Multimedia Delivery over Heterogeneous Wireless Networks

by

Min Xing
B.Eng., Soochow University, 2007
M.Sc., Tongji University, 2010

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Multimedia Delivery over Heterogeneous Wireless Networks

by

Min Xing
B.Eng., Soochow University, 2007
M.Sc., Tongji University, 2010

Supervisory Committee

Dr. Lin Cai, Supervisor
(Department of Electrical and Computer Engineering)

Dr. T. Aaron Gulliver, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Alex Thomo, Outside Member
(Department of Computer Science)

Supervisory Committee

Dr. Lin Cai, Supervisor
(Department of Electrical and Computer Engineering)

Dr. T. Aaron Gulliver, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Alex Thomo, Outside Member
(Department of Computer Science)

ABSTRACT

There is an increasing demand for multimedia services in heterogeneous wireless networks. Considering the highly dynamic wireless channels and the relatively large size of the multimedia data, how to support efficient and reliable multimedia delivery is a pressing issue. In this dissertation, we investigate the multimedia delivery algorithms in heterogeneous wireless networks from three different aspects.

First, we study the single-flow rate adaptation of video streaming algorithm over multiple wireless interfaces. In order to maintain high video streaming quality while reducing the wireless service cost, the optimal video streaming process with multiple links is formulated as a Markov Decision Process (MDP). The reward function is designed to consider the quality of service (QoS) requirements for video traffic, such as the startup latency, playback fluency, average playback quality, playback smoothness and wireless service cost. To solve the MDP in real time, we propose an adaptive, best-action search algorithm to obtain a sub-optimal solution. To evaluate the performance of the proposed adaptation algorithm, we implemented a testbed using the Android mobile phone and the Scalable Video Coding (SVC) codec and conducted experiments with real video flow.

Then, with the multiple multimedia flows competing for limited wireless resources, we propose a utility-based scheduling algorithm for multimedia transmission in Drive-

thru Internet. A utility model is devised to map the throughput to user's satisfaction level in terms of multimedia data quality, such as Peak Signal-to-Noise Ratio (PSNR) of video. The objective of the scheduling problem is to maximize the total utility. Then the optimization problem is formulated as a finite-state decision problem with the assumption that future arrival information is known, and it is solved by a searching algorithm as the benchmark. To obtain a real-time solution, a practical heuristic algorithm based on the concept of utility potential is devised. We further implemented the solution and conducted extensive simulations using NS-3.

Finally, the multimedia dissemination problem in large-scale VANETs is investigated. We first utilize a hybrid-network framework to address the mobility and scalability issues in large-scale VANETs content distribution. Then, we formulate a utility-based maximization problem to find the best delivery strategy and select an optimal path for the multimedia data dissemination, where the utility function has taken the delivery delay, the Quality of Services (QoS) and the storage cost into consideration. We obtain the closed-form of the utility function, and then obtain the optimal solution of the problem with the convex optimization theory. Finally, we conducted extensive trace-driven simulations to evaluate the performance of the proposed algorithm with real traces collected by taxis in Shanghai.

In summary, the research outcomes of the dissertation can contribute to three different aspects of multimedia delivery in heterogeneous wireless networks. First, we have proposed a real-time rate adaptation algorithm for video streaming with multiple wireless interfaces, to maintain the high quality while reducing the wireless services cost. Second, we have presented an optimal scheduling algorithm which can maximize the total satisfaction for multimedia transmission in Drive-thru Internet. Third, we have derived the theoretical analysis of the utility functions including delivery delay, QoS and the storage cost, and have obtained an optimal solution for multimedia data dissemination in large-scale VANETs to achieve the highest utility.

Contents

List of Tables

List of Figures

List of Abbreviations

AP	Access Point
APQ	Average Playback Quality
CS	Compressed Sensing
DASH	Dynamic Adaptive Streaming over HTTP
DTN	Delay Tolerant Networks
FCFS	First Come First Service
GoP	Group of Pictures
MAC	Media Access Control
MDP	Markov Decision Process
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
RSU	Road Side Units
RTT	Round-Trip Time
SL	Startup latency
SNR	Signal-to-Noise Ratio
SVC	Scalable Video Coding
TDMA	Time Division Multiplex Access
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANETs	Vehicular Ad-Hoc Networks

ACKNOWLEDGEMENTS

Foremost, I would like to express my deepest gratitude to my supervisor Dr. Lin Cai for her guidance, inspiration and support throughout my Ph.D study and research at the University of Victoria. She is always available for my questions and lead me to the right source, theory and perspective.

I would also like to extend my sincerest gratitude to Dr. Jianping Pan for his valuable comments and suggestions, Dr. T. Aaron Gulliver and Dr. Alex Thomo for serving as my committee members, and Dr. Mohamed Hefeeda for being my external examiner and providing insightful feedback.

My warm thanks to all my fellow lab mates and friends of University of Victoria, Dr. Zhe Yang, Dr. Siyuan Xiang, Dr. Yuanqian Luo, Dr. Jianping He, Dr. Xuan Wang, Dr. Lei Zheng, Kan Zhou, Yi Chen, Zhe Wei, Haoyuan Zhang, Lei Zhang, Yimian Du, Dong Zhang, Tianyang Li and all others I have not mentioned here. The time we worked and had fun together will never be forgot.

Last but certainly not least, my deepest thanks to my parents for their endless support and love.

Min Xing, Victoria, BC, Canada

DEDICATION

To my parents, and all of my friends.

Chapter 1

Introduction

In this dissertation work, we study various aspects of how to deliver multimedia traffic in heterogeneous wireless networks. First, a dynamic rate adaptive video streaming algorithm for single user with multiple wireless access networks has been proposed and the feasibility and the effectiveness of the proposed algorithm has been demonstrated. Second, with the limited wireless communication resources and sojourn time, a multimedia transmission scheduling algorithm for Drive-thru Internet has been proposed to improve the total users' satisfaction level. Third, the multimedia data dissemination in large-scale Vehicular Ad-Hoc Networks (VANETs) has been investigated, the expected utility for any given path has been analyzed, and the maximum utility dissemination algorithm has been proposed.

1.1 Background

With the rapid development of the wireless communication technologies, the improved bandwidth and reliability can better support multimedia applications for wireless networks. It has been forecast that mobile video data traffic will account for over 75% of the total mobile data traffic by 2018 [22]. Considering the relatively large size of the multimedia traffic flow, transmission efficiency and reliability are key requirements to support high quality multimedia applications. Since the wireless channels are highly dynamic, it is very challenging to provide high quality multimedia services, *e.g.*, video streaming, for mobile users consistently.

For video streaming applications, dynamic adaptive streaming over HTTP (DASH) [63] is a promising approach, as it is simple yet effective. Typically, the design of the

DASH algorithm must take several Quality of Service (QoS) metrics into consideration, to ensure the client's watching experience. For instance, the appropriate quality version of the video should be requested to avoid frequent video quality fluctuations and video playback interruptions. The latest mobile devices, such as smart phones and tablets, are equipped with multiple wireless network interfaces, *e.g.,* cellular data communications, WiFi and Bluetooth. With multiple wireless links together, the aggregated bandwidth and the reliability of the transmission can be substantially improved. Thus, using multiple wireless links simultaneously in DASH becomes a desirable approach. Considering the bandwidth and cost differences of different wireless links, how to optimize the rate adaptation process for video streaming over multiple wireless links is an open issue.

For people on the road, there is also an increasing demand of multimedia services. With multimedia services, not only the road safety can be enhanced, but also the traveling experience for passengers can be improved. Relying on the vehicle-to-infrastructure (V2I) communications, Road Side Units (RSU) are deployed along the roads as access points (APs) to provide high speed yet low cost Internet services. When vehicles travel through the coverage area of the RSUs, they can access the Internet by the RSUs. Due to the high cost, it is impossible to deploy enough RSUs to cover the entire road. For such Drive-thru Internet, the total throughput of each vehicle is restricted, due to the limited sojourn time. When there are multiple vehicles competing for the limited wireless resources, the throughput will be further reduced. Therefore, how to design a scheduling algorithm for multimedia transmissions of multiple vehicles to achieve the highest total satisfactions needs investigation.

Then we expand our vision from the Drive-thru Internet to the large-scale Vehicular Ad-Hoc Networks (VANETs), where vehicle-to-vehicle (V2V) message exchanges are possible. There are lots of potential multimedia applications. For instance, the large-scale VANETs can provide low cost media-rich entertainments or location-aware applications, compared to the high-cost of cellular networks. The multimedia data generated at the source node can be carried and forwarded to the destination by the passing vehicles through V2V communications. If the vehicle carrying the multimedia data cannot find any appropriate vehicle to forward, the multimedia data may be dropped and the delivery is terminated. Hence, an important requirement for such applications is to find a reliable routing path to forward. Due to the high mobility, large scale, and the limited contact time between vehicles, it is quite challenging to support the multimedia data dissemination in VANETs.

1.2 Research Objectives and Contributions

The above open issues motivated the dissertation work, which focuses on multimedia delivery in heterogeneous wireless networks, and the detailed objectives and contributions are discussed as follows.

1.2.1 A Real-time Adaptive Algorithm for Video Streaming over Multiple Wireless Access Networks

The latest smart devices are equipped with multiple wireless network interfaces, *e.g.*, smart phones are integrated with cellular networks, WiFi and Bluetooth. Using multiple wireless interfaces to support video streaming applications is a promising approach, since the aggregated bandwidth can support higher video quality, and the improved network reliability can reduce the number of video playback interruptions. With multiple wireless links, although the aggregated bandwidth is increased, the highly dynamic wireless channels may not be able to support high quality video consistently, and it is more difficult to estimate the future bandwidth accurately. In addition, considering the cost differences of the different wireless links, *e.g.*, cellular networks charge significantly higher than WiFi, we cannot allocate too much transmission load to cellular link due to the cost. Therefore, how to optimize this rate adaptation process for video streaming over multiple wireless links, considering the video QoS requirements, the wireless channel profiles, and the wireless service costs of multiple links is an open issue.

Generally speaking, the video streaming research can be classified into two categories. In the first category, the authors considered the multicast scenario and tried to optimize the performance on the server or base station side [34, 35, 44, 51, 60]. For the other category, the authors focused on the pull-based approaches on the client side [40, 50, 52, 66, 75]. DASH [63] is a promising technique to be extended to support multiple wireless links, *i.e.*, we can let each link request different section of the video data through HTTP range request. Recently, there are several works devoted to this topic [25–27, 38, 83]. Kaspar *et al.* [38] considered to request multiple video segments simultaneously by transmitting each segment over one link. Due to the link data rate difference, the aggregated bandwidth may not be fully utilized, and the "last-segment" problem may be introduced. To utilize bandwidth more efficiently, [25–27] tried to divide each segment into small sub-segments, and transmit them over dif-

ferent links. Although they tried to request the highest possible video quality, the playback smoothness was ignored. In addition, they did not take the wireless service cost into consideration.

In order to maintain high video streaming quality while reducing the wireless service cost, in this dissertation, we have investigated how to design an optimal adaptive video streaming algorithm with multiple wireless interfaces. Multiple QoS requirements for video traffic, such as the startup latency, playback fluency, average playback quality, playback smoothness and wireless service cost have been take into consideration to design a real time adaptive video streaming algorithm. We have also demonstrated that the proposed algorithm can outperform the state-of-the-art algorithm.

1.2.2 Maximum-Utility Scheduling for Multimedia Transmission in Drive-Thru Internet

Different from the previous section, when there are multiple video flows competing for the limited wireless resource, the key issue is how to allocate the wireless resources and schedule multiple transmissions to efficiently and fairly share the resources. The scheduling of multimedia transmission in Drive-thru Internet is studied in this work. For the Drive-Thru Internet, since the sojourn time is limited for each vehicle, the total throughput of each vehicle is restricted. The throughput will be further reduced with the competition of multiple vehicles. As a result, multimedia data may not be transmitted completely due to the relatively large size. In addition, since the RSU does not know the future vehicle arrival information, it is difficult to make the optimal scheduling decisions.

There is little work done for the multimedia transmission scheduling problem in Drive-thru Internet. Most of the exiting works can be classified into two categories. In the first category, the authors focused on modeling the throughput performance in the MAC layer only [48, 65, 91]. In the other category, they proposed several MAC layer scheduling algorithms [9, 20, 21] to reduce the transmission backlog or the wireless service cost. Since the QoS metrics of the multimedia data are not considered, they are not suitable for multimedia transmission. Zhang *et al.* proposed an application-layer service scheduling algorithm with the consideration of both service deadline and data size [89]. Assuming constant wireless data rate may not be practical.

To overcome the above problems, we have proposed a novel scheduling algorithm

for multimedia transmission in Drive-thru Internet which aims to maximize the total satisfaction of multimedia quality. The QoS metrics of the multimedia data have been incorporated into the design of the scheduling algorithm. To make the problem more practical, we have considered not only the current vehicles in the RSU's coverage, but also estimated the future arrival to achieve good performance.

1.2.3 Maximum-Utility Multimedia Dissemination in Large-scale VANETs

In large-scale VANETs, a multimedia message can be carried and forwarded using V2V communications to reach the destination via multiple vehicles. Most of the multimedia applications depend on the efficient and reliable multimedia data dissemination. However, since the contact time between vehicles is usually limited, the multimedia data cannot be fully transmitted between two contacting vehicles by V2V communication, which brings a new challenge for data dissemination. Therefore, how to design the dissemination scheme to support efficient and reliable multimedia data dissemination in the large-scale VANETs is an interesting and challenging problem.

There are lots of work considering epidemic routing [11, 62], density based routing [70] or prediction based routing [84] to solve data dissemination problems in VANETs. Due to the flooding nature of these routing schemes, they may not be suitable for multimedia data dissemination in large-scale VANETs. First, considering the huge number of vehicles in the urban large-scale VANETs, it is impractical to let each vehicle to maintain a list of pair-wise contact probability and pattern, and the flooding algorithms cannot scale well. Second, because of the relatively large size of the multimedia data considering the limited bandwidth, it is hard to replicate multiple copies and forward them to the unexpected connections, i.e., dynamic and limited contact pattern and time. Third, within the short contact time, it is difficult to make a good data forwarding and routing strategy.

To address the above issues, we are motivated to devise a low cost yet efficient and reliable multimedia data dissemination algorithm in large-scale VANETs. We have adopted a hybrid-network framework, where the mobility and scalability issues are addressed. In addition to the delivery delay, we have taken the QoS of the multimedia data and the storage cost of the drop box into consideration. The performance of the proposed algorithm has been evaluated with the real trace collected by taxicabs in Shanghai.

1.3 Dissertation Outline

This work focuses on multimedia delivery in heterogeneous wireless networks. The rest of the dissertation is organized as follows.

In Chapter 2, we discuss the dynamic rate adaptation for the single-user video streaming algorithm with multiple wireless access networks. The video streaming process is formulated as a Markov Decision Process (MDP), and reward functions are defined with the consideration of QoS requirements. An on-line adaptive search algorithm is proposed to solve the problem.

In Chapter 3, we present the multimedia transmission scheduling in Drive-thru Internet for VANETs. The scheduling process is formulated as finite-state decision problem to maximize the total users' satisfaction level. By exploiting the utility potential in the future, we propose a real-time heuristic algorithm to solve the scheduling problem.

In Chapter 4, we investigate the multimedia data dissemination problem in large-scale VANETs. Under an existing hybrid framework, the dissemination process is formulated as an optimization problem with the objective to maximize the final utility. The theoretical analysis of the expected delay and utility are derived for any given path. The optimal solution is obtained with the convex optimization theory.

Chapter 5 concludes the dissertation and suggests the future research directions.

1.4 Bibliographic Notes

Most of the work presented in this dissertation have appeared in research papers. The works in Chapter 2 have been published in [81, 82]. The works in Chapter 3 have been accepted in [79]. The works in Chapter 4 have been submitted as [80].

Chapter 2

A Real-time Adaptive Algorithm for Video Streaming over Multiple Wireless Access Networks

In this chapter, we investigate the efficient and cost-effective rate adaptation algorithm for video streaming over multiple wireless interfaces. The video streaming process with multiple links is formulated as a Markov Decision Process (MDP). The reward function is designed to consider the quality of service (QoS) requirements for video traffic, such as the startup latency, playback fluency, average playback quality, playback smoothness and wireless service cost. To solve the MDP in real time, we propose an adaptive, best-action search algorithm to obtain a sub-optimal solution. To evaluate the performance of the proposed adaptation algorithm, we implemented a testbed using the Android mobile phone and the Scalable Video Coding (SVC) codec. Experiment results demonstrate the feasibility and effectiveness of the proposed adaptation algorithm for mobile video streaming applications, which outperforms the existing state-of-the-art adaptation algorithms.

2.1 Introduction

In order to improve the reliability and throughput of the wireless Internet, it is a promising trend to use multiple wireless network interfaces with different wireless communication techniques for mobile devices. For example, smart phones and tablets are usually equipped with cellular, WiFi and Bluetooth interfaces. Utilizing multiple

links simultaneously can improve video streaming in several aspects: the aggregated higher bandwidth can support video of higher bit rate; when one wireless link suffers poor link quality or congestion, the others can compensate for it.

High resilience to bandwidth variation and easy deployment are both important requirements for video streaming applications. Currently, progressive download, one of the most popular and widely deployed streaming techniques, buffers a large amount of video data to absorb the variations of bandwidth. Meanwhile, as video data are transmitted over HTTP protocols, the video streaming service can be deployed on any web server. However, the video quality version can only be manually selected by the user and such decision can be error-prone. Since the smart phones only have limited storage space, it is impractical to maintain a very large buffer size. In addition, the buffered unwatched video may be wasted if the user turns off the video player or switches to other videos.

To improve the performance of progressive download, dynamic adaptive streaming over HTTP (DASH) [63] has been proposed. In a DASH system, multiple copies of pre-compressed videos with different resolution and quality are stored in segments. The rate adaptation decision is made at the client side. For each segment, the client can request the appropriate quality version based on its screen resolution, current available bandwidth, and buffer occupancy status. This pull-based DASH scheme can be extended to support multiple links, *i.e.*, we can let the client request different parts of one segment over different links. How to optimize this rate adaptation process for video streaming over multiple wireless links, considering the video quality of service (QoS) requirements, the wireless channel profiles, and the wireless service costs of multiple links is an open issue.

In this chapter, we formulate the multi-link video streaming process as a reinforcement learning task. For each streaming step, we define a state to describe the current situation, including the index of the requested segment, the current available bandwidth and other system parameters. A finite-state Markov Decision Process (MDP) can be modeled for this reinforcement learning task. The reward function is carefully designed to consider the video QoS requirements, such as the interruption rate, average playback quality, and playback smoothness, as well as the service costs. To make a trade-off between different QoS metrics and the cost, we can adjust the parameters of the reward function. To solve the MDP in real time, we proposed an adaptive best-action search algorithm to obtain a sub-optimal solution. A realistic testbed is implemented to better evaluate the performance of our solution.

The main contributions of this chapter are threefold. First, we formulate the video streaming process over multiple links as an MDP problem. To achieve smooth and high quality video streaming, we define several actions and reward functions for each state. Second, we propose a depth-first real-time search algorithm. The proposed adaptation algorithm will take several future steps into consideration to avoid playback interruption and achieve better smoothness and quality. Last, we implement a realistic testbed using an Android phone and Scalable Video Coding (SVC) encoded videos to evaluate the performance. The experiment results show that the proposed adaptation algorithm is feasible for video streaming over multiple wireless access networks, and it outperforms the existing state-of-the-art algorithms.

2.2 Background and Related Work

DASH has been a hot topic in recent years. There are many commercial products which have implemented DASH in different ways, such as Apple HTTP Live Streaming and Microsoft Smooth Streaming. Since the clients may have different available bandwidth and display size, each video will be encoded several times with different quality, bit rate and resolution. All the encoded videos will be chopped into small segments and stored on the server, which can be a typical web server. These small segments will be downloaded to the browsers' cache and played by the client (browser or browser plug-in). The video rate adaptation is performed at the client side, which is also called the pull-based approach. The client will determine the quality version of the requested video segment according to its current available bandwidth, resolution and the number of buffered unwatched segments. After the current segment is completely downloaded, the rate adaptation algorithm will be invoked again for the next segment.

There is extensive work covering this topic [40, 50, 52, 66, 75]. The authors in [66] proposed to estimate the bandwidth by a statistical method, and they took both the quality contribution and decoding time of each segment into consideration. K.P. Mok *et al.* presented a QoE aware DASH system [50]. Their algorithm estimates the available bandwidth by probing with the video data. In order to keep the quality level as smooth as possible, their algorithm will switch the video quality version gradually and will try to maintain the buffer level being stable. S. Akhshabi designed an evaluation method in [8] to test the performance of several existing commercial DASH products, such as Smooth Streaming, Netflix, and OSMF. In [40], T. Kupka

proposed to evaluate the performance of live DASH under on/off traffic and tested four different methods to improve the performance. In [52], how to reduce unnecessary video quality variations using a probing method to identify the effective available bandwidth was given. In [75, 76], the authors designed the optimal rate adaptation algorithm for streaming scalable video coding (SVC) over HTTP using MDP. With SVC, each video frame is encoded into a base layer and several enhancement layers. Higher video quality can be achieved when more layers are received. These works only considered the single-link case, and in this work, we consider the more challenging case with multiple access links.

Recently, a few approaches have appeared to extend the DASH technique to support multiple links. In [83], the authors summarized three typical schemes of utilizing multiple links. They compared the performance of these schemes through extensive simulations. Kaspar *et al.* proposed an approach to implementing DASH over multiple links [38]. In their algorithm, each segment will be transmitted over one link. Thus multiple segments can be transmitted at the same time. To reduce the overhead, they used HTTP pipelining to improve the performance. This approach may lead to the "last-segment problem" that the later segments may finish transmission before earlier segments, due to the link transmission speed difference. To overcome this disadvantage, in [25–27], Evensen *et al.* suggested to divide each segment into small sub-segments, and these sub-segments can be downloaded through different links. Their algorithm estimates the available bandwidth according to the throughput of the previous segment, and selects the video quality version most close to the estimated bandwidth. In the evaluation section, we will compare the performance of our proposed solution with the above state-of-the-art one [27].

