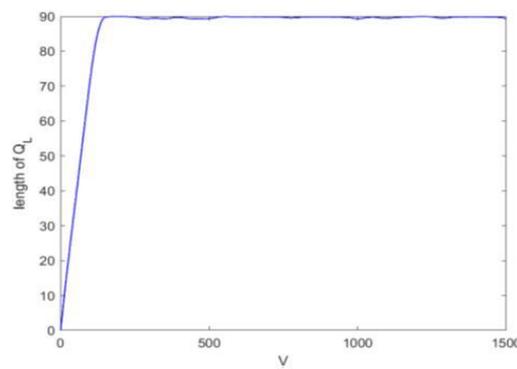


Figure 5. Curve of low-priority queue length changing with weight.

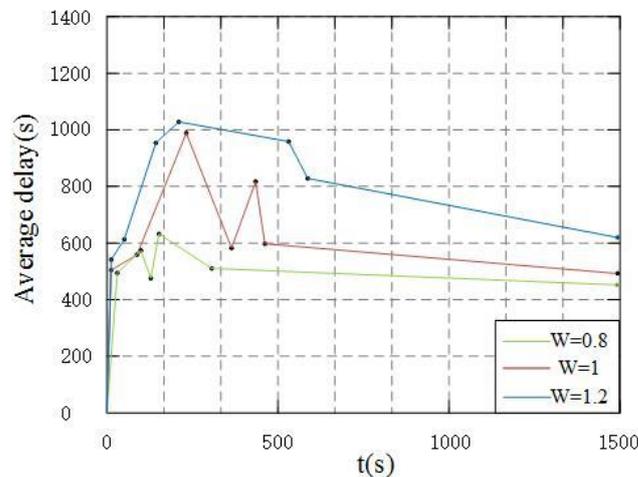
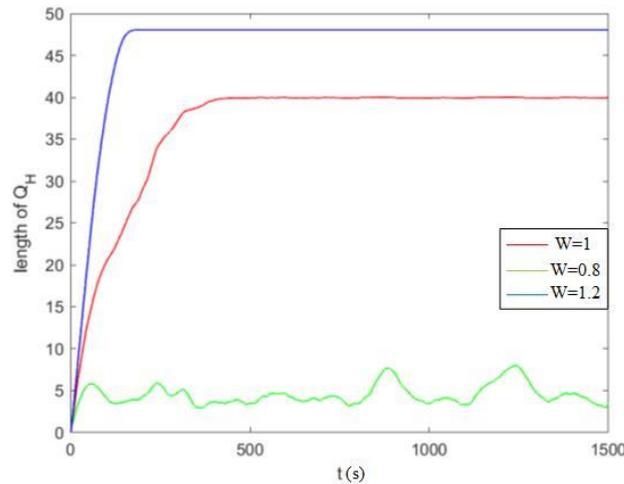


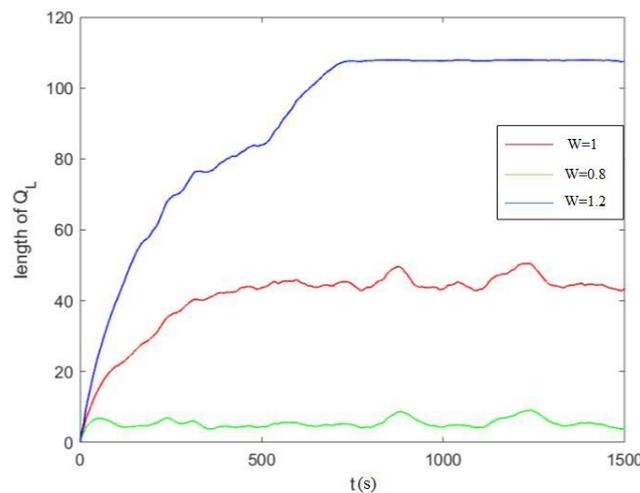
Figure 6. Comparison of average delay(s) under different packet arrival rates.

We consider that the size of a single data packet is 1 MB. In order to ensure the long-term stability of the data queue and avoid the high queuing delay of the data packet, which makes the total end-to-end delay exceed the deadline, the system sets high priority and low priority queue length should not exceed the threshold  $\omega\zeta$ , that is,  $\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{Q_n^H(t)\} \leq \omega$  and  $\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{Q_n^L(t)\} \leq \zeta$ . When the average packet arrival rate is  $\bar{h} = \sum_{u=0}^U u\theta_u = 1$ , the system sets  $\omega = 40$ ,  $\zeta = 90$ , that is, the queue length of high priority should not exceed 40 MB, and the queue length of low priority should not exceed 90 MB. As shown in Figures 7 and 8, the increase in packet arrival rate increases the system's queue length. Furthermore, the threshold  $\omega\zeta$  will increase with an increase in packet arrival rate. Figures 6–8

show that the system’s average delay and queue length can be manipulated under different packet arrival rates and finally converge. The results show that the DDPS strategy proposed in this paper can dynamically adjust the packet scheduling strategy according to the change in packet arrival rate. This method ensures the system’s optimal performance and stability.



**Figure 7.** Comparison of high-priority queue lengths under different packet arrival rates.



**Figure 8.** Comparison diagram of low-priority queue lengths under different packet arrival rates.

The above experiments analyze the performance of the DDPS and compare it to the EAS and QS to prove the advantages of the DDPS. We set the experiment weight to  $V = 2$ . The simulation results are shown in Figures 9–11. Figure 9 shows the average delay of the system, and Figures 10 and 11 compare the high- and low-priority queue lengths. The simulation results show that when the system works for a long time, the average DDPS delay is low. When it tends to be stable, the sum of high- and low-priority queue lengths is also low. DDPS improves system efficiency by dynamically adjusting the queue and the number of packets sent according to the speed status of passing vehicles and the length of high- and low-priority queues. On the other hand, the EAS’s insensitivity to vehicles’ high-speed mobility and packet queue length affects its performance. The QS only optimizes the system regarding queue length without considering the effect of vehicle speed on transmission delays. The DDPS proposed in this paper has obvious advantages over these two strategies.

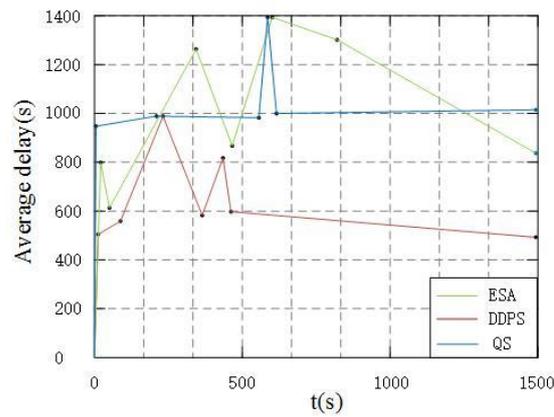


Figure 9. Comparison of average time delays under different algorithms.

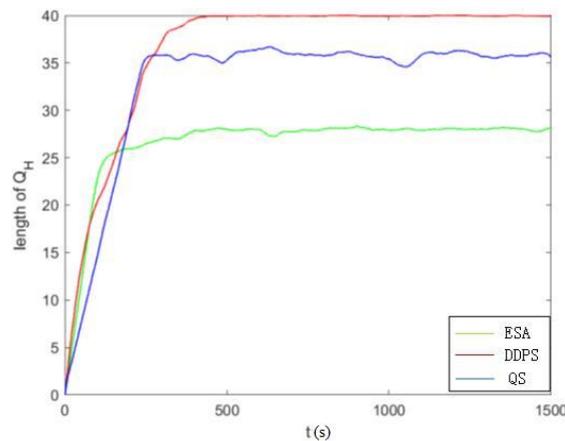


Figure 10. Comparison of high-priority queues under different algorithms.

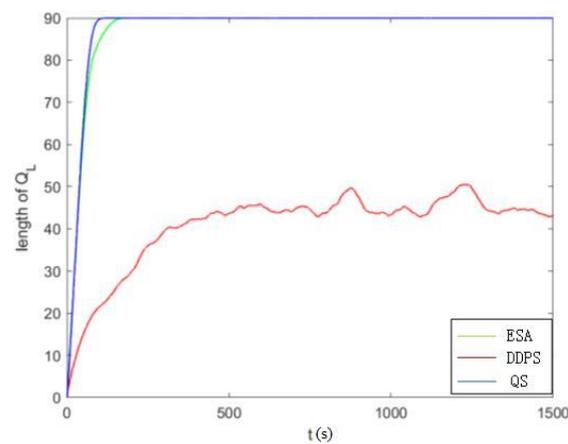


Figure 11. Comparison of low-priority queues under different algorithms.