2.3 System Model and Streaming Process Formulation

2.3.1 System Model

We consider how to utilize multiple wireless access networks together for video streaming, *e.g.*, using a combination of cellular, WiFi, and/or Bluetooth simultaneously. Here, as an example, Bluetooth and WiFi access networks are considered as we do not have end-to-end control over cellular links, and our work can be extended when

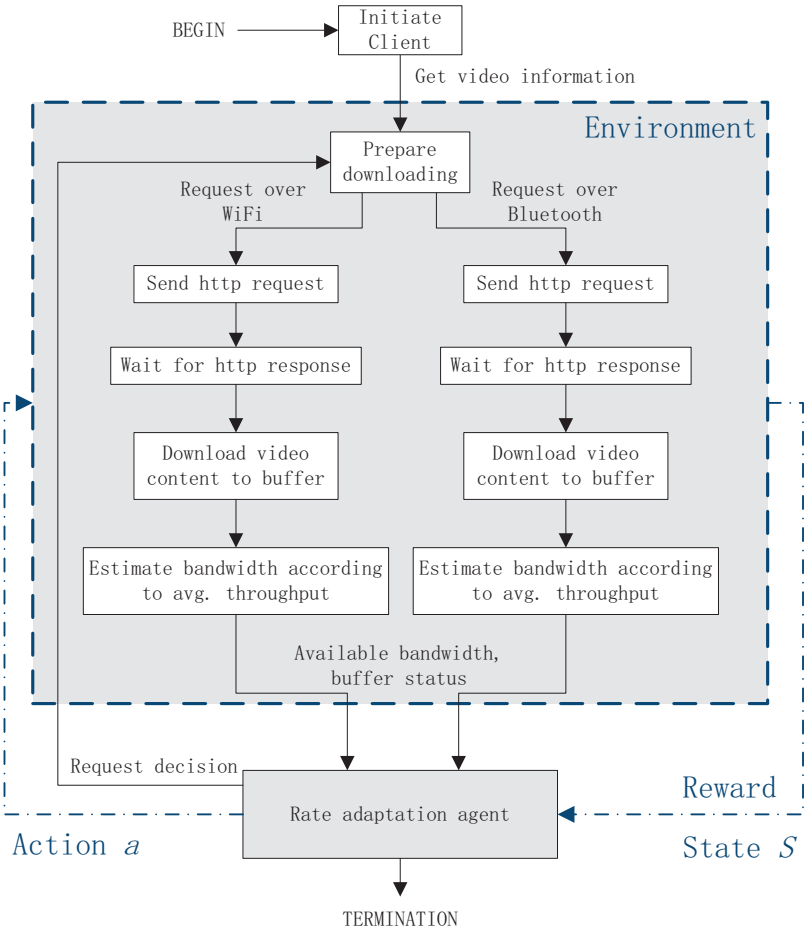


Figure 2.1: System Model.

other types of wireless access networks or more than two wireless access networks are used[1]. Since a wireless channel may suffer from time-varying fading, shadowing, interference and congestion, the available bandwidth of a wireless link may vary all the time. In addition, different smart phones or tablets may have different screen size and resolution. Taking these two aspects into consideration, the server should store several copies of video with different quality. The videos are encoded with SVC into a base layer and several enhancement layers, and chopped into segments and each segment can be played with a fixed duration.

We design a pull-based algorithm for video streaming, as shown in Fig. 2.1. After initialization, the client will request the video information which includes video reso-

[1]A promising multi-link scenario is to use both WiFi and 3G/LTE celluar links for video streaming. In our testbed, as the Android OS restricted the utilization of WiFi and 3G to access Internet simultaneously, we use WiFi and Bluetooth to test our multi-link streaming solution.

lutions, bit-rates and qualities from the server through both the WiFi and Bluetooth links. The rate adaptation agent will request a video segment of appropriate quality version based on the current queue length and estimated available bandwidth. Once the request decision is made, HTTP requests over both WiFi and Bluetooth will be issued to download the video segment. This process will continue until the completion of downloading the last segment or the termination of the video streaming by the user.

2.3.2 Streaming Process Formulation

The video streaming process can also be considered as the interaction between two modules. As shown in Fig. 2.1, the downloading and estimation steps in the top grey rectangle can be viewed as an integrated environment module, and the rate adaptation agent can be viewed as an agent module. The video streaming process can be formulated as a reinforcement learning task [64]. The environment sends a state signal for each video segment to the agent, and the agent will determine the best action correspondingly. For each action, the environment replies a reward to the agent. Considering the Markov property of the system states, a Markov Decision Process (MDP) can be formulated for the streaming process, and the state transition model of the Markov process needs to be devised.

We define step n as to download segment n, so the total number of steps equals the number of segments. For each step n, we define the state as $s_n = \{q_n, \Delta q_n, v_n, \Delta v_n, t_n, \Delta t_n, bw_n, bg_n, d_n\}$, with the parameters defined as follows. q_n represents the number of unplayed queued segments, which is also called the buffer level (or queue length), with the range between 0 and q_L (which is the maximum queue length). v_n is the SVC video layer index of the n-th segment. In our work, there are L SVC video layers in total, and a larger number represents a higher-quality layer. Δq_n and Δv_n are the variations of q_n and v_n respectively, *i.e.*, $\Delta q_n = q_n - q_{n-1}$, and $\Delta v_n = v_n - v_{n-1}$. The total traffic of Bluetooth used to download the previous segment is recorded by t_n, and $\Delta t_n = t_n - t_{n-1}$. The current bandwidth states of the WiFi link and the Bluetooth link are described by bw_n and bt_n, respectively. d_n indicates the current requested segment index. As the total number of video segments is N_T, d_n is in the range of $[0, N_T]$.

When the rate adaptation agent receives input of state s_n from the environment, it will make the decision of which action should the environment take. We define four

types of actions, A_b, A_u, A_w, and A_s. The first two actions are for downloading the next segment at the quality determined by v, and Δv determines whether to upgrade, downgrade or maintain the same quality. Once the quality v is determined, the base layer and all enhancement layers belong to quality v will be combined together to be downloaded. Considering that drastic video quality variation will significantly decrease the perceived video quality, we avoid those drastic layer change actions by restricting the video layer change level Δv to one. The difference of A_b and A_u is that, with A_b, both the WiFi and Bluetooth links are used to download the segment simultaneously, while with A_u, only the WiFi link is used. As a short distance wireless communication technology, Bluetooth can only help the client connect to the web server indirectly via tethering cellular data services. Since cellular services are charged at much higher rates than WiFi services, and using more links simultaneously may cost more energy, it is better to limit the usage of the Bluetooth link to reduce the bandwidth and energy cost. In addition, Bluetooth connection is not quite reliable as it is easy to be interfered. Therefore, sometimes we prefer to use WiFi only. For A_b actions, the download load of WiFi and Bluetooth will be determined based on their current available bandwidth. In other words, assuming that the available bandwidth of WiFi and Bluetooth are bw and bt, the segment size is SZ, then the downloading load of WiFi is $SZ \cdot bw/(bw+bt)$. A_w represents the waiting action, and the client will wait for W seconds, equal to the duration of one segment. Since SVC video is encoded into different layers and chopped into segments to be stored on the web server, as long as the segment has not been played yet, the higher enhancement layers can be requested to improve the perceived streaming quality and smoothness. Therefore, the smooth action A_s is introduced in our work.

There are three principles to follow for the smooth action. First, when the queue length is low, the smooth action will not be taken as there is a high probability to experience playback freeze soon. Thus the smooth action will be invoked only when the queue length is sufficiently large, such as when the queue length q is larger than a certain threshold T_s. Second, for the segment which will be played soon, we will not take the smooth action, because the requested enhancement layers may miss the playback deadline. Therefore we only take the smooth action for a few number of buffered segments which are stored in the tail part of the buffer. A smooth window with size L_s is defined in our work to determine how many buffered segments will be the candidate segments for the smooth action. In our approach, the last L_s buffered segments are in the smooth window as they are least likely to miss the playback

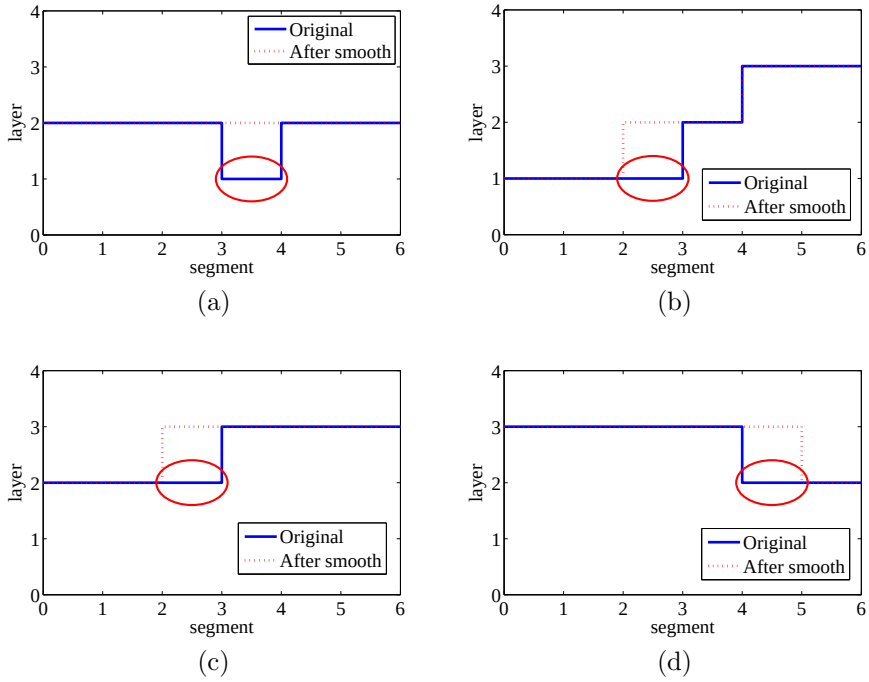


Figure 2.2: Different Smooth Scenarios.

deadline. The last principle of the smooth action is to ensure that it will not bring in any additional layer variation. As shown in Fig. 2.2, generally, we will take the smooth action in four different scenarios. Assume the smooth window size $L_s = 6$, and thus all segments in Fig. 2.2 are in the smooth window. If only one segment in the smooth window has the lowest number of the enhancement layer, then we will smooth that segment by requesting the same number of enhancement layers as the other segments, which is shown in Fig. 2.2-(a). When there are multiple layer variations in the smooth window, such as those in Fig. 2.2-(b), the smooth action will be applied to the segment with the smallest number of layers. To avoid additional layer variation caused by the smooth action, we will first smooth the segment at the edge of layer variation. For example, in Figs. 2.2-(c) and -(d), segments to be smoothed are marked with red circles.

For the optimal rate adaptation, one difficulty is to estimate the statistics of the wireless link bandwidth accurately. In this work, a state transition model is userd to estimate the bandwidth. Then, two discrete-time finite-state models are employed for describing the available bandwidth of the WiFi and Bluetooth links, respectively. The state transition models for the channels can be obtained according to the recent

Table 2.1: Rewards Associated with States

State $s_t = s$	Reward $R(s)$
$(*, *, *, *, *, *, *, *, N_T)$	0
$(q, *, *, *, *, *, *, *, *)$, if $q < T_{qmin}$	$-(Q_L - q) + \Delta q$
$(q, *, *, *, *, *, *, *, *)$, if $q > T_{qmax}$	$-q - \Delta q$
$(*, \Delta q, *, \Delta v, t, \Delta t, *, *, *)$	$\min(-\lvert\Delta v\rvert, -\lvert\Delta q\rvert) - \Delta t R_t$
other states	0
any state with smooth action	R_s

measurements or using the history data. With the wireless channel model, we can derive the available bandwidth state transition probabilities. The state at any time instance only depends on its immediately previous state. Given any state s and action a, the transition probability of the MDP can be given as

$$\mathcal{P}^a_{ss'} = \Pr\{s_{n+1} = s' | s_n = s, a_n = a\}. \tag{2.1}$$

Considering two wireless links in our model, and assuming they are independent, the transition probability is calculated as

$$\mathcal{P}^a_{ss'} = \Pr\{bw' | bw\} \cdot \Pr\{bt' | bt\}. \tag{2.2}$$

With the above transition probability, for any state $s = \{q, \Delta q, v, \Delta v, t, \Delta t, bw, bt, d\}$, if action A_b is taken, then the next state $s' = \{q', \Delta q', v', \Delta v', t', \Delta t', bw', bt', d'\}$ can be derived as

$$\begin{aligned} q' &= q - \lceil SZ^{v'}_{d+1}/(bw' + bt') \rceil + 1, \\ \Delta q' &= q' - q, \quad v' = v + m, \quad \Delta v' = m, \\ \Delta t' &= SZ^{v'}_{d+1} \cdot bw'/(bw' + bt'), \\ t' &= t' + \Delta t', \quad d' = d + 1, \end{aligned} \tag{2.3}$$

where $SZ^{v'}_{d+1}$ is the size of segment $d{+}1$ at quality version v', and m is the action value. The video segment information can be obtained by the client during the initialization step. Similarly, it is easy to derive the next state with A_u, A_w and A_s actions.

In order to measure how good the actions are, we define a reward value r associated with each action a. The reward value r_n at step n is determined by the previous state s_{n-1}, *i.e.*, $r_n = R(s_{n-1})$. As shown in Table 2.1, we define the reward values for

different states. In the reward function table, $*$ means that the state can be of any value. The reward of a state will be looked up from the top to the bottom in Table 2.1, until one entry is matched. When $d = N_T$, all the segments have been downloaded, and thus reward 0 is given. The reward functions are highly related to the streaming QoS. A higher reward is more desirable so the corresponding action is preferable. Video playback freezes could be the most undesirable experience, and thus a negative reward with a large magnitude is given when q is less than the threshold T_{qmin}, as there is a high probability of playback freeze when the queue length is very small. Such a negative reward can also reduce the startup latency, since we prefer the actions to fill up the buffer with less time to avoid large negative rewards. Therefore, low-layer segments will be requested when the queue length is small. Similarly, we also give a negative reward with a large magnitude when q is larger than the threshold T_{qmax}, in order to avoid buffer overflow. When the queue length is between the two thresholds, we prefer less variation of q, so we set the reward to no larger than $-\Delta q$. To obtain a smooth and high quality video streaming, we also give negative rewards when there are high fluctuations of Δv. According to [33], we can assume that the cost function of WiFi $C_w(x)$ is a constant (flat fee) independent of the traffic load:

$$C_w(x) = R_w, \tag{2.4}$$

and the cost function of Bluetooth $C_t(x)$ is a constant plus a linear function of the usage

$$C_t(x) = K_t + R_t x. \tag{2.5}$$

Here, the cost function of Bluetooth is quite similar to the cellular data plan. (Other cost models can be considered, which are not included in this work due to space limit.) When the usage exceeds a cap K_t, then additional data will be charged at a higher rate. Therefore, when the cap is reached, the usage of Bluetooth traffic must be restricted. We give a negative reward equals the traffic cost $\Delta t R_t$ when additional Bluetooth data transfer is used. To achieve a more smoothed perceived streaming quality, we will assign additional reward R_s upon the invoking of the smooth action.

The streaming policy π is a mapping of the possible action at each step. The

long-term reward $V^{\pi}(s)$ under policy π can be computed as:

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{n=0}^{N_T} r_n | s_n = s\right\} = \sum_{s'} \mathcal{P}^{a}_{ss'}[R^{a}(s) + \gamma V^{\pi}(s')], \tag{2.6}$$

where R is the next reward of taking action a at state s, γ is the discount rate and $0 \leq \gamma \leq 1$. The parameter γ makes a trade-off between myopic video quality and future interruptions and variations. A small γ lets future reward weigh less, and thus makes the adaptation decision more myopically. Meanwhile, when fewer future steps are considered, it also results in a more myopic decision.

Obviously, finding the optimal strategy policy $\pi^{*}(s)$ which can maximize the long-term reward is the goal of the reinforcement learning task of video streaming. Thus, our multi-link video streaming task can be finally formulated as an optimization problem:

$$\pi^{*}(s) = \arg\max_{\pi} \sum_{s'} \mathcal{P}^{a}_{ss'}[R^{a}(s) + \gamma V^{*}(s')]. \tag{2.7}$$

2.4 Practical Algorithm Design

Given the formulated video streaming process as an optimization problem, our goal is to find the best solution of the problem, which is also the optimal streaming policy. Theoretically, dynamic programming can be employed to solve the above optimization problem by value iteration. The computation time and the memory consumption of the dynamic programming algorithm are determined by the number of states. With about 50 segments, the computation time and the solution table may exceed one hour and 600 MBs on a high-end desktop, respectively. Therefore, this approach is not suitable for real-time adaptive streaming. To overcome this problem, we aim to develop a real-time best streaming action search algorithm to find a sub-optimal solution for the optimization problem formulated in the previous section.

2.4.1 Bandwidth Estimation

Rapid network load changes and short-term outages are difficult to predict, and the resultant available bandwidth for a session becomes a time-varying random process. Thus, instead of using a homogeneous Markov chain to estimate the available band-

width, in our work, a heterogeneous and time-varying Markov model is used to estimate the future bandwidth. The bandwidth of each link will be divided into several regions. Each region will represent a state of the channel state transition model, and the total number of the states is equal to the number of regions. Assume that there are n states, then an $n \times n$ transition matrix P will be used for the channel state transition model. Each element p_{ij} is the transition probability from state i to j. To obtain the transition probability, another $n \times n$ matrix C is used to count the number of transitions for each state. Once a segment has been successfully downloaded, the transmission bandwidth can be calculated by dividing the total size of the data transmitted over the total transmission time. Then the bandwidth region can be determined and we will increase the corresponding c_{ij} by one. p_{ij} is updated by the following equation:

$$p_{ij} = \frac{c_{ij} + 1}{\sum_{j=1}^{n} c_{ij} + n}. \tag{2.8}$$

Initially, if there is no history data available, $c_{ij} = 0$, and p_{ij} is set to $1/n$. The transition matrix will be updated after each segment has been successfully downloaded, so the transition matrix can better predict the future bandwidth variations with the recent measurements.

2.4.2 Real-time Search Algorithm

According to the formulation of the video streaming process, the calculation of the best long-term reward at state s with action a in (2.6) can also be written as:

$$Q^*(s, a) = R(s) + \gamma \sum_{s'} \mathcal{P}^a_{ss'} V^*(s'), \tag{2.9}$$

where $V^*(s')$ is the best long-term reward for state s'. It is easy to note that the best long-term reward for the current state is determined by all the possible future states. Since dynamic programming considers all the possible future steps to obtain the optimal solution, it results in an extremely long computation time. If only part of the future steps are considered, a sub-optimal solution can be obtained. Based on this idea, we develop a real-time recursive best-action search algorithm, which is shown in Algorithm 1.

To meet the requirement of the real-time search, an important issue is to reduce the search duration for each state to an acceptable value. We achieve this goal by

Algorithm 1 Real-Time Best-Action Search Algorithm

```
1: procedure GetBestAction(s)
2:     Initialize action ← −1, Q_max ← −∞
3:     Generate all possible actions A(s) for state s
4:     for all Action a ∈ A(s) do
5:         q ← RewardSearch(s, a, 0)
6:         if q > Q_max then
7:             Q_max ← q, action ← a
8:         end if
9:     end for
10:    return action
11: end procedure

12: procedure RewardSearch(s, a, d)
13:    q ← reward of (s,a)
14:    if d ≥ D then
15:        return q
16:    end if
17:    Generate all possible next states S′ of (s,a)
18:    for all s′ from S′ do
19:        Q_max ← −∞
20:        Generate all possible actions A′(s) for state s′
21:        for all Action a′ ∈ A′(s) do
22:            Q_t ← RewardSearch(s′, a′, d + 1)
23:            if Q_t > Q_max then
24:                Q_max ← Q_t
25:            end if
26:        end for
27:        q ← q + γP_ss′Q_max
28:    end for
29:    return q
30: end procedure
```

setting a small search depth D to invoke the search algorithm. For the current state s, all the possible actions $\mathcal{A}(s)$ will be enumerated. The recursive reward search algorithm is invoked to obtain the reward of state s with action a by enumerating all the possible future states S' and their associated actions $\mathcal{A}'(s)$.

2.4.3 Adaptive Search Depth

Search depth is an important issue in our work. The search depth can determine how good the search result is, and a larger value of depth will achieve a better result. Meanwhile, with the increment of the search depth, the search time to obtain the action for a segment will be increased exponentially. Therefore, the search depth can be viewed as a trade-off between the video quality and the search time.

Based on several preliminary experiment results, when the search depth D is larger than three, it will take more than two seconds to obtain a decision on the test Android smart phone. Thus, the maximum search depth D_{max} is set to three. As the perceived video streaming fluency is generally considered as one of the most important QoS for the user, the search depth D is determined by the current queue length in our work. We divide the buffer queue into three regions, $[0, q_1)$, $[q_1, q_2)$ and $[q_2, q_L]$. For each state, the search depth D is determined according to its queue length q as follows:

$$D = \begin{cases} 1 & \text{if } q \in [0, q_1) \\ 2 & \text{if } q \in [q_1, q_2) \\ 3 & \text{if } q \in [q_2, q_L] \end{cases} \tag{2.10}$$

When the queue length is low, there is a high probability that a playback interruption may occur soon, and thus a short search time and depth is preferred. When the queue length is high, there is sufficient time to search a deeper depth to obtain a better result.

2.4.4 Discussion

According to [64], the MDP can be viewed as a decision tree. The current state represents the root of the decision tree, and the future possible actions and states form the node and leaves. Since the recursive search will not try the next action until it reaches the leaves. Thus, our real-time search algorithm is a depth-first algorithm. It is easy to find that the computational complexity of our real-time search algorithm

is $O(b^D)$, where b is the total number of possible actions and D is the search depth which equals the total number of the remaining video segments. If we only search one step, it is a typical greedy algorithm. When the search depth is equal to the total number of video segments N_t, then it is exactly identical to the dynamic programming algorithm. There is no need to store all the states and actions in the stack while searching the tree, so the memory consumption of our recursive search algorithm is not high. Generally, the space complexity of our algorithm is bound by $O(bD)$.

2.5 Performance Evaluation

2.5.1 Testbed Implementation

In order to better evaluate the performance of our proposed video streaming algorithm, we established a realistic testbed to measure the performance with a real video stream and an Android mobile phone.

A smart phone with Android OS is used as the client. The smart phone is integrated with WiFi, 3G and Bluetooth wireless interfaces. Since WiFi and 3G cannot work together to access the Internet on Android OS, WiFi and Bluetooth are utilized together to transmit the video segments simultaneously according to our multi-link streaming solution.

We implemented a video streaming application which can request the video contents from the web server through the HTTP/1.1 protocol. The video streaming application contains three main components: the streaming action search module, WiFi connection management module and Bluetooth connection management module. When the video streaming process begins, the streaming action search module will make the decision on how many enhancement layers to be requested and how to assign the transmission load to each link. Once the decision is made, WiFi and Bluetooth modules send HTTP/1.1 requests to the server to fetch the corresponding video segments. This process will continue until the last segment is successfully fetched. As there is no available SVC decoder for Android OS yet, we only use the smart phone to request the video segments and record the transmission trace. With the trace, we can decode the SVC video on a PC to evaluate the experiment results.

The video information is stored in a simplified manifest file in our implementation. In the manifest file, first the general video information is listed: the total number of segments, the duration of a segment, the total number of enhancement layers and the

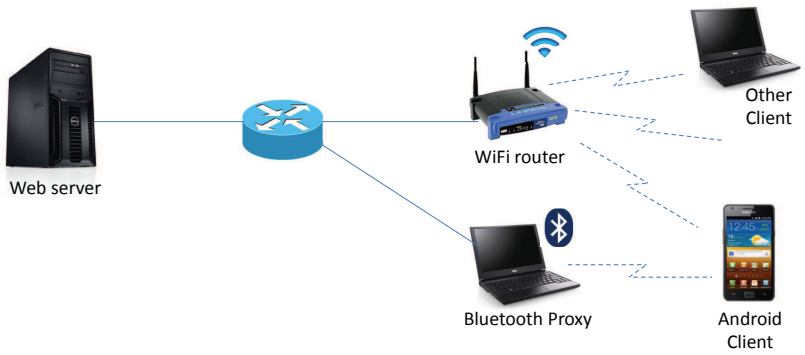


Figure 2.3: Testbed Network Topology.

Table 2.2: Video Coding Configurations

	Resolution	Avg. bit-rate (Kbps)	Std bit-rate deviation	Y-PSNR (dB)	Layer
Cfg 1	320×180	112.84	39.01	30.99	1
	320×180	238.94	88.84	32.63	2
	640×360	363.82	140.33	35.9	3
Cfg 2	640×360	235.4	92.09	35.37	1
	1280×720	531.1	215.97	38.53	2
	1280×720	1,056.9	469.1	41.5	3

correspondent resolutions. Then the detailed bit-rate of each segment and the size of each layer for each segment are listed. The manifest file will be transmitted to the client first before the stream begins.

As Bluetooth cannot connect the client to the web server directly, we implemented a Bluetooth proxy to let the client access the web server. The Bluecove Java Bluetooth library is used to implement the proxy on a laptop. When the client sends an HTTP/1.1 request, the proxy will forward the request to the web server, fetch the video segments from the server and also forward the fetched segments to the client.

2.5.2 Experiment Settings

The network topology of our testbed is shown in Fig. 2.3. The video streaming server is deployed on a web server, which is deployed on a desktop by Apache HTTP Server 2.2.24 with Ubuntu 10.04 OS. Both the wireless router and the Bluetooth proxy are connected with the web server through wired links. The wireless router is flashed with OpenWrt to set up customized configurations. In our experiment, the WiFi network is configured to be in the IEEE 802.11b mode and the wireless transmission rate is set

to 1 Mbps or 2 Mbps. The minimum round-trip time (RTT) of WiFi without queuing delay is about 13.19 ms. The Bluetooth proxy runs on a laptop which is equipped with a dual-core 2.53 GHz Intel CPU and 2 GB memory. We used a Samsung i9100 Galaxy II smart phone with Android 4.0.3 OS as the client. The Samsung i9100 is equipped with a dual-core 1.2 GHz Cortex-A9 CPU and 1 GB RAM. The minimum RTT between the smart phone and the proxy is about 25.83 ms, and between the proxy and the server is about 1.79 ms, respectively. Thus the total minimum RTT for the Bluetooth link is about 27.62 ms.

The open-source SVC codec JSVM [57] is used to encode a test video "Big Buck Bunny [1]", and we did not pack the encoded video with any container. According to [39], the multi-segment strategy is adopted in our implementation. The test video is first chopped into small segments and then encoded into a base layer and two enhancement layers. In order to better test the feasibility and robustness of our approach, two different configurations of the encoded videos are employed, which are shown in Table 2.2. The frame rate and Group of Pictures (GoP) of both configurations are set to 24 fps and 8 frames which include one I frame and seven hierarchically predicted B frame. To ensure that each segment contains complete GoPs, the length of the two configuration 1 and 2 are set to 17 frames (one independent I frame and two GoPs) and 49 frames (one independent I frame and six GoPs) per segment respectively. Therefore, the duration of each segment is about 700 ms and 2,042 ms for the two configurations, respectively. In our experiments, $N_t = 200$ segments are selected for configuration 1 and $N_t = 100$ segments for configuration 2.

At the client, the buffer size is set to $q_L = 20$ segments, which means that it can cache at most 20 video segments. T_{qmin} and T_{qmax} are set to 2 and 18 segments, respectively. For the channel state transition model, the available bandwidth for each link is divided into four regions, and thus two 4×4 transition matrices were used for the WiFi and the Bluetooth links, respectively.

To reduce the search time of our real-time best-action search algorithm, we implemented it in a non-recursive way. According to the experiments, we found that the running time of the algorithm is about 2 ms when the depth $D = 1$, 42 ms when $D = 2$, and 513 ms when $D = 3$, respectively. Therefore, we set the adaptive searching depth region as $q_1 = 8$ segments and $q_2 = 15$ segments.

2.5.3 QoS Metrics

Several objective tests were conducted to evaluate the performance of the proposed approach using several QoE related QoS metrics. For instance, according to [55], the layer variations will decrease the users' watching experience, and thus the number of layer variations was used to reflect the smoothness in our experiments. The performance metrics used in our experiments are listed as follows:

- **Startup latency (SL)**: It is defined as the duration from sending out the first segment request to the beginning of playing back the first segment, which can be also regarded as buffering time.
- **Playback fluency**: The number of segments that miss the playback deadline, and the playback freeze ratio, which calculates the percentage of the duration of playback freezes over the total video streaming time are evaluated.
- **Average playback quality (APQ)**: The average PSNR of all received frames.
- **Layer variation**: To measure how smooth the perceived video quality is, we count the number of layer variations.

Besides, the Bluetooth traffic usage was also utilized as one evaluation metric.

2.5.4 Experiment Results

The available bandwidth in the wireless network is an uncontrollable, random process. In order to make the experiments reproducible, the traces of the wireless bandwidth were recorded [77], and a Linux traffic shaping command tool *tc* is utilized to shape the bandwidth based on the traces. For each scenario, the experiment is repeated 10 times to obtain the average. Since the commercial DASH solutions are not open-sourced, it is difficult to modify them to support multi-link and SVC. Thus, we compared our algorithm (RTRA) with the state-of-the-art one, the rate adaptation video streaming algorithm by K. Evensen (KERA) published in [27]. The only difference of the implemented KERA algorithm is that we modified it to support SVC videos. We tested the single-link case by enabling only one link during the experiments for both algorithms (RTRA_S and KERA_S) as well, which can also validate the benefit of multiple links.

Slow-changing Bandwidth Scenario

First, we investigate the performance with the slow-changing bandwidth scenario. To generate the bandwidth variation, we added slow-changing on/off background traffic by letting another laptop request a large file and then letting it sleep for 10 seconds. This procedure will repeat during the whole experiments. The maximum WiFi rate is set to 1 Mbps for the first video configuration and 2 Mbps for the second one, respectively.

The parameter α is set to one to make a good trade-off between the smoothness and video quality. As the total sizes of all the video segments of the first configuration are 1.9 MB for the base layer, 2.13 MB for the first enhancement layer, and 2.1 MB for the second enhancement layer, we set the maximum threshold of the Bluetooth traffic to $BT_t = 2$ MB to limit the overuse of Bluetooth traffic. Similarly, the $BT_t = 5$ MB is set for the second configuration video.

The results of the first set of experiments are shown as Case 1 in the first and fourth rows of Table 2.3. We can notice that the proposed RTRA needs significantly less buffering time than KERA under both configurations. As a low buffer level brings a large-magnitude, negative reward which is undesired, the proposed RTRA will only request the base layer for the first several segments to quickly fill the buffer. KERA does not have such a low-layer start mechanism, so it may request high-layer video segments at the beginning, which results in a longer SL. Even with a single link, RTRA_S still only experiences almost a half SL when compared with that of KERA_S. Without the Bluetooth link, slightly longer SLs for both algorithms are observed.

As the aggregated bandwidth is sufficient to support the base layer for both video configurations, there is no segment missing the playback deadline for both algorithms with two links, and therefore 0% playback freeze ratio is achieved as well. Even with a single link, the proposed RTRA_S can still avoid playback freeze. By giving a large-magnitude negative reward to the low buffer level, the proposed RTRA can maintain a relatively high buffer level, which guarantees the playback fluency as long as the average bandwidth can support the base layer. For KERA_S, the sudden bandwidth decrease makes the bandwidth estimation inaccurate. Thus, when there is no other link can compensate for the bandwidth, the low bandwidth can no longer support the high quality video and playback freeze happens.

The proposed RTRA can achieve similar perceived video quality as KERA. With

Table 2.3: Experiment Results

			SL (ms)	# of missed segments	Playback freeze ratio	PSNR (dB)	# of layer variation	BT traffic (KB)
Cfg 1	Case 1	RTRA	1,404	0	0%	34.07	19	1,624
		RTRA_S	2,184	0	0%	31.9	48.1	0
		KERA	2,718	0	0%	33.78	109	1,613
		KERA_S	3,144	26.1	5.36%	33.05	141.4	0
	Case 2A	RTRA	1,447	0	0%	32.19	40	1,687
		RTRA_S	2,393	56	14.4%	31.02	26.6	0
		KERA	3,215	0	0%	31.64	90	1,855
		KERA_S	3,661	98.7	21.63%	31.44	41.8	0
	Case 2B	RTRA	1,360	0	0%	35.72	5.8	1,136
		RTRA_S	1,597	0	0%	34.11	18.4	0
		KERA	1,985	0	0%	35.46	35.5	1,390
		KERA_S	2,744	0	0%	34.92	49.7	0
Cfg 2	Case 1	RTRA	3,573	0	0%	38.17	18.8	2,047
		RTRA_S	3,635	0	0%	36.51	31.8	0
		KERA	5,409	0	0%	38.24	59.9	2,466
		KERA_S	6,204	0.6	0.18%	37.73	57.6	0
	Case 2A	RTRA	5,131	0	0%	36.56	36.3	2,273
		RTRA_S	6,046	18.3	10.71%	35.64	13.2	0
		KERA	5,433	0	0%	36.57	42	2,597
		KERA_S	6,697	25.4	13.49%	36.0	24.5	0
	Case 2B	RTRA	2,505	0	0%	40.16	14.8	2,182
		RTRA_S	2,644	0	0%	39.17	17.4	0
		KERA	5,534	0	0%	40.34	31.6	2,256
		KERA_S	5,778	0	0%	39.95	38.7	0
	Robustness Case 2A/B	RTRA	4,023	0	0%	36.3	30	2,213
		RTRA	2,306	0	0%	40.37	7.1	2,102

the low-layer startup, the buffer can be filled very quickly. When the buffer level reaches a certain level, drastic buffer level increment will bring a negative reward, and thus the video quality will be upgraded to avoid such a penalty. In this way, high quality video streaming can be guaranteed. As KERA is quite greedy, it always tries to request the highest possible layer, and KERA can achieve the similar quality as RTRA.

Since layer change may bring a negative reward, the proposed RTRA tends to maintain the current video quality. Thus, the average number of layer variations for RTRA is quite small. As mentioned before, KERA is a greedy algorithm. One of the drawbacks of the greedy algorithm is that the smoothness is ignored as it only considers the instantaneous video quality. Thus, there is a significant number of layer variations for KERA. With a single WiFi link, as WiFi cannot sustain the first enhancement layer, sometimes it has to switch from the first enhancement layer to base layer in order to avoid playback freeze, which results in more layer variations for RTRA_S than RTRA.

Furthermore, with the cap of Bluetooth traffic, RTRA uses much less Bluetooth traffic than KERA which can avoid the additional cost and negative reward.

Rapid-changing Bandwidth Scenario

We then evaluate the performance with the rapid-changing on/off background traffic sharing the WiFi link with a much shorter on or off duration. Two different cases are generated: 1) We let another laptop request a file (600 KB) and then sleep for only 1 second before repeating the requesting procedure. In this way, there are some positive spikes of the available bandwidth in the WiFi link. 2) To generate the negative spikes for the available bandwidth in the WiFi link, we let the laptop request a small file (100 KB) and then sleep for 10 seconds. We conducted the two sets of experiments under the above two types of on/off background traffic.

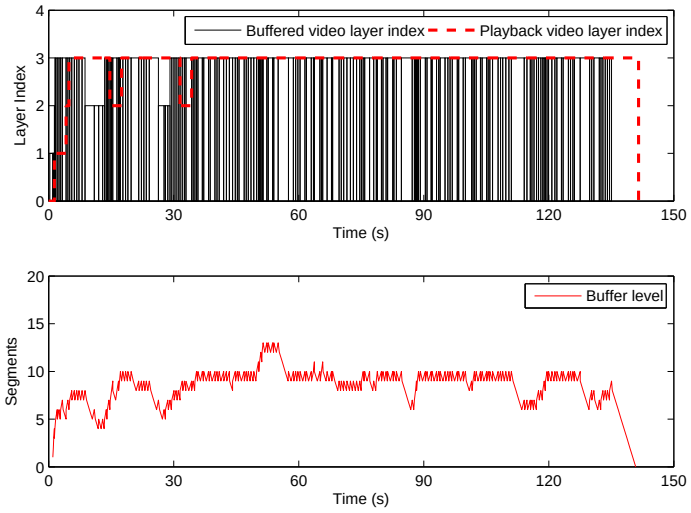
In these two sets of experiments, the purpose is to evaluate the performance of the proposed RTRA with short-term bandwidth variations. We show the experiment results in the case 2A and case 2B rows of Table 2.3 for the positive and negative spikes cases, respectively. For the positive-spike case, the base layer cannot be sustained only with WiFi, so there are many segments missing the playback deadline under both video configurations for both algorithms with single link. When the second link is available, it is sufficient to support the base layer, therefore no segment missed the

playback deadline and zero playback freeze ratio.

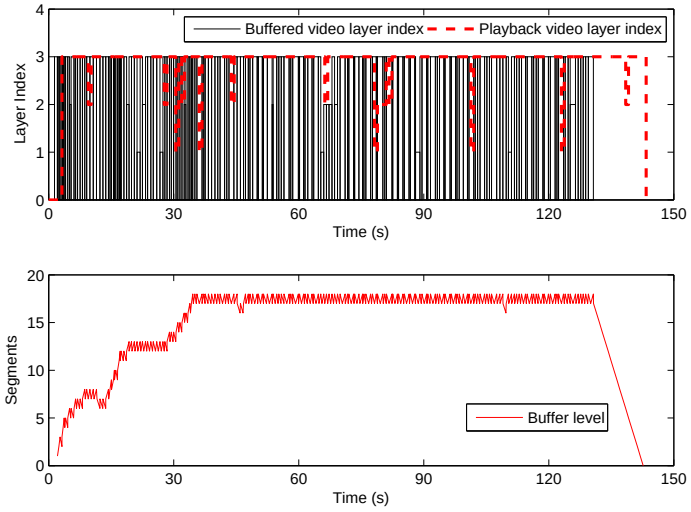
Under the positive-spike traffic, as the average bandwidth can support the base layer only, the RTRA algorithm tried to stay with the base layer. Even when the bandwidth was increased dramatically, quickly increasing the buffer level will bring a higher reward than improving the video quality by requesting more enhancement layers. Therefore, the smoothness can be guaranteed. While under the negative spike traffic, the buffer will be quickly filled to reach a certain stable level. More enhancement layers will be requested to improve the video quality with a low playback freeze probability. The sudden bandwidth decreases do not have much impact on the video request decision, as the buffer has accumulated enough segments to maintain the current quality. KERA is very sensitive to the bandwidth variation, therefore it may switch to the base layer only during the negative spikes. As a result, compared to RTRA, a much lower smoothness can be achieved by KERA, no matter with one or two links.

To further demonstrate the performance of the proposed approach, we select one run of the experiment of case 2B and show the playback traces and buffer occupancy status of the two algorithms in Fig. 2.4. In the top sub-figures, the black rectangles represent the requested segment layer index, and the red curves represent the segment playback index. There are only several layer variations at the beginning of the streaming process. With the updated state transition matrix, the prediction of future bandwidth becomes more accurate. When the buffer level reaches a certain level, RTRA can stay with the second enhancement layer all the time to obtain a high smoothness. In Fig. 2.4-(b), there are lots of layer variations during the whole streaming process. We can also notice that the buffer was almost full from the thirty-first second. With the almost full buffer, those layer variations indicate substantial bandwidth wastage.

Figure 2.5 compares the video layer and bandwidth traces using KERA and the proposed RTRA under the first video configuration. All three sub-figures clearly show the sensitivity of KERA with regard to the variation of bandwidth. Layer variations occur frequently with KERA, which is along with the bandwidth variation. Different from KERA, the proposed RTRA only requests the second layer unless the buffer occupancy is accumulated to a certain level before it requests the higher layer, which is shown in Fig. 2.5-(a). As there may be negative reward for the layer variation, it enables RTRA being immune to the bandwidth spikes, which can be found from Figs. 2.5-(b) and -(c).

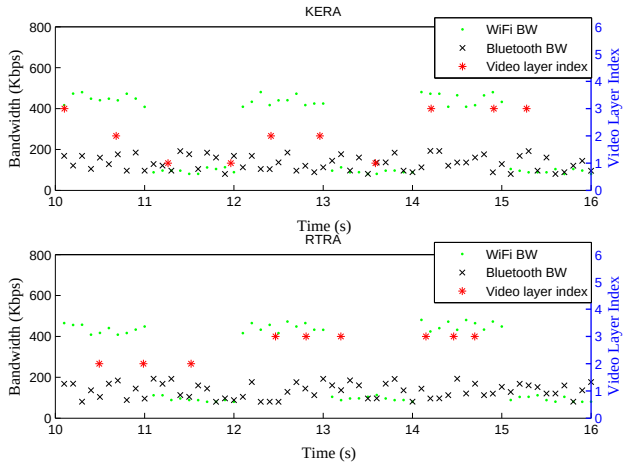


(a) RTRA

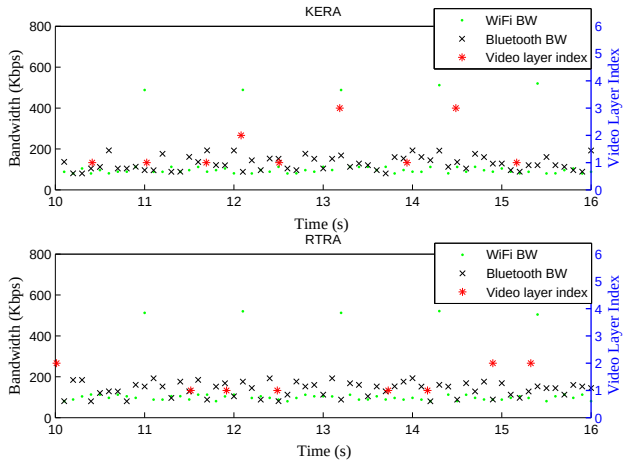


(b) KERA

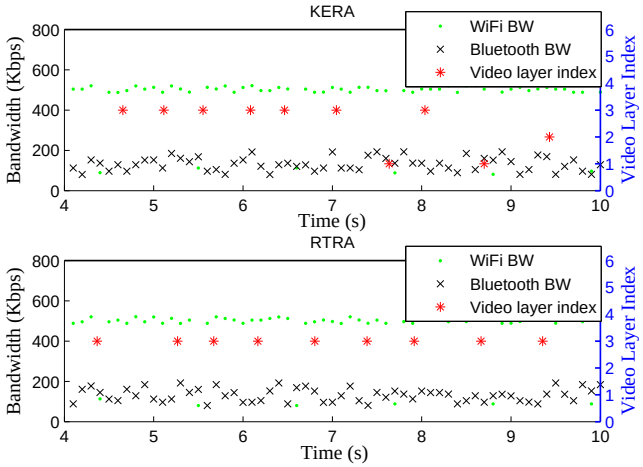
Figure 2.4: Playback Traces and Buffer Occupancy Traces.



(a) Case 1



(b) Case 2A



(c) Case 2B

Figure 2.5: Comparison of Experiment Traces.

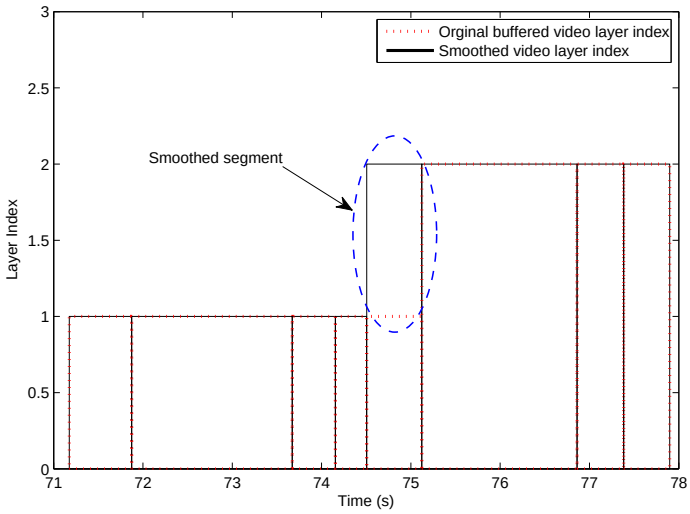


Figure 2.6: Effect of The Smooth Action.

For RTRA, when the buffer level reaches a certain level, the smooth action will become available. By taking the advantage of SVC coding, we split the video segments into layers. Thus the enhancement layer can be requested individually to improve both the smoothness and quality. In Fig. 2.6, the effect of the smooth action is shown. The segment in the blue circle is smoothed by requesting the next enhancement layer.

Robustness Evaluation

Finally, we evaluate the robustness and the effectiveness of the proposed RTRA algorithm based on the channel state transition model. In the previous experiments, the initial state transition probability between any two states is set to be equal. With the recorded bandwidth traces, we can process the traces and obtain the state transition rate of the traces. Then, in the following experiment, we used the state transition rate to initialize the state transition probability matrix. The matrices are used as the initial matrix in our experiment. The experimental results are shown in the last row of Table 2.3.

Since the initial state transition probability matrix in this experiment can accurately reflect the bandwidth variations, the results outperforms the previous results with a very limited margin in every aspect. In the previous practical approach, although the initial state transition matrix is not accurate, by updating it with the measured throughput, we still can achieve a satisfactory performance. Meanwhile,

this also confirms the robustness of the proposed algorithm, as the accurate initial state transition matrix is not essential.