### 6. Conclusions

In this paper, we proposed an optimization strategy to minimize end-to-end delays in multi-RSU-distributed packet scheduling based on vehicle carrier relays in CVIS. This strategy determines packet scheduling decisions according to the double buffer queue lengths and vehicle speed status of multiple RSUs on the highway. The advantage of Lyapunov theory is that it can maximize the network performance on the premise of ensuring the stability of the whole network, so Lyapunov optimization theory was used to transform the problem. Lyapunov drift parameters and weights were also introduced to the average

system delay and length of high- and low-priority packet queues, which were treated as knapsack problems. This method can dynamically adjust the packet scheduling strategy through the real-time changes of multiple RSUs' high- and low-priority queue lengths and vehicle speed status on the road, ensuring that the packet is transmitted to the destination before the deadline and minimizing end-to-end delays.

The simulation results show that the average end-to-end delay of the DDPS strategy proposed in this paper is 39% lower than the ESA strategy and 50% lower than the QS strategy when the queue length of the proposed ESA policy is higher and lower than that of the QS strategy and is significantly lower than that of the QS strategy. With the continuous change of parameters, the end-to-end average delay of the DDPS strategy does not change much, so it can be known that the DDPS strategy has good robustness.

In the assumption of this paper, a single lane of vehicle arrival in a highway section obeys a Poisson distribution with an arrival rate of  $\lambda$ , considering that the vehicle arrival rate is independently and identically distributed in multiple lanes, that is, the number of vehicle arrivals in multiple lanes can be expressed as multiple superpositions of a single lane. In the simulation, the DDPS proposed in this paper has little change in system performance when the arrival rate is high, so the DDPS strategy is also suitable for multi-lane scenarios.

In the hypothetical scenario of this paper, the transmission of data packets does not take into account the influence of the weather environment, and in the real-world scenario, vehicles may be affected by rainy weather and road conditions in the process of data store–carry–forward. Therefore, we will further study this limitation in future work.

**Author Contributions:** Conceptualization, L.D.; methodology, B.S.; validation, L.D., Y.J. and B.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Key Research and Development Program of China (2021YFB2601401), the Natural Science Foundation of Shaanxi Province, China (2019JQ-264) and the Fundamental Research Funds for the Central Universities, CHD(300102323201).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

## References

1. Li, C.; Zhang, Y.; Luan, T.H.; Fu, Y. Building Transmission Backbone for Highway Vehicular Networks: Framework and Analysis. *IEEE Trans. Veh. Technol.* **2018**, *67*, 8709–8722. [[CrossRef](#)]
2. Li, J.; Cao, X.; Guo, D.; Xie, J.; Chen, H. Task Scheduling With UAV-Assisted Vehicular Cloud for Road Detection in Highway Scenario. *IEEE Internet Things J.* **2020**, *7*, 7702–7713. [[CrossRef](#)]
3. Atallah, R.F.; Khabbaz, M.J.; Assi, C.M. Modeling and Performance Analysis of Medium Access Control Schemes for Drive-Thru Internet Access Provisioning Systems. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3238–3248. [[CrossRef](#)]
4. He, J.; Cai, L.; Pan, J.; Cheng, P. Delay Analysis and Routing for Two-Dimensional VANETs Using Carry-and-Forward Mechanism. *IEEE Trans. Mob. Comput.* **2017**, *16*, 1830–1841. [[CrossRef](#)]
5. Huang, L.; Jiang, H.; Zhang, Z.; Yan, Z.; Guo, H. Efficient Data Traffic Forwarding for Infrastructure-to-Infrastructure Communications in VANETs. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 839–853. [[CrossRef](#)]
6. Si, P.; He, Y.; Yao, H.; Yang, R.; Zhang, Y. DaVe: Offloading Delay-Tolerant Data Traffic to Connected Vehicle Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 3941–3953. [[CrossRef](#)]
7. Liu, C.-F.; Bennis, M.; Debbah, M.; Poor, H.V. Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing. *IEEE Trans. Commun.* **2019**, *67*, 4132–4150. [[CrossRef](#)]
8. Bok, K.; Lim, J.; Hong, S.; Yoo, J. A Multiple RSU Scheduling for V2I-based Data Services. In Proceedings of the International Conference on Big Data and Smart Computing (BigComp), Hong Kong, China, 18–20 January 2016; pp. 163–168.
9. Guo, T.; Li, C.; Zhang, Y.; Miao, Z.; Xiong, L. Integrity-oriented service scheduling for vehicle-to-roadside data access. In Proceedings of the 2016 19th International Symposium on Wireless Personal Multimedia Communications (WPMC), Shenzhen, China, 14–16 November 2016; pp. 377–382.