2.6 Conclusions

In this chapter, we proposed a real-time adaptive best-action search algorithm for video streaming over multiple wireless access networks. First, we formulated the video streaming process as an MDP. To achieve smooth video streaming with high quality, we carefully designed the reward functions. Second, with the proposed rate adaptation algorithm, we can solve the MDP to obtain a sub-optimal solution in real time. Last, we implemented the proposed algorithm and conducted realistic experiments to evaluate its performance and compare it with the state-of-the-art algorithms. The experiment results showed that the proposed solution can achieve a lower startup latency, higher video quality and better smoothness.

Chapter 3

Maximum-Utility Scheduling for Multimedia Transmission in Drive-Thru Internet

In the previous chapter, we have discussed the rate adaptation of video streaming for mobile user with multiple wireless links. When there are multiple video streams, the transmission scheduling needs careful consideration. In this chapter, we investigate the scheduling of multimedia transmissions in Drive-thru Internet. A utility model is devised to map the throughput to user's satisfaction level. The scheduling problem is formulated as an optimization problem to maximize the total utility. We devise a practical heuristic algorithm based on the concept of utility potential to obtain real-time solution. We further implemented the solution and conducted extensive simulations using NS-3, and the simulation results show that the proposed heuristic algorithm can outperform the state-of-the-art one.

3.1 Introduction

There is an increasing demand of Internet service nowadays, even for people in the moving vehicles. Due to the high mobility for them, it is quite challenging to provide high-speed, low-cost and reliable Internet services. Vehicle to infrastructure communications (V2I) is a promising solution to cope with the mobility challenge. It is a promising way to disseminate safety messages, such as the traffic accident and detouring videos to alleviate traffic congestion. Road Side Units (RSU) are deployed

along the road to act as Internet access points (AP). When vehicles pass through the coverage area of a RSU, they can access the Internet by connecting to the RSU. The above system, termed Drive-thru Internet, has attracted many research interests recently [21, 88, 89].

Multimedia applications in Drive-thru Internet have emerged recently [10, 53, 61, 67]. For instance, video and voice are desirable media to deliver advertisements, news and etc. to improve the experience of passengers. Meanwhile, multimedia transmission can also take an important role to enhance road safety [56]. For example, video clips of dangerous road conditions taken by the vehicles ahead can assist overtaking on rural roads [71]. However, the transmission of videos and other multimedia traffic over Drive-thru Internet can be quite challenging. Due to mobility, the sojourn time of the vehicle in the RSU's coverage range is limited which restricts the total amount of data that can be transmitted. When multiple vehicles are within the coverage of a single RSU, the competing of limited bandwidth will further reduce the throughput of each vehicle. Therefore, the scheduling problem of multimedia transmission in Drive-thru Internet is an open issue.

In this chapter, the scheduling problem of multimedia traffic including video, voice and data, among multiple vehicles accessing a single RSU is investigated. Videos encoded with various techniques are taken as sample applications in our work, and other kinds of multimedia traffic can be easily incorporated to our approach as well. Motivated by [46], a time division multiplex access (TDMA)-based scheduling algorithm is devised to maximize the total utility of the transmitted videos. The utility model is devised as a function to map the total flow throughput to user's satisfaction level, such as the decoded quality of video. Channel time will be divided into time slots. For each time slot, only one vehicle will be allocated to transmit. The scheduling problem is then formulated as an optimization problem to maximize the total achieved utility. The optimal solution is difficult to obtain since the solution depends on the future arrivals, and the optimal decision cannot be simply decoupled into per-slot optimization problems. To solve this challenging issue, we propose a heuristic algorithm based on the concept of utility potential.

The main contributions of this chapter are summarized as follows. First, considering the new problem of scheduling heterogeneous multimedia flows in Drive-thru Internet, we define utility models to map the throughput to user's satisfaction level, and formulate the optimization problem to maximize the total utility. Second, by converting the optimization problem into a finite-state decision problem with the as-

sumption that the future arrivals are known, the optimal results can be obtained by a searching algorithm. As the assumption is not realistic and the searching algorithm is too complicated to use, we propose a heuristic algorithm based on the utility potential to make scheduling decisions in real time. Finally, we implement the proposed scheduling algorithm and conduct extensive simulations using NS-3 [4] to evaluate its performance. The results show that the proposed heuristic algorithm can substantially outperform the state-of-the-art one [89] in terms of total utility, resource utilization, and fairness.

3.2 Related Work

Vehicle to infrastructure communications (V2I) has attracted research interests in recent years, and there are extensive work studying the Drive-thru communication scenarios [9, 19–21, 48, 65, 89, 91]. Tan *et al.* proposed an analytical model in [65] to characterize the downlink average throughput and distribution achieved for each vehicle during the sojourn time by a Markov reward model, with the assumption that the wireless resource is evenly shared by all the passing vehicles. In [91], the authors tried to model the uplink performance of the last-hop Drive-thru communication with the consideration of vehicle density and speed. They showed that with an optimal admission control scheme, the throughput can be maximized for each Drive-thru vehicle. For the non-real-time traffic transmission of V2I communication, Alcaraz *et al.* proposed a contention-free, poll-based link-layer scheduling algorithm [9], aiming to reduce the residual queue backlog for each user by assigning the user with the lower packet error rate a higher priority. Cheung *et al.* studied the random access problem in the Drive-thru scenario [20, 21]. They formulated the optimal transmission problem as a finite-horizon sequential decision problem and solved it by dynamic programming. Due to the high computational complexity, only an offline solution was obtained. Zhang *et al.* proposed an application-layer service scheduling algorithm with the consideration of both service deadline and data size [89]. The variation of communication distance was ignored in their work and a constant transmission rate was assumed.

Recently, utility-based TDMA scheduling has been applied in many works [7, 14, 17, 18, 36, 46, 54, 90]. Considering the mobility pattern of unmanned aircraft system, [7] formulated the data collection problem as a potential game between the unmanned aircraft and the ground nodes to maximize the energy efficiency of the ground nodes.

How to improve both spectrum and energy efficiency was considered in [54]. Utility-based flow control optimization problems were formulated for wireless sensor networks with lifetime constraint [17, 18]. In [46], the problem of utility maximization of concurrent transmission scheduling in UWB networks was studied. The definition of utility function depends on the traffic type. By modeling the network into a graph, a heuristic scheduling algorithm based on the "exclusive region" was proposed. Similarly, Hwang *et al.* considered the video multicast problem in wireless mesh networks which was also modeled into graphs [36]. By jointly considering routing and scheduling the transmissions over the graph, the system utility was maximized. In [14], the optimal utility scheduling of adaptive video streaming over a small cell network was formulated into an optimization problem and solved by liner programming. Scheduling multimedia applications over heterogeneous wireless networks was investigated in [90], and a novel distributed approach was proposed to maximize the utility which jointly considered QoS, reliability and availability. However, the mobility of nodes was not considered in the above work, and we cannot apply or extend the above solutions easily to support multimedia services in Drive-thru network, which motivated this work.

3.3 System Model and Problem Formulation

We consider how to schedule multiple vehicles to transmit the multimedia data in a highway Drive-thru scenario, which is shown in Fig. 3.1. In our scenario, both uplink and downlink are considered, such that each vehicle may request or upload multimedia data through V2I communications to RSU. The RSU is deployed on the road side with the transmission range of R. Due to the high cost, it is impossible to deploy enough RSUs to cover the entire road, and each RSU only covers a certain range. As a result, each vehicle moving on the highway can transmit multimedia data only when it passes a RSU. As mentioned before, videos encoding with various techniques are considered as examples in this work, and other multimedia traffic, such as voice, image, data, and etc., can also be incorporated into our approach easily. For each vehicle, we consider only one video flow will be carried, and it will be encoded by one encoding technique. The encoded video will be chopped into small packets for transmission. A vehicle is allowed to transmit only during its allocated time slots. When the multimedia data is successfully transmitted, it does not request wireless network access any more.

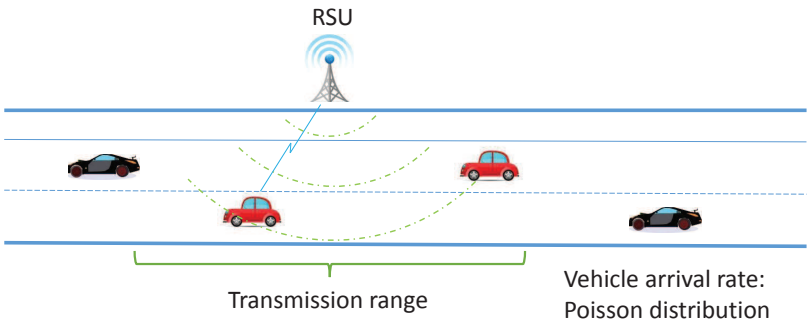


Figure 3.1: System Model.

3.3.1 Vehicle Mobility Model

According to the vehicle traffic model [28, 91], the arrival process of vehicles can be modeled as a Poisson distribution process. The arrival rate of the vehicles for each lane is assumed to be λ (number of vehicles per time unit). With multiple lanes in both directions, the aggregated arrival still follows a Poisson distribution. To simplify the notation in the following, we assume that all vehicles are in a single lane and the same approach can be applied to multiple lanes cases.

With the consideration of a speed limit in highway, it is reasonable to assume that all vehicles are moving with the average speed v (distance traveled per time unit). Let k represent the vehicle density (number of vehicles per kilometer), and $k = \lambda/v$. Then the average number of vehicles passing an observation point per time unit is $q = kv$. According to [28], the relationship between vehicle density and speed can be modeled as:

$$v = v_f(1 - k/k_{max}), \tag{3.1}$$

where v_f is the free-flow speed and k_{max} is the vehicle jam density. With the vehicle's speed v, the total duration of vehicle with the coverage range of the RSU is easy to obtain:

$$T_s = 2R/v, \tag{3.2}$$

where R is the transmission range of the RSU. The number of vehicles in the coverage range of the RSU can also be calculated:

$$N_v = \lfloor T_s k v \rfloor. \tag{3.3}$$

3.3.2 Wireless Model

As we focus on the resource allocation in a general wireless system that can support transmission scheduling, we consider a general wireless communication model, which suffers from fading, shadowing and path loss. It is possible that the RSU uses cellular or WiFi technologies. If using the cellular technology, the TDMA mode can support the proposed scheduling directly. Although CDMA is found to be most efficient for the circuit-switching based cellular system, as it can naturally take advantage of the voice activity factor and achieve multiplexing gain. When all cellular systems migrated to all-IP networks, the advantages of CDMA is not as obvious. In LTE/A, TDMA + OFDM (can be viewed as an enhanced FDMA) are widely used. If using

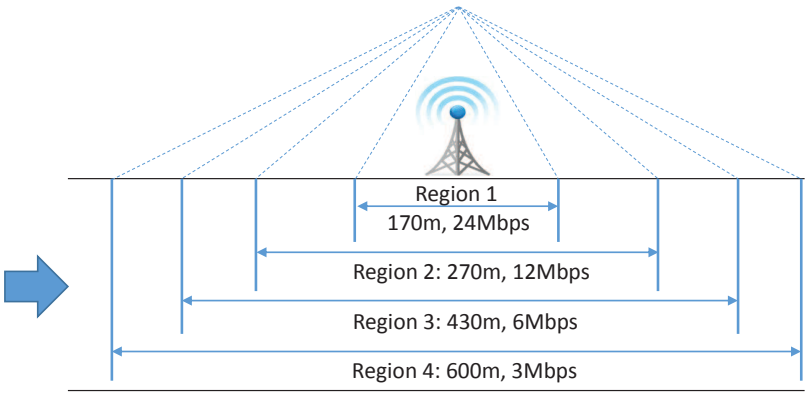


Figure 3.2: Wireless Model.

WiFi with contention-based MAC, then we may rely on the association process to deploy the proposed scheduling algorithm.

High speed often results in high Doppler shift which affects communication quality. However, when vehicles are in high speed, typically, their speed (high way, train) does not change frequently, so we can estimate the speed and compensate the Doppler effect. Also, in high speed trains, a smart solution is to install an array of antenna on top of the train. The first antenna is used to do channel measurement, and after a deterministic time, the following antennas will experience the almost same channel, which can also improve the overall communication performance.

Broadband wireless systems typically support scalable modulation and coding techniques to adjust the data rate according to the received signal to noise ratio (SNR). Further, the wireless channel quality between the RSU and the moving vehicle highly depends on the path loss which is a function of the communication distance d. Thus, we can simplify the wireless model by mapping the transmission distance to the data rate. Although shadowing and fading may also affect the instantaneous signal strength and should be considered for the data communication, but this part is not considered by the scheduler to reduce the control overhead and simplify the scheduling decision. This strategy is also adopted by references [21, 65]. Therefore, the coverage area of the RSU is divided into several regions, which is shown in Fig. 3.2. Each region has a constant transmission data rate. The closer to the RSU, the higher the data rate it supports.

3.3.3 Utility Model

Different from the previous work, we consider different utility functions for different types of video encoding techniques. As a widely accepted video quality metric, Peak Signal-to-Noise Ratio (PSNR) is used to represent the value of utility function [45,74]. The utility function F_u maps the amount of transmitted video data to the decoded quality, which can be represented by:

$$U = F_u^j(D), j = \{1, 2, \ldots, N_e\}, \tag{3.4}$$

where D is the amount of transmitted video data, N_e is the total number of encoding techniques. For each video, with a different encoding technique, the required amount of data to decode is different. For example, for the H.264/AVC codec, only if all the video packets are received successfully, the video can be decoded. While for

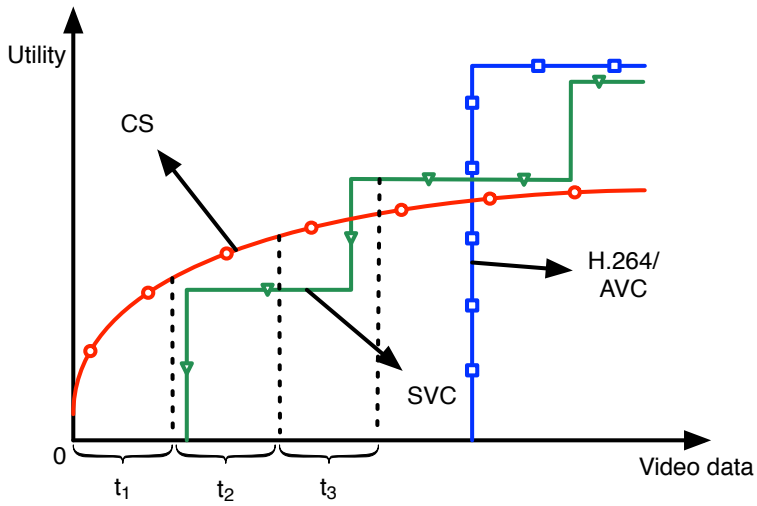


Figure 3.3: Utility Model.

compressed sensing (CS) based encoding technique, the video can always be decoded with different quality depending on the amount of received data. Therefore, the utility function can be either discrete or continuous functions. For instance, the utility function F_u for H.264/AVC codec can be defined as:

$$F_u = \begin{cases} u, & \text{if } D \geq S_v, \\ 0, & \text{otherwise,} \end{cases}$$

where S_v is the total size of the encoded video clips. The above discrete utility function is shown as the blue curve with square marker in Fig. 3.3. Similarly, for Scalable Video Coding (SVC), as the video is encoded into layers, with more video data transmitted, more enhancement layers can be decoded to improve the video quality. Thus, the utility function for SVC is a stair-case discrete function, as the green curve with triangle marker shown in Fig. 3.3. While for the CS based encoding technique, since the video can always be decoded with any amount of received data, the utility function can be written as a quadratic function [74]:

$$F_u = aD^b + c,$$

where a, b and c are the parameters determined by the encoder, shown as the red curve with circle marker in Figure 3.3.

3.3.4 Problem Formulation

As there is a speed limit for highway, it is reasonable to assume that all the vehicles move with the average speed v. Similar to the typical MAC approach, the total sojourn time can be divided into time slots with the duration of Δt. When the time slot Δt is small enough, there will be at most one vehicle arrive in the next time slot. As each vehicle travels with stable speed in the coverage of the RSU, the total distance traveled during each time slot is identical. If we consider the total distance traveled during one time slot as a small region, then the whole coverage of the RSU can be regarded as virtually divided into small pieces. The number of small regions is equal to the number of total time slots traveling within the RSU's coverage.

According to the wireless model, since the available transmission rate is mainly dominated by the relative distance d to the RSU, the transmission rate for each region can be represented by a vector $\mathcal{W} = \{w_1, w_2, \ldots, w_{N_T}\}$, where N_T is total number of

small regions and $N_T = T_s/\Delta t$. The value of each element w_i can be easily obtained by the wireless model based on the relative distance to RSU.

Since the wireless channel is broadcast and shared in nature, in each time slot, it only allows at most one vehicle to transmit. Let $t = 1$ represents the first time slot, for the t-th time slot, the transmission scheduling decision for each small region can be represented by a vector:

$$\mathcal{A}(t) = \{a_1(t), a_2(t), \ldots, a_{N_T}(t)\}, \tag{3.5}$$

where

$$a_i(t) = \begin{cases} 1, & \text{vehicle in the } i\text{-th region is assigned,} \\ 0, & \text{vehicle in the } i\text{-th region is NOT assigned.} \end{cases} \tag{3.6}$$

There is a constraint $\sum_{i=1}^{N_T} a_i(t) \leq 1$ for the vector $\mathcal{A}(t)$, which means that at most one vehicle will be assigned to transmit for slot t.

With multiple vehicles competing for the limited wireless resources, the problem is formulated as finding the best time slot allocation for a total time period T to schedule the transmission, such that the total utility is maximized. Therefore, we formulate the utility maximization problem (UMP) as:

$$\textbf{UMP: max} \ \sum_{i=1}^{N_c} U_i + \sum_{n=1}^{T} \mathbf{E}(U_n), \tag{3.7}$$

$$\textbf{s.t.} \ a_j(t) \in \{0, 1\}, j = 1, 2, ..., N_T, \tag{3.8}$$

$$\sum_{l=1}^{N_T} a_l(t) \leq 1, \tag{3.9}$$

where, $\sum_{i=1}^{N_c} U_i$ is the total utility of the N_c vehicles currently in the RSU's transmission range, and $\sum_{n=1}^{T} \mathbf{E}(U_n)$ is the expectation of total utility achieved of future arrival vehicles in the next T time slots.

For the current N_c vehicles in the RSU's transmission range, the i-th vehicle is supposed to arrive at t_i, and the decision vector $[a_1(t_i), a_2(t_i + 1), \ldots, a_{N_T}(t_i + N_T - 1)]$ indicates whether the vehicle is allocated to transmit. Then the achieved utility U_i of the i-th vehicle is obtained by the utility function F_u based on the total throughput:

$$U_i = F_u \left(\sum_{j=1}^{N_T} a_j(t_i + j - 1) w_j \Delta t \right). \tag{3.10}$$

Since the future arrival information is unavailable, only the expectation of utility achieved by future arrival vehicle can be calculated. For the future arrived vehicle, the encoding technique of the video to transmit can be selected from the set of N_e encoding techniques with certain probability P_e. Assume that there is a new vehicle arrives in the next time slot with probability P_a, then the expected utility $\mathbf{E}(U_n)$ for the new arrival vehicle can be calculated by:

$$\mathbf{E}(U_n) = P_e P_a \sum_{i=1}^{N_e} F_u \left(\sum_{j=1}^{N_T} a_j (t_n + j - 1) w_j \Delta t \right). \tag{3.11}$$

Apparently, the formulated UMP problem is an integer programming problem [2]. Since the variable $a(t)$ in UMP can be only 0 or 1, the formulated UMP is a 0-1 programming problem and it is NP-hard according to [3]. For each time slot, N_v vehicles in the RSU's transmission range correspond to N_v choices for the scheduler. As the variable $a(t)$ must be solved sequentially, the decision at the current time slot $a(t)$ will affect the future decision. With the scheduling for T time slots, the computation complexity is $O((N_v)^T)$.

Meanwhile, without the future arrival information, it is even more difficult to obtain the optimal solution. Therefore, in the next step we first consider how to obtain the optimal solution with the assumption that the future arrival information is known already, which is not practical and serves as a benchmark. Then we consider how to devise an heuristic algorithm which is only based on the current vehicle information to achieve high utility.

3.4 Algorithm Design

3.4.1 Optimal Solution

With the assumption that the RSU has the knowledge of future vehicle arrival pattern, the formulated problem can be translated into a finite state sequential decision problem.

In our problem, the goal is to obtain the maximum utility achieved in T time slots. For each time slot t, we define state S_t to represent the current status of N_v vehicles in the RSU's coverage:

$$S_t = (V_1, V_2, \ldots, V_{N_v}), \tag{3.12}$$

where V_i is the information set of the i-th vehicle in the coverage and is defined as:

$$V_i = \{I_i, P_i, u_i^T, D_i^S, D_i^R\}, \tag{3.13}$$

which includes the vehicle ID I_i, the current position P_i, the utility type u_i^T, the total amount of data D_i^S to transmit, and the amount of remaining data D_i^R.

For each time slot, we only need to determine which vehicle is allocated to transmit. Thus the actions is defined as:

$$A_t = a,\ a \in [1, 2, \ldots, N_v]. \tag{3.14}$$

If the action $A_t = i$, then the i-th vehicle is allocated to transmit the video data during the current slot.

For the i-th vehicle, we can calculate the current relative distance d_i to the RSU according to its current position P_i. The transmission data rate w_i can be estimated accordingly. Then the total amount of data that can be transmitted in the current time slot is obtained by:

$$\Delta D = w_i \Delta t. \tag{3.15}$$

With the above amount of data transmitted in the current time slot, the amount of remaining data in the next time slot D_i^{R+} is updated by:

$$D_i^{R+} = D_i^R - \Delta D. \tag{3.16}$$

For the other vehicles, without the chance to transmit, their remaining amounts of the video do not change.

Since all the vehicles will move with distance $v\Delta t$, the information for each vehicle in the next time slot V_i^+ needs to be updated accordingly. Only the vehicle that is allocated to transmit in the current time slot has to update the remaining amount of video. For the next time slot, the next state S_t^+ can be represented by:

$$S_t^+ = (V_1^+, V_2^+, \ldots, V_{N_v^+}^+). \tag{3.17}$$

To evaluate how good the decision is, we define the reward R_t for each action at each time slot t. Since the goal of our schedule algorithm is to achieve the maximum utility, the utility achieved by the current N_v vehicles in the current time slot will be

used as the reward. Therefore, the reward can be obtained by:

$$R_t = \sum_{i=1}^{N_v} [F_u(D_i^S - D_i^{R+}) - F_u(D_i^S - D_i^R)]. \tag{3.18}$$

The total reward achieved by all the time slots is the summation of the reward of each time slot. Then the problem can be formulated as the maximization of the long term reward during the whole running time:

$$\begin{aligned} \textbf{max} \ & \sum_{t=1}^{T} R_t, \\ \textbf{s.t.} \ & A_t \in [1, N_v], t \in [1, T]. \end{aligned} \tag{3.19}$$

Note that, for each time slot, if we always take the action which can bring the highest reward for the current slot, such a greedy approach cannot guarantee the long-term highest utility. For instance, as shown in Fig. 3.3, within one time slot, the vehicle V_s with SVC video will achieve 0 utility as the amount of transmitted data cannot even decode the base layer, while another vehicle V_c with CS video can always achieve certain amount of utility. With higher utility to achieve, V_c will be allocated to transmit. However, if V_s can transmit for three consecutive time slots, the reward it can bring will be higher than that of V_c. So, the greedy algorithm which results in immediate optimum cannot guarantee the long-term optimum.

The global optimal solution of the above optimization problem can be solved by searching all the states. Obviously, the computational complexity is still $O({N_v}^T)$, which is prohibitively high. Meanwhile, since the prior knowledge of future arrival is required, such a solution cannot be obtained in practice and it can only be considered as a benchmark offline algorithm and the performance upper bound.

3.4.2 Max Utility Potential Algorithm

Considering the high mobility and dynamic arrival of the vehicles, it is impractical to have future vehicle arrival information. Thus, an online heuristic algorithm should be devised without the prior knowledge of future arrival. In this section, based on the current vehicles' information, we design a simple yet effective algorithm to achieve high utility.

As discussed before, the immediate optimum of single slot cannot guarantee the

long-term optimal results, and more consecutive time slots should be considered to improve the results. Therefore, for each vehicle, all the time slots before it leaves the RSU's transmission range can be considered to check the maximum potential of the achieved utility. Meanwhile, since the future vehicle arrival is unavailable and quite difficult to predict, only the vehicles currently in the RSU's coverage will be taken into consideration.

For each vehicle, with the position P_i and the moving velocity v_i, the RSU can accurately predict how long the vehicle will stay in the transmission range. Then the total number of time slots that the vehicle will stay in the RSU's transmission range can be calculated:

$$n_s = \frac{2R - P_i}{v_i \Delta t}. \tag{3.20}$$

The maximum possible utility achieved for each vehicle can be estimated if the vehicle can always be allocated as long as it is inside the RSU's transmission range. Then the average utility potential in the next n slots U_n is defined as the total amount of utility obtained during the next n slots over n, which is computed as:

$$U_n = \frac{F_u\left(\sum_{i=1}^{n} w_i \Delta t\right)}{n}. \tag{3.21}$$

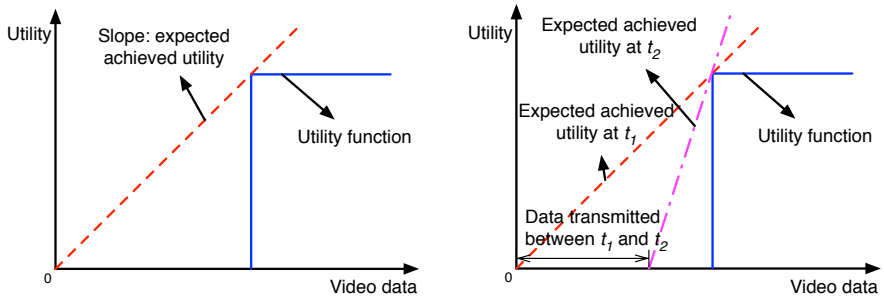
If there is only a small amount of data left to be transmitted, only a fewer number of slots are needed. So if too many future time slots are allocated to the vehicle, many will be wasted. In this case, as the total amount of utility achieved in the future will not change, with more time slots, the average utility potential will be decreased. Besides, for certain types of utility function, with more data downloaded, the increasing rate of achieved utility will slow down. Therefore, the maximum utility potential $\max(U_1, U_2, \ldots, U_{n_s})$ should be computed for the scheduling, which is shown in lines 6 to 12 of Algorithm 2.

The computation of utility potential can be considered as to find the vehicle which can bring the highest utility-to-throughput ratio in the following couple of slots till it leaves the RSU coverage. For a discrete utility type, the computation is to find the steepest stair, while for a continuous utility function, it is close to find the maximum slope.

With the utility potential computed, all the vehicles in the RSU's transmission range will be sorted according to the utility potential. The vehicle with the highest utility potential will be selected to transmit during the current time slot.

Algorithm 2 Max Utility Potential Computation Algorithm

Input: the vehicle's moving velocity v and current position p.
Output: the maximum utility potential of the vehicle
procedure ComputeMaxUtility(v, p)
    Initialize $max\_utility \leftarrow -1$
    Compute the total number of slots it will stay in the RSU's transmission range: $n_s \leftarrow (2R - P_i)/(v_i)$
    Initialize the total amount of downloaded data $D \leftarrow 0$
    for $i = 1$ to n_s **do**
        $D \leftarrow D + w_i \Delta t$
        $utility \leftarrow F_u(D)/i$
        if $utility > max\_utility$ **then**
            $max\_utility \leftarrow utility$
        end if
    end for
    return $max\_utility$
end procedure



(a) Utility Potential at time t_0 (b) Utility Potential at time t_1

Figure 3.4: Illustration of potential achieved utility.

3.4.3 Discussion

From Algorithm 2, it can be noticed that the calculation of utility potential for each vehicle will consider at most N_T time slots. According to the traffic model, given the vehicle density k, the maximum number of vehicles N_v currently in the RSU's transmission range can be estimated by (3.3). Therefore, for each time slot, there will be at most N_v vehicles to be considered for the utility potential. Then the worst-case computation complexity can be derived as $O(N_v N_T)$, and thus the proposed algorithm is simple enough to run in real time.

The estimation of the utility potential can be considered as the slope between the amount of transmitted data and the utility, which is shown in Fig. 3.4(a). Although the proposed algorithm works in a myopic way, it still can achieve good performance without taking the future prediction into consideration. Supposing that one vehicle has been selected to transmit, the future vehicle arrival can be categorized in two scenarios:

- In the next several future time slots, the utility potential of incoming vehicles cannot exceed the current selected vehicle. In this case, the proposed algorithm definitely makes the best decision.

- The future arrival vehicle will achieve a higher utility potential. If shortly after the current slot, there is one incoming vehicle with a higher utility potential, then our algorithm will select the new vehicle to transmit. Within a short time period, the data transmitted by the previous vehicle is very limited, and thus any possible throughput wastage is limited. For a vehicle with stair-case type of utility functions (e.g., for SVC or H.264 video), once it is served, the

utility potential for the selected vehicle will become higher, which is shown in Fig. 3.4(b). With more amount of video data transmitted, the slope also becomes larger for flows with continuous utility functions. Therefore, there is a very small probability that the utility potential of the future arrival vehicles can exceed the current selected vehicle, so we can largely avoid the situation of wasting throughput.

The performance of the proposed utility potential based approach is affected by the utility function. If the vehicle with a stair-case utility function is assigned to transmit for a few time slots, the transmitted video data may not be able to be decoded, therefore 0 utility is achieved, and the throughput is wasted. While for the continuous type utility function, even if the vehicle is only allocated to transmit for one slot, the limited throughput can still achieve certain utility. Therefore, if there are more vehicles with continuous utility functions, the total achieved utility will be closer to the optimal result.

3.5 Performance Evaluation

3.5.1 Simulation Setup

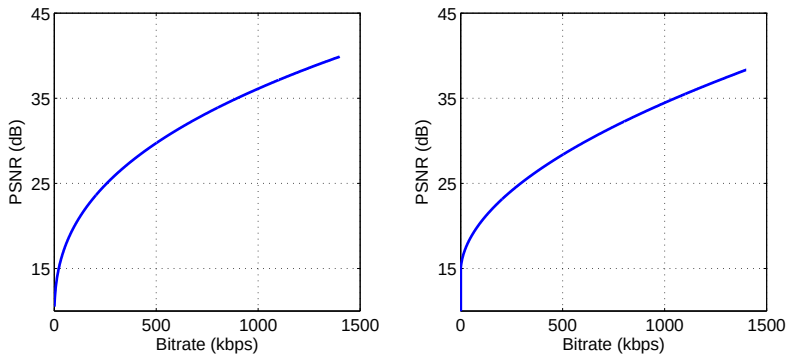


Figure 3.5: Bitrate and PSNR of Compressed sensing video.

To evaluate the performance of the proposed scheduling algorithm, we implemented simulation with NS-3 and conducted simulations with real video traces. Two video traces *Foreman* and *Football* [5], which have 300 and 260 frames respectively, are used in our work. The videos are encoded with H.264, SVC and CS techniques,

Table 3.1: Video Encoding Configurations

Video	Codec	Avg. Bitrate	Avg. PSNR
Foreman	H.264	726.8 kbps	39.5 dB
	SVC layer 1	98.6 kbps	29.2 dB
	SVC layer 2	158 kbps	32.89 dB
	SVC layer 3	374 kbps	36.8 dB
	CS	Avg. PSNR $\approx 2 \times \text{Bitrate}^{0.38} + 8.51$	
Football	H.264	603.31 kbps	33.75 dB
	SVC layer 1	537.9 kbps	29.16 dB
	SVC layer 2	958.2 kbps	33.27 dB
	SVC layer 3	1,408.7 kbps	35.53 dB
	CS	Avg. PSNR $\approx 0.51 \times \text{Bitrate}^{0.53} + 14.62$	

Table 3.2: Simulation Setup

Parameter	Value
RSU transmission range	300 m
Traffic jam density k_{jam}	120 veh/km/lane
Number of lanes	2
Free-way speed v_f	160 km/h

and the encoding configurations are listed in Table 3.1. Other types of encoding techniques and utility functions can also be supported by our algorithm.

The settings of simulation parameters are shown in Table 3.2. Based on the traffic flow model, the free-way speed is always impossible to achieve. For different simulation scenarios, the actual vehicle speed is set between 60 km/h to 130 km/h. With the assumption that the wireless data rate is mainly determined by the relative distance, the wireless network achievable data rate is estimated according to [15, 32]. The wireless configuration is shown in Table 3.3.

To better understand the performance of the proposed algorithm, we compare the results with the D*S/R algorithm proposed in [89], and two other simple schemes, the greedy algorithm which always allocates the wireless resources to the vehicle with the highest data rate, and the first come first service (FCFS) scheme which serves the vehicle that will leave the RSU's coverage first. We run the simulation with several different scenarios. For each scenario, we conduct 100 runs with different random seeds to obtain the average, and each run will last for 10 to 10,000 time slots.

Table 3.3: Wireless Network Setup

Distance range (m)	Data Rate (Mbps)
[0,85]	3
[85,135]	6
[135,215]	12
[215,300]	24

3.5.2 Evaluation Metrics

In order to maximize the total achieved utility, the following metrics are used for evaluation purpose,

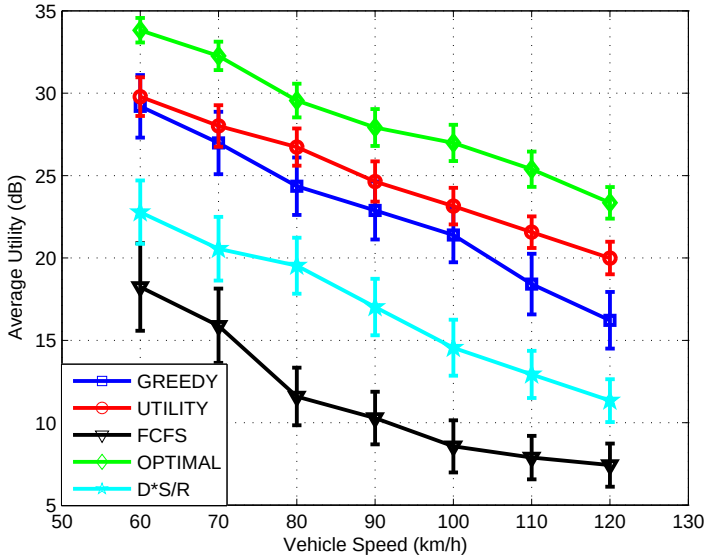
- **Average Utility**: defined as the average utility achieved by all the vehicles during the total time slots.
- **Wasted Throughput Ratio**: defined as the amount of throughput which does not contribute to any utility over the total throughput.
- **Scheduling Fairness Index**: calculated based on the achieved utilities from all the vehicles to evaluate the scheduling fairness.

3.5.3 Simulation Results

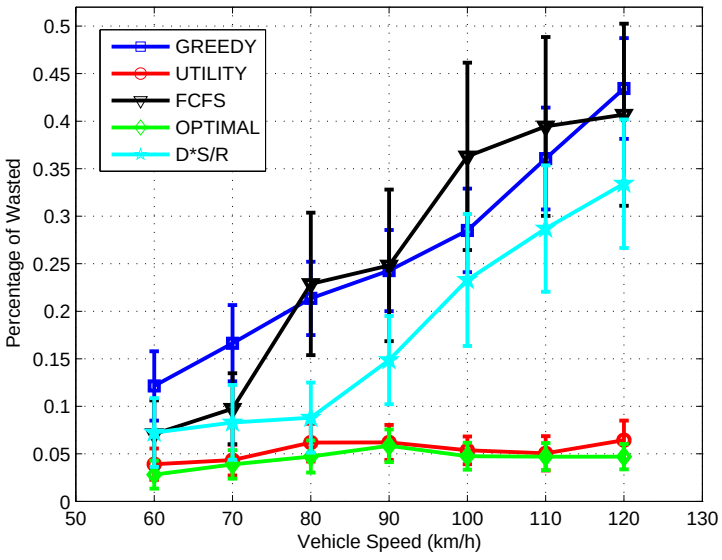
First, we evaluate the performance gap between solutions from the proposed algorithm to the optimal solution. As mentioned before, the computation complexity is very high for the optimal solution. To reduce the computation load, we set the total time slots $T = 10$, and the duration of each time slot is $\Delta t = 1$ second with a very low vehicle density (5 vehicles per kilometer per lane). With the above setting, each run of optimal solution can be solved in around 10 minutes.

The results are shown in Fig. 3.6. The average utility achieved versus the vehicle speed is plotted with 95% confidence interval in Fig. 3.6(a). Obviously, the optimal solution achieves the best performance. As the optimal solution always searches all possible allocations to find the best solution, the performance affected by the randomness is limited. Although the proposed algorithm achieves less average utility than the optimal solution, small variances can be achieved as well. The greedy algorithm always chooses the vehicle with the highest data rate to transmit, which brings high utility. In the implementation, the deadline of D*S/R algorithm for each vehicle is considered as the time it leaves the transmission range. Thus, the vehicle leaving the transmission range sooner will be given a higher priority to transmit. Meanwhile, the low data rate caused by relatively long distance to the RSU leads to low average utility. Similar to the D*S/R algorithm, FCFS always selects the vehicle closest to the transmission edge to transmit, and the worst performance is achieved, compared to the other solutions.

For all the algorithms, the average achieved utility decreases with the increment of the vehicle speed. This is because when the vehicle speed is faster, the sojourn time



(a) Average achieved utility.



(b) Percentage of wasted throughput.

Figure 3.6: Results of Case 1, with vehicle density $k = 5$ vehicles per kilometer per lane, and $T = 10$ time slots.

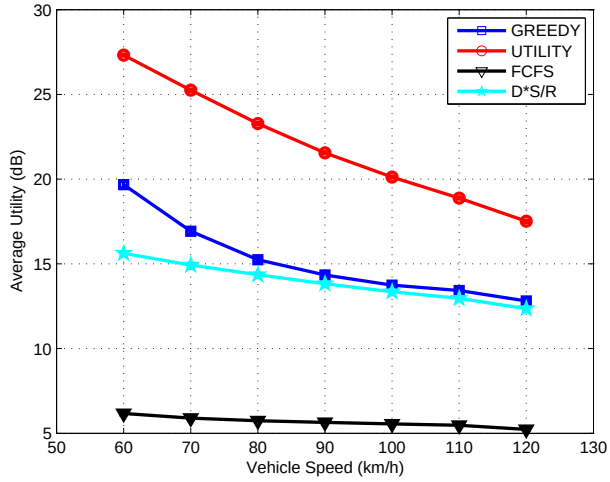
is shorter with the fixed transmission range of RSU. Therefore, the total throughput for each vehicle will be be reduced, which leads to a smaller average utility.

The result of the percentage of the wasted throughput is plotted in Fig. 3.6(b) with 95% confidence interval as well. From the results, we can find that the proposed algorithm can reach similar low bandwidth wastage (around 3%) and small variances to the optimal solution. As the proposed algorithm always tries to find the vehicle with the highest potential to transmit, the throughput will not be wasted much. In this scenario, since we only run the simulation for 10 slots, the data requested during the last several slots may be wasted, as a significant percentage of vehicles are still within the coverage of RSU when the simulation was ended. For the other three algorithms, the wasted throughput is much higher, and the percentage of wastage increases linearly with the increment of vehicle speed. With a higher speed, since the sojourn time becomes less, the probability that a vehicle cannot finish the transmission becomes higher which leads to a higher percentage of wasted throughput.

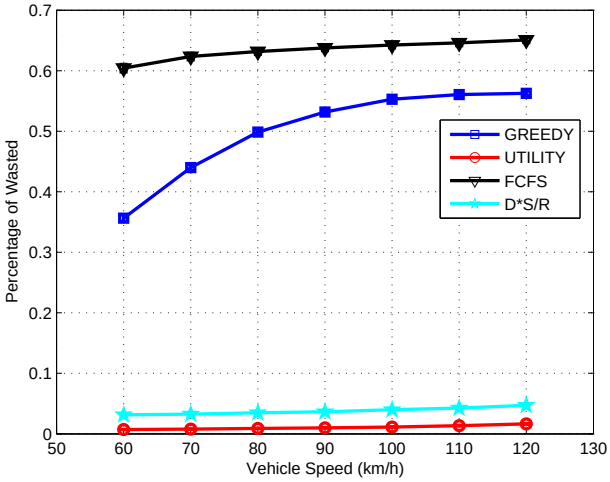
Next, we conduct the simulations with more time slots. In this case, the time slot duration is set to $\Delta t = 0.1$s to ensure that there is at most one vehicle arrival during each time slot. The total time slots is set to $T = 10,000$. The optimal solution cannot be obtained for comparison as the total amount of time slots is too large, such that the computation load has already exceeded our computation capacity. In order to evaluate the performance under different vehicle densities, which are set to 10 and 30 vehicles per kilometer per lane respectively. For each scenario, all vehicles run with same average speed with different variations. The results are plotted in Figs. 3.7 and 3.8 with 95% confidence interval.

Figures 3.7(a) and 3.8(a) show the results of the average achieved utility. From the figures, it can be noticed that performance variances for all the algorithms have been greatly reduced. According to our simulation, when the simulation runs for a long time, the performance of each algorithm will finally converge. Compared to the previous case, the results show the similar trend. Less average utility can be obtained due to the higher traffic. With more vehicles, there will be more competitions for the limited wireless resources. As a result, the average throughput for each vehicle will be reduced and will lead to less average utility. By considering the possible highest utility potential, our utility maximization based algorithm can outperform the other algorithms.

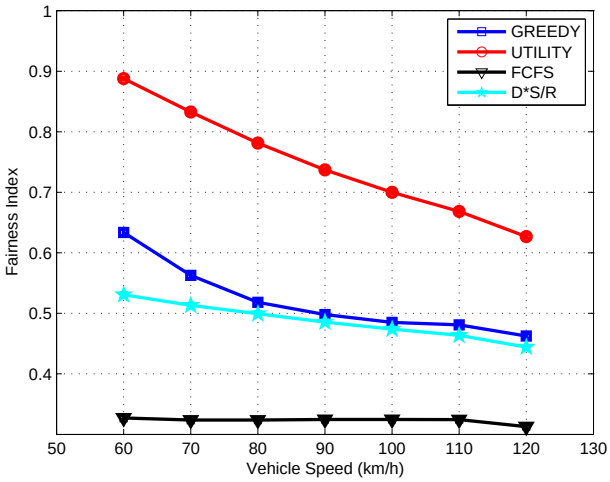
In Figs. 3.7(b) and 3.8(b), the results of the percentage of wasted throughput are shown. With the long running time, the proposed algorithm wastes very limited



(a) Average achieved utility.

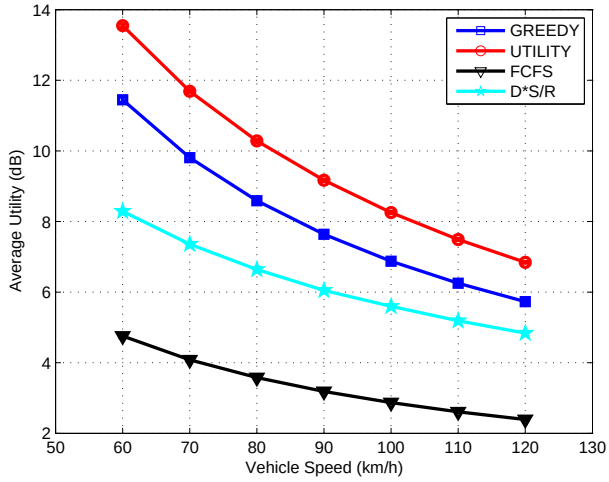


(b) Percentage of wasted throughput.

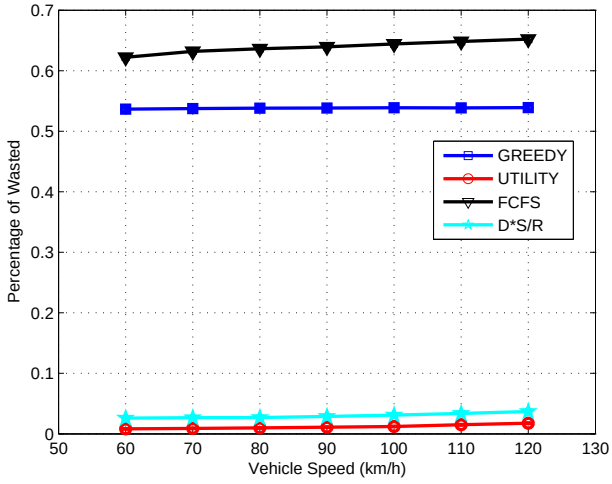


(c) Fairness index.