10. Tan, P.H.; Zheng, J.; Li, Z.; Sun, S. Joint Rate Control and Scheduling for Video Streaming over OFDMA Networks. In Proceedings of the IEEE International Conference on Communication Systems (IEEE ICCS), Macau, China, 19–21 November 2014; pp. 579–583.
11. Bethanabhotla, D.; Caire, G.; Neely, M. Adaptive Video Streaming for Wireless Networks with Multiple Users and Helpers. *IEEE Trans. Commun.* **2014**, *63*, 268–285. [[CrossRef](#)]
12. Cai, Y.; Llorca, J.; Tulino, A.M.; Molisch, A.F. Optimal Cloud Network Control with Strict Latency Constraints. In Proceedings of the ICC 2021–IEEE International Conference on Communications, Montreal, QC, Canada, 14–23 June 2021; pp. 1–6. [[CrossRef](#)]
13. Hu, B.; Shi, Y.; Cao, Z. Adaptive Energy-Minimized Scheduling of Real-Time Applications in Vehicular Edge Computing. *IEEE Trans. Ind. Inform.* **2022**, 1–11. [[CrossRef](#)]
14. Mukherjee, M.; Guo, M.; Lloret, J.; Iqbal, R.; Zhang, Q. Deadline-Aware Fair Scheduling for Offloaded Tasks in Fog Computing With Inter-Fog Dependency. *IEEE Commun. Lett.* **2020**, *24*, 307–311. [[CrossRef](#)]
15. Sethi, V.; Pal, S.; Vyas, A. Online Energy-efficient Scheduling Algorithm for Renewable Energy-powered Roadside units in VANETs. In Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Delhi, India, 10–13 December 2020; pp. 506–514. [[CrossRef](#)]
16. Wu, H.; Chen, J.; Nguyen, T.N.; Tang, H. Lyapunov-Guided Delay-Aware Energy Efficient Offloading in IIoT-MEC Systems. *IEEE Trans. Ind. Inform.* **2023**, *19*, 2117–2128. [[CrossRef](#)]
17. Xiang, X.; Fan, B.; Dai, M.; Wu, Y.; Xu, C.-Z. V2X Communication Aided Emergency Message Dissemination in Intelligent Transportation Systems. In Proceedings of the 2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR), Taicang, China, 6–8 June 2022; pp. 35–40. [[CrossRef](#)]
18. Dai, L.; Zhang, Y.; Qian, C.; Meng, Y.; Huang, H. Optimal Packet Scheduling Strategy for Roadside Units' Bursty Traffic Based on Relaying Vehicles. *Acta Autom. Sin.* **2021**, *47*, 1098–1110.
19. Wen, C.; Zheng, J. An RSU On/Off Scheduling Mechanism for Energy Efficiency in Sparse Vehicular Networks. In Proceedings of the 7th IEEE International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 15–17 October 2015.
20. Dai, L.; Chen, T.; Zhai, Y.; Wang, G. Energy-Efficient Distributed Packet Scheduling Optimization Strategy in Cooperative Vehicle Infrastructure Systems. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6661623. [[CrossRef](#)]
21. Chen, Y.; Zhang, N.; Zhang, Y.; Chen, X.; Wu, W.; Shen, X. Energy Efficient Dynamic Offloading in Mobile Edge Computing for Internet of Things. *IEEE Trans. Cloud Comput.* **2021**, *9*, 1050–1060. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.