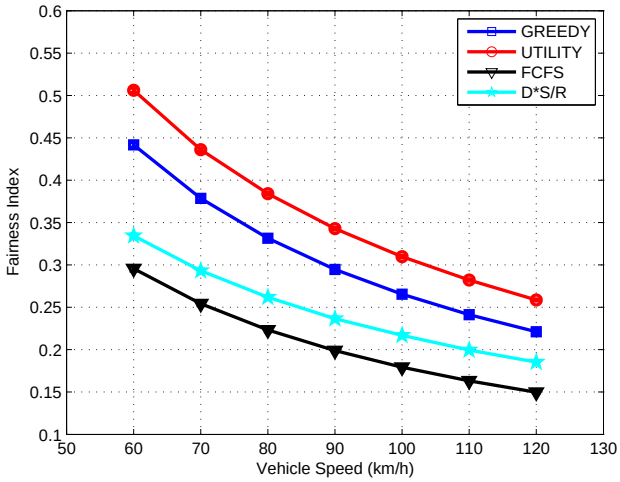
Figure 3.7: Results of Case 2, with vehicle density $k = 10$ vehicles per kilometer per lane, and $T = 10,000$ time slots.



(a) Average achieved utility.

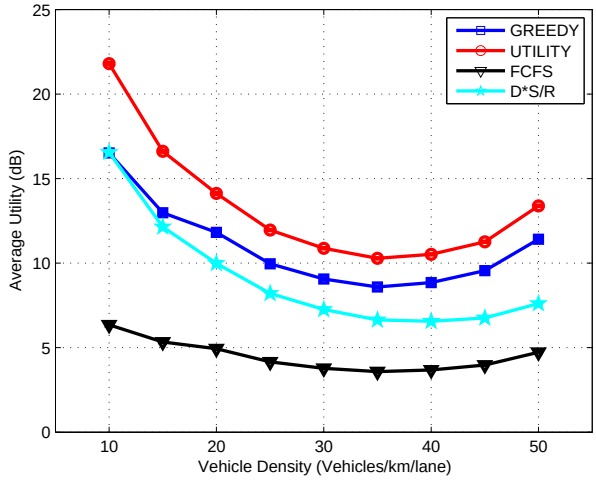


(b) Percentage of wasted throughput.

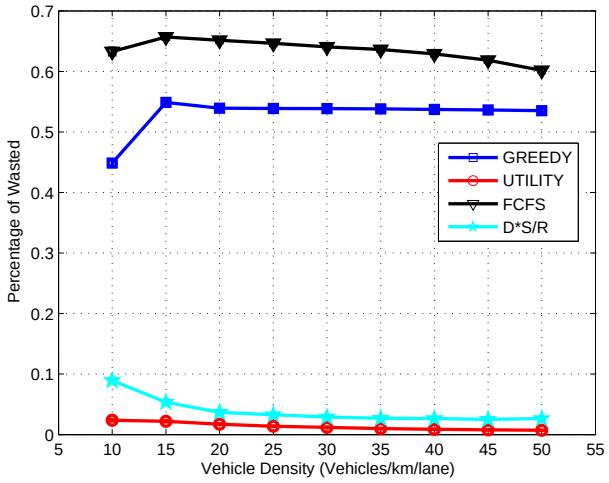


(c) Fairness index.

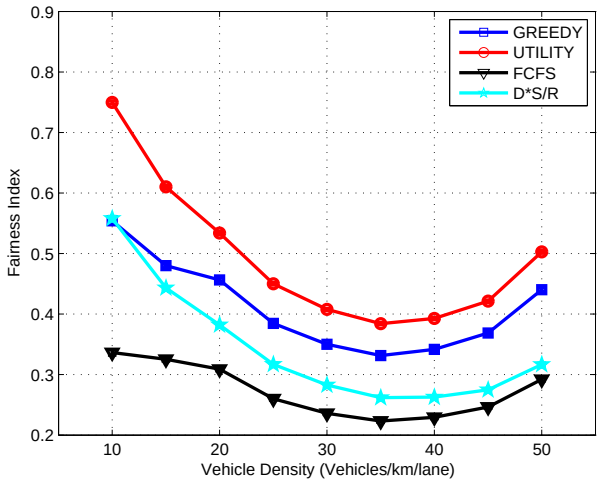
Figure 3.8: Results of Case 3, with vehicle density $k = 30$ vehicles per kilometer per lane, and $T = 10,000$ time slots



(a) Average achieved utility.



(b) Percentage of wasted throughput.



(c) Fairness index.

Figure 3.9: Results of Case 4, with $T = 10,000$ time slots.

amount of throughput. Theoretically, the proposed algorithm only let the vehicles which can gain utility to transmit, so none of the throughput should be wasted. However, with the stair-type utility function, the throughput may be wasted. For instance, as shown in Fig. 3.3, during t_3, the residual base layer data and part of enhancement layer data will be transmitted simultaneously. Since the transmitted enhancement layer is not enough to decode, the bandwidth to transmit the enhancement layer data is wasted. Meanwhile, the performance does not change much with different vehicle speed. This can be explained that the estimation of the potential achieved utility is independent of the vehicle speed, thus it does not affect the performance.

The results of the fairness index are shown in Figs. 3.7(c) and 3.8(c). It can be noticed that the trend of the fairness index is similar to the average throughput. With the increment of vehicle speed, the average utility achieved will decrease. The lower average utility implies that more vehicles cannot achieve any utility, which leads to a lower fairness index. Therefore, as the proposed algorithm can achieve a higher average utility, the schedule fairness can be improved at the same time.

Finally, we conducted the simulations with the change of vehicle density. We still set the time slot duration to 0.1s and run the simulation with 10,000 time slots. According to the traffic model, the vehicle speed can be estimated by (3.1). The results are plotted with 95% confidence interval and shown in Fig. 3.9.

Figure 3.9(a) shows the average achieved utility. Different from the previous results, when the vehicle density and the vehicle speed change simultaneously, the performance results do not change monotonically. With a low vehicle density, the low competition will bring relative high throughput. Thus a higher average utility will be achieved correspondingly. However, when the vehicle density reaches a certain level, according to (3.1), the vehicle speed will be decreased a lot. Therefore, each vehicle will have a longer sojourn time to improve the throughput.

From Fig. 3.9(b), we can still achieve low throughput wastage with the proposed algorithm. As mentioned before, the estimation of the utility potential is independent with the traffic arrival rate or density. Therefore, a low throughput wastage can always be guaranteed. D*S/R takes both the deadline and throughput into consideration, which can reduce the throughput wastage as well. The greedy and FCFS did not consider whether the throughput can bring certain utility, thus more than half of the throughput was wasted.

At last, we show the fairness index in Figure 3.9(c). As explained before, the fairness index has a similar trend to the average achieved utility. With the higher

average utility achieved, the proposed algorithm can outperform the other algorithms in terms of better fairness.

3.6 Conclusions

In this chapter, the problem of multimedia transmission scheduling among multiple vehicles over the Drive-thru Internet has been investigated. The utility model was devised to map the throughput to the user's satisfaction level, such as PSNR of video. First, the scheduling problem was formulated as an optimization problem to maximize the total achieved utility. Since the optimization problem is NP-hard, the problem was then converted to a finite state decision problem and solved by a searching algorithm which is served as the performance upper bound. A heuristic algorithm based on the utility potential was proposed to obtain the solution in real time with high performance. Finally, we implemented and conducted extensive simulations to evaluate the performance.

Chapter 4

Utility Maximization for Multimedia Data Dissemination in Large-scale VANETs

In the previous chapter, we have discussed the scheduling of the multimedia transmission by V2I communications in Drive-thru Internet. For the large-scale VANETs, considering the high competition from huge number of vehicles of the V2I communication bandwidth and the expensive cost of the infrastructure construction, V2V communications plays an important role for data dissemination algorithms. In this chapter, we investigate the V2V based multimedia dissemination problem in large-scale VANETs. We first utilize a hybrid-network framework to model the VANETs to address the mobility and scalability issues. Then, we formulate a utility-based maximization problem to find the best delivery strategy and select an optimal path for the multimedia data dissemination, where the utility function has taken the delivery delay, the Quality of Services (QoS) and the storage cost into consideration. With rigorous analysis, we obtain the closed-form of the utility function, and then obtain the optimal solution of the problem with the convex optimization theory. Finally, we conducted extensive trace-driven simulations to evaluate the performance of the proposed algorithm with real traces collected by taxis in Shanghai.

4.1 Introduction

The emerging of the auto operating systems (OS), such as Google Auto Link and Apple Carplay, have made a significant step to support more applications in Vehicular Ad-Hoc Networks (VANETs). Besides the safety applications, there is an increasing demand to provide multimedia applications, such as media-rich entertainment and location-aware applications in VANETs [10, 58, 59, 61] for the smart city scenario. These multimedia applications depend on efficient and reliable multimedia data dissemination in VANETs. However, due to the randomness and high mobility of the vehicles (e.g., taxis), it is difficult to predict the movement of the vehicles. Meanwhile, since the contact time between vehicles is usually limited, the multimedia data may not be fully transmitted between two contacting vehicles by vehicle-to-vehicle (V2V) communication only, which brings a new challenge for data dissemination. How to support efficient and reliable multimedia data dissemination in the large-scale VANETs is an interesting and challenging problem, which is the mainly concerned in this chapter.

There are lots of work considering epidemic routing [11, 62], density based routing [70] or prediction based routing [84] to solve data dissemination problems in VANETs. When the data can be transmitted successfully in a short communication time between vehicles, they can overcome the high mobility and the inherent intermittent connectivity problems of VANETs. However, due to the flooding nature of these routing schemes, they may not be suitable for multimedia data dissemination in large-scale VANETs. First, considering the huge number of vehicles in the urban large-scale VANETs, it is impractical to let each vehicle to maintain a list of pair-wise contact probability and pattern, and the flooding algorithms cannot scale well. Second, because of the relatively large size of the multimedia data, it is hard to replicate multiple copies and forward them to the unexpected connections, i.e., dynamic and limited contact pattern and time. Third, within the short contact time, it is difficult to make a good data forwarding and routing strategy. Although the third-/fourth-generation cellular networks can provide ubiquitous and reliable data communication, the high cost is a major concern, especially for bulk multimedia data.

To address the above issues, in this chapter, we devise a low cost yet efficient and reliable multimedia data dissemination algorithm in large-scale VANETs. Recently, several works [30, 31, 73] introduce a scalable hybrid network framework, combining the vehicle-to-infrastructure (V2I) and V2V communications, for data dissemination

in large-scale VANETs. This framework provides a topology modeling method of a VANETs and introduce Road Side Units (RSU) and drop box to help for dissemination. We adopt the same framework for topology modeling in this work. But RSUs collect the vehicle's mobility information and help the vehicle with the multimedia data to find other passing vehicles to carry the data or wait for an appropriate future arrival vehicles to deliver the multimedia data storing in drop box. More importantly, in addition to the delivery delay considered in [30], we consider the Quality of Service (QoS) metrics of the multimedia and the storage cost of drop box, which are important and realistic for multimedia dissemination. Then, we formulate the utility-based maximization problem to find the optimal path and delivery strategy.

The main contributions of this chapter are three-fold. First, we investigate a novel multimedia data dissemination for large-scale VANETs. Since the multimedia data may not be transmitted completely during two vehicle contact with each other, we use the hybrid framework proposed in [30] for VANETs' topology modeling and formulate utility-based maximization problem to solve the dissemination problem. Second, based on the rigorous theoretical analysis, we obtain the closed-form of the expected utility, including the delivery delay, QoS of the multimedia data and the storage cost. We also use the convex optimization theory to obtain the optimal solution. Finally, we implement and conduct trace-driven simulations to evaluate the performance of the proposed algorithm. The simulation results demonstrate the rigorousness of the theoretical results and the high performance of the proposed algorithm.

4.2 Related Work

There are extensive work studying how to devise efficient and reliable routing scheme for traditional delay-tolerant networks (DTN) or large-scale VANETs [6, 11, 24, 43, 47, 62, 70, 84–87]. In [87], based on the analysis of the real bus traces, the authors proposed a route-level model with finer granularity to better predict the inter-contact pattern between buses. To reduce the energy waste of the flooding based routing, Spyropoulos et al. [62] proposed the spray routing algorithm which tried to apply the single-copy message routing technique to the controlled number of multiple-copy case. Considering the dynamic and unexpected connections, [85] tried to apply the random linear network coding to the epidemic routing algorithms. In this way, the delivery ratio are improved as not all packets must be successfully transmitted. Zhang et al. [86] proposed a new clustering-based geocast dissemination algorithm with both

taxi and bus for urban VANETs. By dividing the whole VANETs area into small regions, the mobility pattern can be predicted more accurately. Luan et al. in [47] devised a practical and low cost infrastructure for large-scale VANETs. Distributed RSUs are deployed around the whole city without a central controller.

Recently, great efforts have been devoted for multimedia data dissemination [10, 12, 13, 23, 37, 41, 42, 58, 59, 61, 68, 69, 72]. Relying on the V2V communications, Soldo et al. proposed a fully distributed live video broadcast algorithms in [61] to achieve high satisfaction and fairness. In [12], due to the limited contact time, the authors considered how to determine the piece data and size to make a trade-off between the data size and the piece overhead. They devised a popularity based piece selection algorithm to solve the problem. To cope with the dynamic channel of VANETs, [13] presented a vehicle to infrastructure (V2I) based video transmission system with scalable video coding. In [58, 59], Rezende et al. proposed receiver based density-aware video dissemination algorithm. The estimated location based relay selection process was decoupled from the video transmission process. To reduce the forwarding delay of real-time video, De Felice et al. proposed a distributed beaconless routing protocol in [23]. V2V based high quality routes were maintained as the backbone for fast video delivery. For these existing works, they failed to take the delivery delay or the QoS metrics for the multimedia data into consideration, which are the main concerns in this work.

4.3 System Models and Problem Formulation

In this section, we first briefly describe the system scenario and models, followed by the problem formulation.

4.3.1 Network Scenario

This chapter investigates a multimedia data dissemination problem in VANETs, and the objective is to find a best path and the optimal strategy to disseminate different kinds of multimedia data through vehicles (e.g. taxicabs) with the assistance of RSUs. As shown in Fig. 4.1, multimedia data generated at the source node, and be carried and forwarded by the passing vehicles to the destination region. During the message delivering process, the multimedia data can be forwarded from one vehicle to another vehicle by V2V communications. When one vehicle enters the coverage of a RSU, it

will report its travel plan to the RSU, which is responsible for coordination of V2V communications only. If the vehicle is carrying the multimedia data, then it will try to find the best candidate vehicle with the assistance of RSU to forward. If there is no appropriate vehicle to forward, it will store the multimedia data in drop box temporarily, and then rely on the RSU to find the future arrival vehicle to forward the data.

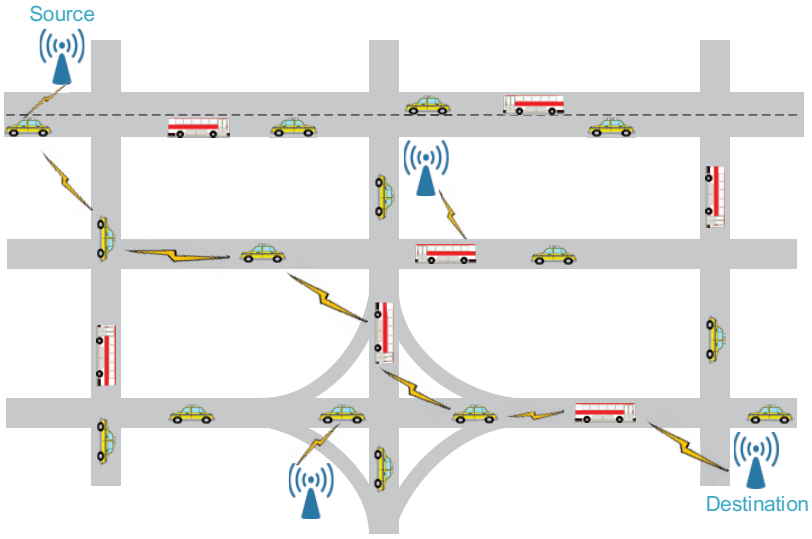


Figure 4.1: Network for Multimedia Data Dissemination in VANET.

Similar to [47], the RSUs are deployed at different locations around the whole city without the central controller. The whole city is divided into regions. For each region, one RSU is deployed and the coverage area of the RSU is defined as a hot-spot. A unique identification will be allocated to each RSU. RSUs may or may not connect with each other, but the global RSU distribution information are available for all RSUs. In addition, a drop box will be installed at each hot-spot as well for temporary storage of the multimedia data when the tagged vehicle cannot find an appropriate passing vehicle to forward. The drop box can be standalone storage devices without Internet connectivity, or the storage device of the parking vehicles in the hot-spot. With the large number of parking vehicles in the hot-spot, their storage and communication capacities are large enough to support heavy load. To better describe the the problem in this work, we abstract the whole city into a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the set of vertexes and edges, respectively. Each vertex $\mathcal{V}_i$ can be considered as the RSU at the i-th hot-spot, and each edge $\mathcal{E}_{ij} =< i, j >$ denotes the direct link from i to j, which is viewed as the traffic links between hot-spots.

4.3.2 Vehicle Mobility Model

The vehicles with a random mobility pattern, e.g., taxis, are considered as the data forwarders. With the analysis of real traffic trace collected by taxis in Shanghai (partially available at http://www.cse.ust.hk/scrg), it is found that the vehicles from any hot-spot i to j is a Poisson process [30, 86]. Hence, the arrival rate of vehicles from hot-spot i to j is assumed to follow a Poisson distribution with the parameter λ_{ij} using to denote the vehicle arrival rate.

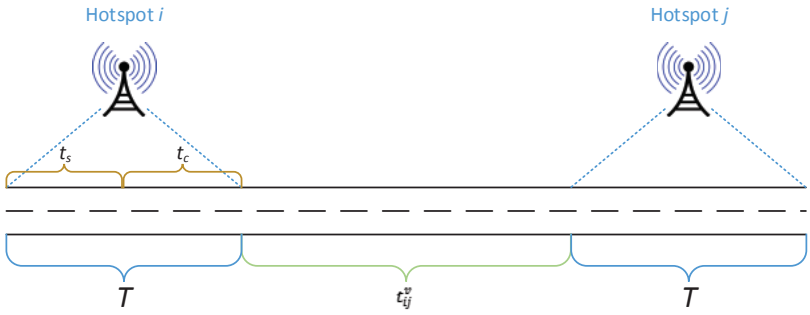


Figure 4.2: Time definition.

As shown in Fig. 4.2, let t_{ij}^v represents the travel time from hot-spot i to j, where the travel time inside the RSU's coverage is not included. Assume that the coverage of each RSU is identical, and given the maximum traveling speed within the coverage of the RSUs, the minimum time that each vehicle in the RSU's coverage is fixed, which is defined as the channel time T. Since the data transmission rate is limited in VANETs' communication and the size of multimedia data is relatively large, the transmission time of the multimedia data plays an important role. In our scenario, the channel time T is divided into two parts: searching time t_s (the time for the vehicle who is carrying the multimedia data to find a vehicle to forward in the next link) and transmission time t_c (the time for the vehicle who is carrying multimedia data to transmit the data to drop box for temporary storage), and $t_s + t_c = T$. Specifically, when the vehicle carrying the multimedia data enters the RSU's coverage, it will search a passing vehicle to forward during searching time t_s with the assistant of RSU. If the tagged vehicle cannot find any vehicle to forward within t_s, then it will forward the multimedia data to the RSU during the remaining time t_c. It should be pointed out that the transmission time t_c determines the minimum amount of multimedia data can be transmitted.

4.3.3 Utility Model

Note that in data dissemination, fast delivery is preferable. Therefore, we define a utility function of the delivery delay to depict the benefit of fast dissemination, which is given by

$$U_D = F_d(T_{sd}), \tag{4.1}$$

where F_d is the delay utility function and T_{sd} is random variable denotes the total delivery delay. F_d is assumed to be a monotonic decreasing function of the delivery delay, *i.e.*, a shorter delay will bring a higher utility.

Meanwhile, the multimedia data generated at the source node will be encoded first, e.g., videos may be encoded with H.264, Scalable Video Coding (SVC) or Compressed Sensing (CS) based codec. However, with different encoding techniques, the amount of data required to decode certain quality video is different. Also, different delivery strategy may cause different quality of the video successfully forwarded to the destination. Therefore, we employ a utility model to map the total throughput of the multimedia data to the user's satisfaction level, which is similar to [78]. Mathematically, the utility is modeled as

$$U_M = F_u^k(D), k = 1, 2, \ldots, N_e, \tag{4.2}$$

where F_u^k is the utility function for the k-th type of encoding technique, D is the total throughput which is a random variable depending on the forwarding strategies, and N_e is the total number of encoding techniques.

In addition, if the tagged vehicle cannot find any passing vehicle to forward the multimedia data, it can only request drop box for temporary storage, which will incur some additional storage cost. Thus, we define the storage cost utility function as follows:

$$U_C = F_c(N_{sd}), N_{sd} \geq 0, \tag{4.3}$$

where F_c is the utility function for the cost, and N_{sd} is the total time of additional storage requests along the path.

By combining all the above utilities together, we defined the total utility function as

$$U = \alpha U_D + \beta U_M - \gamma U_C, \tag{4.4}$$

where α, β and γ are the weight parameters which can be adjusted according to user's

preference.

4.3.4 Problem Formulation

Assume that there are N paths from source s to destination d, and let $\Omega = \{p_{sd}^k | k = 1, 2, \ldots, N\}$ be the set of these paths. Define the delivery strategy set for the nodes by

$$\Theta := \{s(t) | t_c = t, 0 \leq t \leq T\}.$$

Note that a larger t_c can guarantee that the minimum amount of multimedia data can be transmitted is higher, i.e., a higher throughout, but may cause higher delivery delay and storage cost since the probability of the data transmitted to the RSU is increased. This is a trade-off between throughput, delay and cost. Therefore, we formulate a utility maximization problem as follows.

$$\max_{p_{sd}^k \in \Omega} \max_{s(t) \in \Theta} \; U = \alpha U_M + \beta U_D - \gamma U_C, \tag{4.5}$$

where U_D, U_M and U_C are defined respectively in (4.1), (4.2) and (4.3). Here the utility functions in the objective function are assumed to be convex and differentiable functions. Since the variables in the utility functions are random variables and thus the above problem is a random optimization problem. We further simplify this problem to maximize the expected utility as

$$\max_{p_{sd}^k \in \Omega} \max_{s(t) \in \Theta} \; E\{U\} = \alpha E\{U_M\} + \beta E\{U_D\} - \gamma E\{U_C\}. \tag{4.6}$$

The above maximization problem is still a convex optimization problem. To solve this problem, the main challenging is how to obtain the closed-form of the utility functions in the objective function. After obtained that, we can use the general convex optimization approach to obtain the optimal solution.

4.4 Optimal Delivery Strategy

To find the optimal path, according to our utility model, three parts need to be considered, the total delay, the multimedia quality and the storage cost. In this section, we obtain the closed form of the total utility for any given path at first, and solve the stationary point of the utility to obtain the optimal strategy. Then,

we compare the utility under optimal strategy among all possible paths to find the optimal one for data forwarding. Finally, we design an algorithm to solve the utility-based maximization problem.

For simplicity, in the following subsections, we consider path p_{sd} (could be anyone path p_{sd}^k in the set Ω) as the data forwarding path, and assume that there are n links along the path, and the links in the path are $< i_0, i_1 >, < i_1, i_2 >, ..., < i_{n-1}, i_n >$, where $i_0 = s, i_n = d$.

4.4.1 Analysis of the Expected Delay

In this subsection, we try to analyze the expected delay of each link. There are two possible value of the link delivery delay. First, consider the case that the vehicle with multimedia data can find passing vehicle to forward in time t_s ($t_s = T - t_c$), i.e., the vehicle will forward the multimedia data directly to the coming vehicle, and no additional delay will be introduced. In this case, the delay equals the time of vehicle passing the RSU plus the traveling time, i.e., $T + t_{ij}^v$. Since the vehicle arrival rate from i to j follows the Poisson distribution, the probability of this case happens is the probability that the vehicle with multimedia data can find another vehicle going to j within t_s to forward. Thus, this probability satisfies

$$Pr\{\tau \leq t\} = \int_0^{T-t_c} \lambda_{ij} e^{-\lambda_{ij}\tau} d\tau = 1 - e^{-\lambda_{ij}(T-t_c)}. \tag{4.7}$$

Secondly, consider the case that the vehicle with multimedia data cannot find an appropriate vehicle for data forwarding within time t_s at node i, the probability of such case is

$$Pr\{\tau > t\} = 1 - \int_0^{T-t_c} \lambda_{ij} e^{-\lambda_{ij}\tau} d\tau = e^{-\lambda_{ij}(T-t_c)}. \tag{4.8}$$

In this case, the multimedia data is transmitted to and stored at drop box temporarily in the next transmission time t_c. After that, the RSU will find a vehicle going from node i to j, and forward the data to that vehicle. We define the waiting time as t_{ij}^w, i.e., the time of RSU finding a vehicle for data delivering from node i to node j. Clearly, the PDF of t_{ij}^w is $\lambda_{ij} e^{-\lambda_{ij}\tau}$. Hence, for this case, the link delivery delay should be equal to $T + t_{ij}^w + T + t_{ij}^v$, and its expected value is calculated by

$$\int_0^{\infty} (2T + t_{ij}^v + \tau) \lambda_{ij} e^{-\lambda_{ij}\tau} d\tau. \tag{4.9}$$

Let $E\{T_{ij}\}$ be the expected delay from hot-spot i to j. Combing the above two cases, the expected delay can be calculated by

$$\begin{aligned} E\{T_{ij}\} &= Pr\{\tau \le t\}(T + t_{ij}^v) \\ &\quad + Pr\{\tau > t\} \int_0^\infty (2T + t_{ij}^v + \tau)\lambda_{ij} e^{-\lambda_{ij}\tau} d\tau \\ &= (1 - e^{-\lambda_{ij}(T-t_c)})(T + t_{ij}^v) \\ &\quad + e^{-\lambda_{ij}(T-t_c)} \left(2T + t_{ij}^v + \frac{1}{\lambda_{ij}}\right) \\ &= T(1 + e^{-\lambda_{ij}(T-t_c)}) + t_{ij}^v + \frac{e^{-\lambda_{ij}(T-t_c)}}{\lambda_{ij}}. \end{aligned} \tag{4.10}$$

Since the Poisson process is time independent, the link delay between different links are independent with each other. Let T_{sd} be the delivery delay with path p_{sd}. It follows from (4.10) that the expectation of T_{sd} satisfies

$$E\{T_{sd}\} = \sum_{i=1}^{n} [T(1 + e^{-\lambda_{ij}(T-t_c)}) + t_{ij}^v + \frac{e^{-\lambda_{ij}(T-t_c)}}{\lambda_{ij}}]. \tag{4.11}$$

With the expected delay, it is easy to obtain the expected delay utility.

4.4.2 Analysis of the Multimedia Utility

It is obvious that the overall amount of the transmitted data is determined by the lowest throughput among all links in the delivering path. In our scenario, based on path p_{sd}, there are two cases of the lowest throughput for data forwarding, which will be analyzed respectively as follows.

When the data is transmitted to the RSU in one or more nodes during the delivering, and the lowest throughput is $t_c R$, then the corresponding multimedia utility should be $U_M = F_u(t_c R)$. Note that, the lowest throughput is achieved if the vehicle carrying the multimedia data cannot find an appropriate vehicle to forward for at least one link along the delivering path. Given path p_{sd}, it is not difficult to obtain the probability of the vehicle carrying data at each node can always find a suitable vehicle for next link forwarding is $\prod_{k=1}^{n} \int_0^{T-t_c} \lambda_{i_{k-1}i_k} e^{-\lambda_{i_{k-1}i_k}\tau} d\tau$, i.e., the probability that the data are transmitted by V2V communication only in all nodes. Hence, the

probability of $U_M = F_u(t_c R)$ satisfies

$$Pr\{U_M = F_u(t_c R)\} = 1 - \prod_{k=1}^{n} \int_0^{T-t_c} \lambda_{i_{k-1} i_k} e^{-\lambda_{i_{k-1} i_k} \tau} d\tau$$
$$= 1 - \prod_{k=1}^{n} \left(1 - e^{-\lambda_{i_{k-1} i_k} (T-t_c)}\right). \tag{4.12}$$

On the other hand, if there is always a candidate forwarding vehicle for all the links along the path during the delivering process, then the total throughput along the path is greater than $t_c R$, and then we have $U_M > F_u(t_c R)$ since F_u is a monotonically increasing function of the throughput. The probability of $U_M > F_u(t_c R)$ can be derived as,

$$Pr\{U_M > F_u(t_c R)\} = \prod_{k=1}^{n} \int_0^{T-t_c} \lambda_{i_{k-1} i_k} e^{-\lambda_{i_{k-1} i_k} \tau} d\tau$$
$$= \prod_{k=1}^{n} \left(1 - e^{-\lambda_{i_{k-1} i_k} (T-t_c)}\right) \tag{4.13}$$

In this case, to calculate the expected value of U_M, it is needed to obtain the PDF of the lowest throughput, which is a challenging problem. Suppose that τ_{ij} is the actual searching time for the vehicle with the multimedia data to find a passing vehicle to forward at hot-spot i to the next hot-spot j, where $< i, j > \in p_{sd}$. Clearly, we have $0 < \tau_{ij} \leq t_s$. Let $x = \max_{i=1}^{n} \tau_{ij}$. Then, we can obtain that the lowest throughput is $(T - x)R$, and $U_M = F_u((T - x)R)$. According to the definition of PDF and the relationship between probability and PDF, we can obtain the PDF of x, i.e., the PDF of $U_M = F_u((T - x)R)$, with the following lemma.

Lemma 4.1. *Given path p_{sd}, the PDF of the search time x is*

$$f(x) = \sum_{k=1}^{n} \left(\lambda_{i_{k-1} i_k} e^{-\lambda_{i_{k-1} i_k} x} \prod_{l=1, l \neq k}^{n} \left(1 - e^{-\lambda_{i_{l-1} i_l} x}\right) \right) \tag{4.14}$$

for $\forall x \in [0, T - t_c)$.

Proof. Based on the definition of PDF, one infers that

$$Pr\{U_M > F_u(t_c R)\} = \int_0^{T-t_c} f(x) dx \tag{4.15}$$

From (4.13), we have

$$\int_0^{T-t_c} f(x)dx = \prod_{k=1}^{n} \left(1 - e^{-\lambda_{i_{k-1}i_k}(T-t_c)}\right) \tag{4.16}$$

Note that t_c is a variable in $[0, T]$, which means that the above equation holds for any given t_c where $0 \leq t_c \leq T$. Let $y = T - t_c$, it follows from (4.16) that

$$\int_0^{y} f(x)dx = \prod_{k=1}^{n} \left(1 - e^{-\lambda_{i_{k-1}i_k}(y)}\right)$$

for $\forall y \in [0, T]$. Taking derivative of both sides of the above equation over y yields

$$\begin{aligned} f(y) &= \frac{d\left(\prod_{k=1}^{n}\left(1 - e^{-\lambda_{i_{k-1}i_k}y}\right)\right)}{d\ y} \\ &= \sum_{k=1}^{n}\left(\lambda_{i_{k-1}i_k} e^{-\lambda_{i_{k-1}i_k}y} \prod_{l=1,l\neq k}^{n}\left(1 - e^{-\lambda_{i_{l-1}i_l}y}\right)\right) \end{aligned} \tag{4.17}$$

for $\forall y \in [0, T]$. Hence, given $t_c \in [0, T]$, we have (4.14) hold. □

In (4.14), $\lambda_{i_{k-1}i_k} e^{-\lambda_{i_{k-1}i_k}x} \prod_{l=1,l\neq k}^{n}\left(1 - e^{-\lambda_{i_{l-1}i_l}x}\right)$ indicates the probability that the searching time is equal to x exactly for link $i_{k-1}i_k$ along path p_{sd}, while for other $n-1$ links, the waiting time is less or equal than x. Since path p_{sd} includes n links, there can be n different cases that the waiting time for link $i_{k-1}i_k$ is exactly x. Hence, the PDF is the summation of all the different cases, which is corresponding to the intuitive results.

Combining the above discussions of the two cases, the $E\{U_M\}$ is calculated by

$$\begin{aligned} E\{U_M\} &= Pr\{U_M = F_u(t_cR)\}F_u(t_cR) \\ &\quad + \int_0^{T-t_c} F_u((T-x)R)f(x)dx \\ &= \left(1 - \prod_{k=1}^{n}\left(1 - e^{-\lambda_{i_{k-1}i_k}(T-t_c)}\right)\right) F_u(t_cR) \\ &\quad + \int_0^{T-t_c} F_u((T-x)R)f(x)dx. \end{aligned} \tag{4.18}$$

4.4.3 Analysis of the Cost Utility

Since the cost considered in this chapter is the storage cost of drop box, when the data is not transmitted to the drop box for temporary storage during the delivering process, the cost is 0. Otherwise, in each node i, if the data was stored in drop box, it will cause a storage cost.

Note that the probability that a vehicle carrying data cannot find a moving vehicle to forward, i.e., $t_s > T - t_c$, is equivalent to that there is no vehicle coming in $T - t_c$ time, and thus the value of this probability is calculated as:

$$Pr\{t_s > T - t_c\} = 1 - \int_{t=0}^{T-t_c} \lambda_{ij} e^{-\lambda_{ij} t} dt = e^{-\lambda_{ij}(T-t_c)}.$$

Hence, we have

$$E\{U_C\} = F_c \left(\sum_{k=1}^{n} e^{-\lambda_{i_{k-1} i_k}(T-t_c)} \right). \tag{4.19}$$

4.4.4 Dissemination Algorithm

From the previous three subsections, we obtain the closed-form expression of how to calculate the expected delay, the expectation of multimedia utility and cost utility. By combing them together, we can obtain the expression of the total expected utility, i.e., the objective function of (4.6), which is given by

$$\begin{aligned}
E\{U\} &= \alpha E\{U_D\} + \beta E\{U_M\} - \gamma E\{U_C\} \\
&= \alpha F_d \left(\sum_{k=1}^{n} \left(T(1 + e^{-\lambda_{ij}(T-t_c)}) + t_{ij}^v + \frac{e^{-\lambda_{ij}(T-t_c)}}{\lambda_{ij}} \right) \right) \\
&\quad + \beta \left(\int_0^{T-t_c} F_u((T-x)R) f(x) dx \right. \\
&\quad \left. + \left[1 - \prod_{k=1}^{n} \left(1 - e^{-\lambda_{i_{k-1} i_k}(T-t_c)} \right) \right] F_u(t_c R) \right) \\
&\quad - \gamma F_c \left(\sum_{k=1}^{n} e^{-\lambda_{i_{k-1} i_k}(T-t_c)} \right).
\end{aligned} \tag{4.20}$$

From (4.20), the transmission time t_c is the only variable that controls the total utility. Therefore, we need to find an optimal transmission time t_c^* such that the total utility can be maximized.

Theorem 4.2. *Suppose that the optimal transmission time is t_c^* which can maximize the total utility. If τ_c is the stationary point of $E\{U\}$, i.e., $\frac{d(E\{U\})}{d\ t_c}\Big|_{t_c=\tau_c} = 0$, then the optimal searching time t_c^* satisfies*

$$t_c^* = arg \max_{t_c=0,\tau_c,T} E\{U\},\ when\ \tau_c \in [0,T], \tag{4.21}$$

or

$$t_c^* = arg \max_{t_c=0,T} E\{U\},\ when\ \tau_c \notin [0,T]. \tag{4.22}$$

Proof. Since all the utility function is assumed to be convex function and τ_c is the stationary point, the value of $E\{U\}|_{t_c=\tau_c}$ must be an extreme point and be either a maximum or minimum value. For a convex optimization problem (either the objective function is a convex or a concave function), the maximum value can only be chose from the boundaries or the stationary point. Hence, the optimal solution can be obtained from (4.21) or (4.22). □

Based on the above theorem, we can design the algorithm to solve the utility maximization problem (4.6). To solve this problem, we can first use the short-path algorithm to select several candidate paths which have the lowest expected delay to decrease the complexity, since a path with a much larger delay cannot be an optimal path. The expected delay for a given path p_{sd}^k can be calculated with (4.11). Then, N paths with the lowest expected delay are selected as the candidate path set $\Omega = \{p_{sd}^k | k = 1, 2, \ldots, N\}$. With the candidate path set Ω, the optimal path and the corresponding optimal searching time will be computed, and the pseudo code of the algorithm is shown in Algorithm 3. Specifically, in Line 4, we initialize the maximum utility to 0. In Line 6-9, the expected utility and optimal transmission time for each path is computed. The optimal path and the corresponding transmission time is selected by Line 10-12.

Note that, the proposed algorithm assumes the scenario that the vehicle arrival rate will remain stable. Then with the algorithm performed at the source node, the optimal path and corresponding optimal transmission time are determined. If the vehicle arrival rate varies with the change of time, then the proposed algorithm can be invoked again at any hot-spot to capture the traffic dynamic where the vehicle carries multimedia data has just arrived.

Algorithm 3 Maximum-Utility Dissemination Algorithm

Input: The set of paths $\Omega = \{p_{sd}^k | k = 1, 2, \ldots, N\}$
Output: The optimal path p_{sd}^* and the corresponding optimal transmission time t_c^*
procedure GETOPTIMALPATH(Ω)
    Set $MaxUtility \leftarrow 0$
    for each path $p_{sd}^k \in \Omega$ **do**
        Derive the expected utility $E^k\{U\}$ by (4.20)
        Solve $\frac{d(E^k\{U\})}{d\ t_c} = 0$, and get the solution set $\mathcal{T}_c^k$
        Get the optimal transmission time t_c^{k*} by (4.21)
        Get the maximum $E_{max}^k\{U\}$ with t_c^{k*}
        if $E_{max}^k\{U\} > MaxUtility$ **then**
            $MaxUtility = E_{max}^k\{U\}$
            $p_{sd}^* = p_{sd}^k$ and $t_c^* = t_c^{k*}$
        end if
    end for
    return p_{sd}^* and t_c^*
end procedure

4.5 Performance Evaluation

In this section, we conduct the simulations with real world traces to evaluate the performance of the proposed dissemination algorithm.

4.5.1 Simulation Setting

To better evaluate the performance of the proposed algorithm, a practical setting based on the real trace collected from about 2,300 taxis in Shanghai between February and March 2007 is considered in this work. Similar to [30,86], the whole Shanghai city area is divided into 40 regions based on the travel distance, which is shown in Fig. 4.3. The network topology is designed based on the clustering regions, and the RSUs are deployed as hot-spots at each region with the highest vehicle density. According to [16], the coverage radius and data rate for each RSU is set to 300 meters and 1 Mbps respectively for stable transmission, and the vehicle travel speed is set to 5 m/s as the urban scenario is considered. So, it is easy to calculate $T = 300 \times 2/5 = 120s$. The simulation ran with the vehicle arrival rate at different time of one day, and 1,000 runs were conducted for each case to obtain the average.

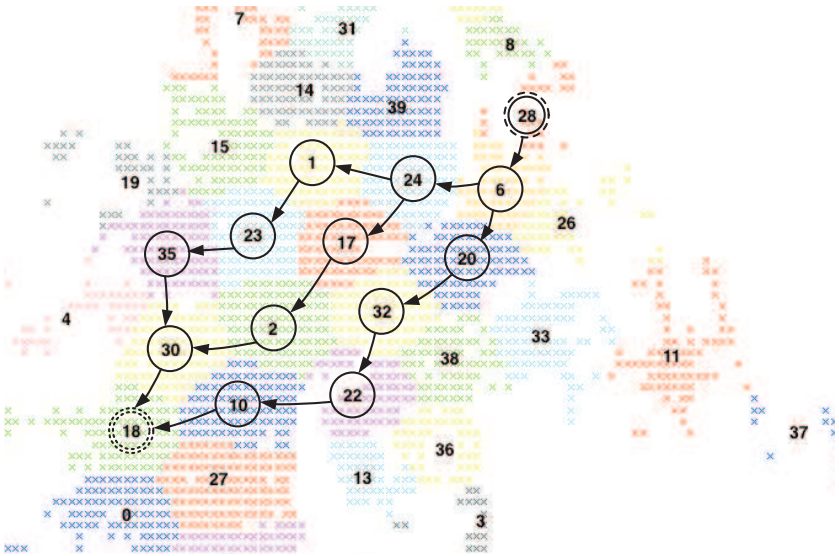


Figure 4.3: Clustering of Shanghai map, with three paths selected from Wujiaochang (cluster 28) to Hongqiao airport (cluster 18).

4.5.2 Case Study

First, as the logarithmic function can guarantee the proportional fairness [29, 49], the utility function of delay is given by

$$F_d = -10 \log \frac{T_{sd}}{T_{max}}, \tag{4.23}$$

where T_{sd} is the delay for any given path p_{sd}, and T_{max} is the maximum tolerable delay. According to some preliminary results, the worst case of the delay is about the twice of the shortest delay, thus we set $T_{max} = 2\sum_{k=1}^{n}(T + t^v_{i_{k-1}i_k})$, 10 is a weight parameter to balance the delay, multimedia quality, and cost utilities.

Second, a real video trace *football* [5] encoded with two different types of video codecs (SVC and CS) are used as an example in our simulation. So, the discrete utility function (for SVC) and continuous utility function (for CS) are both evaluated.

Discrete utility function for multimedia data

Since SVC encodes the video into layers with different quality, thus a stair-case utility function is modeled. Similar to [78], the utility function for the video trace *football* is express as:

$$F_u^{svc} = \begin{cases} 29.16, & 31.5 \text{ Mb} \leq D < 56.13 \text{ Mb} \\ 33.27, & 56.13 \text{ Mb} \leq D < 85.53 \text{ Mb} \\ 35.53, & D \geq 85.53 \text{ Mb} \\ 0, & \text{otherwise.} \end{cases} \tag{4.24}$$

Continuous utility function for multimedia data

When the video is encoded with CS technique, the video quality depends on the total amount of received video measurements. The utility function of the sample video *football* can be express as:

$$F_u^{cs} = 0.51D^{0.53} + 14.62. \tag{4.25}$$

Third, the cost utility can be considered as the summation of the total number of temporary storage. Thus, we have

$$F_c = C\sum_{k=1}^{n} e^{-\lambda_{i_{k-1}i_k}(T-tc)}, \tag{4.26}$$

where C is the weight parameter to balance the cost utility to the other two utilities, and $C = 3$.

For the stair-case utility function, the calculation of the maximal $E\{U\}$ can be simplified, as only the boundaries and the turning points of the utility function need to be considered. While for the CS video, we need to find the the set of values $\mathcal{T}_c$ which will make the derivative of $E\{U\}$ over t_c equal 0, which is easy to calculate. Then the optimal transmission time is obtained by (4.21).

4.5.3 Simulation Results

First, we investigated the vehicle arrival pattern between the adjacent hot-spots. As this work is based on the assumption of the Poisson vehicle arrival, we tried to obtain the distribution of the inter-arrival time of the vehicle arrival of each link, and compare it with the exponential distribution. We obtained all the inter-arrival times of the vehicles traveling from one hot-spot to another in one month from the traces. From the statistics of the average inter-arrival times, we derived the CDF of the inter-arrival time. We found that the derived CDF matches the exponential distribution quite well, which validates the Poisson arrival assumption. In Fig. 4.4, we plot the peak time (9 am - 3 pm), non-peak (1 am - 7 am) time and daily average CDFs of the inter-arrival time from cluster 24 to 17. From the results, we can see that all the three CDFs match the exponential distribution quite well. For other links, we have the similar results.

Then, we conducted simulations based on the Shanghai traces. Since the clusters are divided based on the travel distance, we set the travel time for each link $T_{ij} = 15$ minutes. The source and destination were set to cluster 28 (Wujiaochang) and cluster 18 (Hongqiao airport) respectively. Considering the large number of possible routes from the source to the destination, we selected three shortest paths by combining the Dijkstra's shortest-path and minimum delay derived by (4.11). The selected three shortest paths P1 $\{28, 6, 24, 17, 2, 30, 18\}$, P2 $\{28, 6, 24, 1, 23, 35, 30, 18\}$ and P3 $\{28, 6, 20, 32, 22, 10, 18\}$ are shown in Fig. 4.3. The peak time and non-peak time vehicle arrival rates (in the unit of per second) of the three paths were obtained from the traces, which are shown in Table 4.1. As the existing hybrid framework based dissemination algorithms failed to take the contact time between vehicles into consideration, we compared the results with two simple algorithms, minimum-delay which tries to achieve the minimum delay with the lowest multimedia quality, and

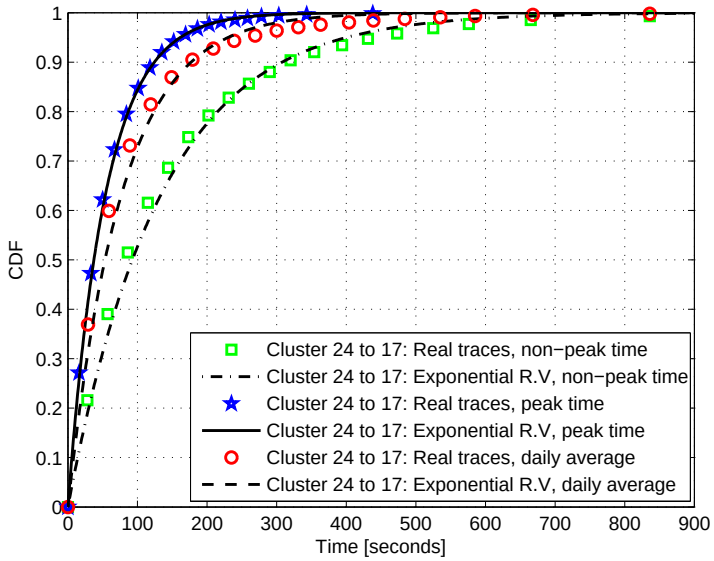


Figure 4.4: The CDF of the vehicle inter-arrival time.

Table 4.1: Vehicle arrival rates of the three shortest paths

	Peak time arrival rate
P1	$\{0.0050, 0.0024, 0.0185, 0.0503, 0.0249, 0.0253\}$
P2	$\{0.0050, 0.0024, 0.0142, 0.0188, 0.0146, 0.0045, 0.0253\}$
P3	$\{0.0050, 0.0160, 0.0197, 0.0133, 0.0021, 0.0042\}$
	Non-peak time arrival rate
P1	$\{0.0014, 0.0011, 0.0086, 0.0200, 0.0127, 0.0106\}$
P2	$\{0.0014, 0.0011, 0.0060, 0.0091, 0.0098, 0.0020, 0.0106\}$
P3	$\{0.0014, 0.0066, 0.0050, 0.0059, 0.0009, 0.0027\}$

maximum-quality which only focuses on the highest multimedia quality while ignores the dissemination delay, to show the benefits of the proposed algorithm.

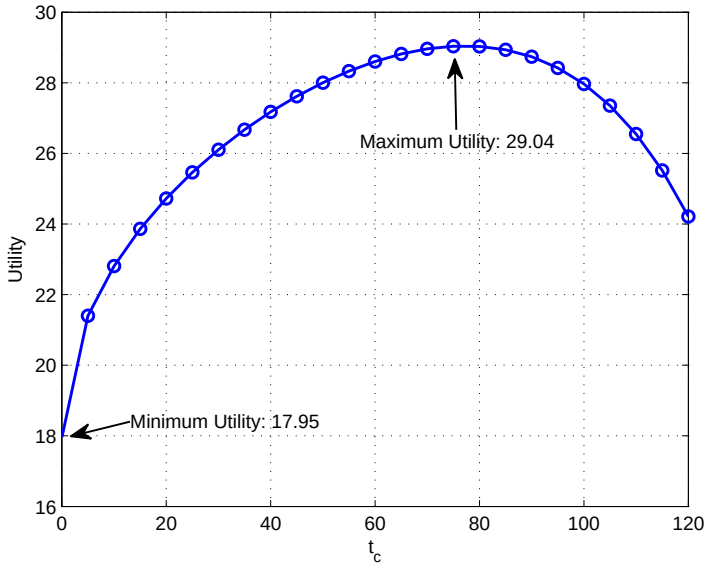


Figure 4.5: The achieved utility versus transmission time t_c.

To show the necessity of the proposed algorithm, we evaluated the influence of the transmission time t_c to the final achieved utility. We tested the different settings of transmission time t_c and calculated the theoretical results of the utility for path 1, and the results are shown in Fig. 4.5. From the results, we can notice that the maximum utility is greater than the minimum utility by 61.8%. The minimum utility is achieved when the transmission time $t_c = 0$, which corresponding to the case that the vehicle carried multimedia data never requests any additional storage help from drop box. In this case, there is no guarantee for the minimum video quality, which will lead

Table 4.2: Results of peak hours

		Theory	Simulation	Delay Minimized	Quality Maximized
SVC	P1	32.38	32.79	6.25	26.25
	P2	28.29	28.44	6.10	22.52
	P3	28.64	29.01	5.95	22.92
	Opt.	32.38	32.79	6.25	26.25
CS	P1	29.04	29.59	6.24	23.13
	P2	24.57	24.73	6.15	18.79
	P3	25.35	25.76	5.9	19.79
	Opt.	29.04	29.59	6.24	23.13

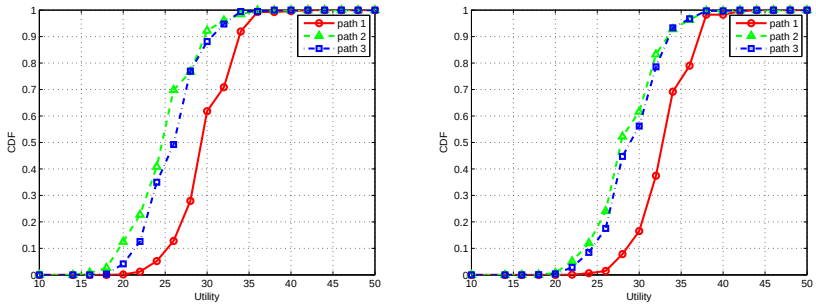
Table 4.3: Results of non-peak hours

		Theory	Simulation	Delay Minimized	Quality Maximized
SVC	P1	26.74	26.78	4.78	22.30
	P2	22.39	22.08	4.45	18.22
	P3	23.13	23.29	3.84	19.56
	OPT	26.74	26.78	4.78	22.30
CS	P1	23.68	23.70	4.67	19.36
	P2	19.58	19.51	4.46	15.49
	P3	21.73	21.70	3.98	17.18
	OPT	23.68	23.70	4.67	19.36

to low utility. While the maximum transmission time $t_c = T$, whenever the tagged vehicle traveling by a RSU, it will request the additional storage services directly without waiting for other vehicles. Although high video quality can be guaranteed, long transmission delay and high cost will bring down the final utility. Therefore, the setting of transmission time t_c to the final achieved utility is crucial.

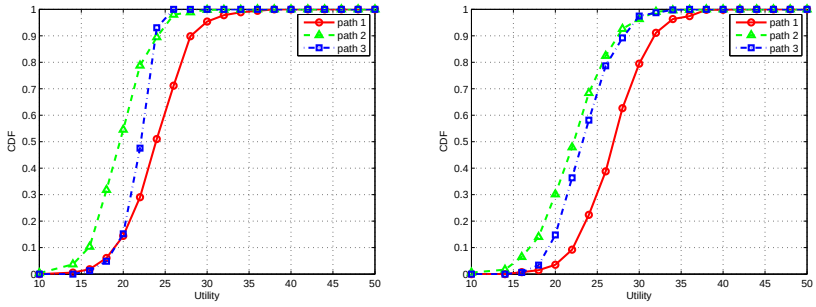
Next we evaluated the performance of the proposed algorithm over the three selected paths, and the simulation results of the peak hours and non-peak hours are shown in Tables 4.2 and 4.3 respectively. These results were obtained with the parameter setting $\alpha = 1$, $\beta = 1$ and $\gamma = 1$. From these two tables, first we can notice that the trace based simulation results match the theoretical results quite well, which demonstrate the correctness of the proposed algorithm. No matter at peak or non-peak time period, transmission along path 1 can always bring the highest utilities for both SVC and CS video. The reason is that the vehicle arrival rate of path 1 is slightly higher than the other 2 paths, which indicates lower probability of requesting

additional storage from drop box. Thus lower cost and higher final utility can be achieved. Since SVC encoding is more efficient than CS encoding, with the same setting, SVC video can bring higher utility than CS video. For the delay minimized algorithm, with the sacrifice of the video quality, the total delay does not reduce much, so very low utility can be achieved. While for the quality maximized algorithm, the high video quality incurred too many additional storage requests, thus lead to lower utility than the proposed algorithm as well. By comparing Tables 4.2 and 4.3, the performance during peak hours can outperform that of non-peak hours, which is quite reasonable. With a higher vehicle arrival rate, there is a larger probability to find a passing vehicle to forward, and higher utility can be achieved.



(a) CS encoded video (b) SVC encoded video

Figure 4.6: CDF of utilities during peak hours.

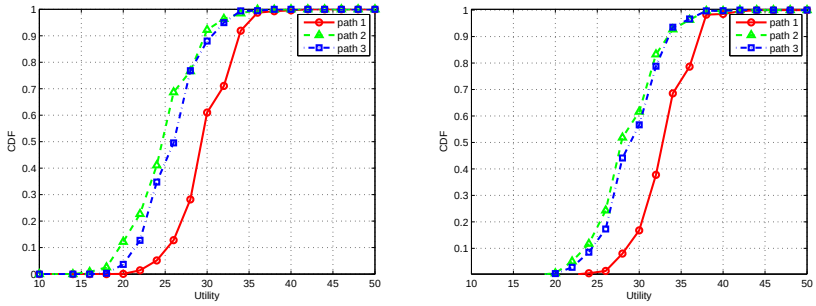


(a) CS encoded video (b) SVC encoded video

Figure 4.7: CDF of utilities during non-peak hours.

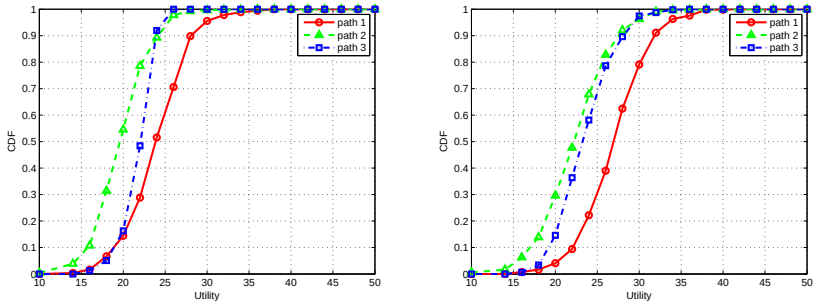
To better show the performance over the three paths, we also plot the CDFs of the final achieved utilities of the three paths during both peak and non-peak hours, as shown in Figs. 4.6 and 4.7. From Fig. 4.6, path 1 has a much higher chance to achieve

high utility than path 2 or 3. Paths 2 and 3 achieve similar results even as they have different number of links. When the total traffic reduces, as shown in Fig. 4.7, the chances to achieve high utility have been greatly reduced. In addition, during non-peak hours, path 3 has a bit higher vehicle arrival rate than path 2, corresponding to a higher achieved utility.



(a) CS encoded video (b) SVC encoded video

Figure 4.8: CDF of utilities with variation during peak hours



(a) CS encoded video (b) SVC encoded video

Figure 4.9: CDF of utilities with variation during non-peak hours

In order to make the simulation settings more practical, we added random variations to the travel time of links between adjacent hot-spots, i.e., $t_{ij} = t_{ij}(1 \pm \varrho 10\%)$. The results are shown in Figs. 4.8 and 4.9 for peak hours and non-peak hours respectively. By comparing the results to that of Figs.4.6 and 4.7, we can achieve almost identical results, even with the 10% fluctuation of the travel time along links. From these results, it can be concluded that the proposed algorithm is robust to the fluctuation of the travel time along links, as it has limited impact to the overall performance.

Table 4.4: The influences of weight parameters setting to results

α	1	3	5	10
Opt. t_c s	77.2 s	74.6 s	72 s	65.4 s
Opt. Utility	29.04	41.05	53.09	83.28
β	1	3	5	10
Opt. t_c	77.2 s	111.8 s	120 s	120 s
Opt. Utility	29.04	97.54	170.11	352.48
γ	1	3	5	10
Opt. t_c	110.4 s	77.2 s	59.3 s	34.3 s
Opt. Utility	36.38	29.04	23.74	13.15

Finally, we investigated the effect of the different weight parameter ($\alpha, \beta,$ and γ) settings with three simulations. For each simulation, we only change one parameter to study the influence to the final utility. The results are shown in Table 4.4 with four different parameter value settings. The top three rows of Table 4.4 show the results of different α. It can be noticed that by increasing the value of α, shorter optimal transmission time will be obtained. It is quite reasonable as shorter transmission time will lead to smaller delay. In the meantime, the utility will be scaled up with larger α as well. β indicates how important the video quality contributes to the final utility. In the middle three rows of Table 4.4, the optimal transmission time is increasing with the increment of β. As the transmission time guarantees the minimum video quality, longer transmission time leads to better video quality. For the penalty parameter, larger γ will significantly reduce the chances of additional storage request. As shown in the bottom three rows of Table 4.4, lower optimal transmission time is obtained with a larger γ. Since lower transmission time means longer waiting time for passing vehicles to forward, and less additional storage requests are needed.

4.6 Conclusion

In this chapter, we have investigated a multimedia dissemination problem in large-scale VANETs under a hybrid framework. A utility-based maximization problem has been formulated to find the best delivery strategy, with the consideration of delivery delay, QoS of the multimedia data and the temporal storage cost. Then, we have obtained the closed-form of the utility functions. The maximization problem has been solved with a maximum-utility dissemination algorithm based on the convex optimization theory. Finally, we have conducted extensive trace-driven simulations to evaluate the performance of the proposed algorithm.

Chapter 5

Conclusions and Further Research Issues

5.1 Conclusions

In this dissertation, we have discussed three different aspects of the multimedia delivery over heterogeneous wireless networks.

- In Chapter 2, we proposed a real-time adaptive best-action search algorithm for video streaming over multiple wireless access networks. First, we formulated the video streaming process as an MDP. To achieve smooth video streaming with high quality, we carefully designed the reward functions. Second, with the proposed rate adaptation algorithm, we can solve the MDP to obtain a sub-optimal solution in real time. Last, we implemented the proposed algorithm and conducted realistic experiments to evaluate its performance and compare it with the state-of-the-art algorithm [27]. The experiment results showed that the proposed solution can achieve a lower startup latency, higher video quality and better smoothness.

- In Chapter 3, the problem of multimedia transmission scheduling among multiple vehicles over the Drive-thru Internet has been investigated. The utility model was devised to map the throughput to the user's satisfaction level. First, the scheduling problem was formulated as an optimization problem to maximize the total achieved utility. Since the optimization problem is NP-hard, the problem was then converted to a finite state decision problem and solved by a

searching algorithm which is served as the performance upper bound. A heuristic algorithm based on the utility potential was proposed to obtain the solution in real time with high performance. Finally, we implemented and conducted extensive simulations to evaluate the performance.

- In Chapter 4, we have investigated a multimedia dissemination problem in large-scale VANETs under a hybrid framework. A utility-based maximization problem has been formulated to find the best delivery strategy, with the consideration of delivery delay, QoS of the multimedia data and the temporal storage cost. Then, we have obtained the closed-form of the utility functions. The maximization problem has been solved with a maximum-utility dissemination algorithm based on the convex optimization theory. Finally, we have conducted extensive trace-driven simulations to evaluate the performance of the proposed algorithm.

5.2 Further Research Issues

There are many open issues beckon for further research in the topics we discussed in this dissertation.

- For the work on real-time adaptive algorithm for video streaming over multiple wireless access networks, we list the possible further research issues as follows. First, to better predict the future bandwidth, the most recent estimation of bandwidth should be assigned with a higher weight, since the bandwidth does not vary dramatically in general. Second, how to set the optimal segment size should be investigated. With a small segment size, the adaptation algorithm can respond to the wireless bandwidth variation more quickly to achieve a smoother video streaming experience. However, more computation and transmission overhead will be brought. While for a large segment size, there is a higher probability to experience playback interruptions or drastic video quality variation. Therefore, it is a trade-off between video streaming experience and computation and transmission overhead. Last but not least, the size of the video segment should be further considered for variable bit rate (VBR) videos to improve the smoothness of video streaming.

- For the work on maximum-utility scheduling for multimedia transmission in

Drive-thru Internet, we have the following directions should be studied in the future. First, the wireless model should take the channel fading and shadowing into consideration. Second, currently we only consider one RSU scenario, how to schedule the transmission when vehicles drive through multiple RSUs need to be investigated. With multiple RSUs, the collaboration between multiple RSUs can better improve the performance. For instance, if one vehicle has not finished the transmission, and it is leaving the coverage of the current RSU soon, it may pause the relatively low rate transmission and wait for the higher data rate in the next RSU's coverage. In this way, a joint scheduling algorithm should be investigated with the consideration of multiple RSUs. Last but not least, it is more practical to consider dynamic vehicle arrival rate.

- For the work on multimedia dissemination in large-scale VANETs, there are several open issues left behind. First, currently only one type of the general vehicles is considered in our work. More types of vehicles with different moving pattern can be incorporated, such as taxi and bus. Since buses run with relatively slower speed under fixed schedule, the delivery strategy should be extended to further consider whether to wait for the next scheduled bus or a random arrival taxi. Second, we set identical transmission time for all RSUs, the different transmission time setting for different RSU should be explored, with the consideration of the vehicle arrival rate. Such as for the hot-spot with low traffic volume, shorter searching time should be set, as there is a low probability to find a passing vehicle to forward. Third, the average vehicle arrival rate of each link is assumed constant in the current work, and dynamic arrival rate should be considered and evaluated. In this circumstance, the dissemination algorithm can be invoked again at the hot-spot which just receives the multimedia data to find an updated optimal path and delivery strategy.

Bibliography

[1] Big Buck Bunny. http://www.bigbuckbunny.org.

[2] Integer programming. http://en.wikipedia.org/wiki/Integer_programming.

[3] Karp's 21 NP-complete problems. http://en.wikipedia.org/wiki/Karp%27s_21_NP-complete_problems.

[4] Network simulator 3. http://www.nsnam.org.

[5] Video Test Media. https://media.xiph.org/video/derf.

[6] Khadige Abboud and Weihua Zhuang. Stochastic analysis of a single-hop communication link in vehicular ad hoc networks. *IEEE Trans. Intell. Transp. Syst.*, 15(5):2297–2307, 2014.

[7] Ahmed EAA Abdulla, Zubair Md Fadlullah, Hiroki Nishiyama, Nei Kato, Fumie Ono, and Ryu Miura. An optimal data collection technique for improved utility in UAS-aided networks. In *IEEE INFOCOM'14*, pages 736–744, 2014.

[8] Saamer Akhshabi, Sethumadhavan Narayanaswamy, Ali C Begen, and Constantine Dovrolis. An experimental evaluation of rate-adaptive video players over HTTP. *Signal Process. Image Commun.*, 27(4):271–287, 2012.

[9] Juan J Alcaraz, Javier Vales-Alonso, and Joan García-Haro. Link-layer scheduling in vehicle to infrastructure networks: An optimal control approach. *IEEE J. Sel. Areas Commun.*, 29(1):103–112, 2011.

[10] Mahdi Asefi, Jon W Mark, and Xuemin Shen. A mobility-aware and quality-driven retransmission limit adaptation scheme for video streaming over VANETs. *IEEE Trans. Wireless Commun.*, 11(5):1817–1827, 2012.

[11] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. DTN routing as a resource allocation problem. *ACM SIGCOMM Comput. Commun. Review*, 37(4):373–384, 2007.

[12] Nadjet Belblidia, Marcelo Dias de Amorim, Luís Henrique MK Costa, Jérémie Leguay, and Vania Conan. Part-whole dissemination of large multimedia contents in opportunistic networks. *Comput. Commun.*, 35(15):1786–1797, 2012.

[13] E. Belyaev, A. Vinel, A. Surak, M. Gabbouj, M. Jonsson, and K. Egiazarian. Robust vehicle-to-infrastructure video transmission for road surveillance applications. *IEEE Trans. Veh. Technol.*, PP(99):1–1, 2014.

[14] Dilip Bethanabhotla, Giuseppe Caire, and Michael J Neely. Utility optimal scheduling and admission control for adaptive video streaming in small cell networks. In *IEEE ISIT'13*, pages 1944–1948, 2013.

[15] Claudia Campolo and Antonella Molinaro. Data rate selection in WBSS-based IEEE 802.11 p/WAVE vehicular ad hoc networks. In *IEEE CSNDSP'10*, pages 412–416, 2010.

[16] Claudia Campolo, Alexey Vinel, Antonella Molinaro, and Yevgeni Koucheryavy. Modeling broadcasting in IEEE 802.11 p/WAVE vehicular networks. *IEEE Commun. Lett.*, 15(2):199–201, 2011.

[17] J. Chen, S. He, Y. Sun, P. Thulasiraman, and X. Shen. Optimal flow control for utility-lifetime tradeoff in wireless sensor networks. *Comput. Netw.*, 53(18):3031 – 3041, 2009.

[18] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiraman, and X. Shen. Utility-based asynchronous flow control algorithm for wireless sensor networks. *IEEE J. Sel. Areas Commun.*, 28(7):1116–1126, September 2010.

[19] Nan Cheng, Ning Lu, Ning Zhang, X Shen, and Jon W Mark. Vehicular WiFi offloading: Challenges and solutions. *Elsevier Veh. Commun.*, 1(1):13–21, 2014.

[20] Man Hon Cheung, Fen Hou, Vincent WS Wong, and Jianwei Huang. Dynamic optimal random access for vehicle-to-roadside communications. In *IEEE ICC'11*, pages 1–6, 2011.

[21] Man Hon Cheung, Fen Hou, Vincent WS Wong, and Jianwei Huang. DORA: Dynamic optimal random access for vehicle-to-roadside communications. *IEEE J. Sel. Areas Commun.*, 30(4):792–803, 2012.

[22] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2013-2018.

[23] Mario De Felice, Eduardo Cerqueira, Adalberto Melo, Mario Gerla, Francesca Cuomo, and Andrea Baiocchi. A distributed beaconless routing protocol for real-time video dissemination in multimedia VANETs. *Comput. Commun.*, 2014.

[24] Jakob Eriksson, Hari Balakrishnan, and Samuel Madden. Cabernet: vehicular content delivery using WiFi. In *ACM Mobicom'08*, pages 199–210, 2008.

[25] K. Evensen, T. Kupka, D. Kaspar, P. Halvorsen, and C. Griwodz. Quality-adaptive scheduling for live streaming over multiple access networks. In *ACM NOSSDAV'10*, pages 21–26, 2010.

[26] KR Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A.F. Hansen, and PE Engelstad. Improving the performance of quality-adaptive video streaming over multiple heterogeneous access networks. In *ACM MMSys'11*, pages 57–69, 2011.

[27] Kristian Evensen, Dominik Kaspar, Carsten Griwodz, Pål Halvorsen, Audun F Hansen, and Paal Engelstad. Using bandwidth aggregation to improve the performance of quality-adaptive streaming. *Signal Process. Image Commun.*, 27(4):312–328, 2012.

[28] Jon D Fricker and Robert K Whitford. *Fundamentals of Transportation Engineering: A Multimodal Systems Approach*. Pearson Prentice Hall, 2004.

[29] J. He, L. Duan, F. Hou, P. Cheng, and J. Chen. Multi-period scheduling for wireless sensor networks: A distributed consensus approach. *IEEE Trans. Signal Process.*, PP(99):1–1, 2015.

[30] Jianping He, Lin Cai, Peng Cheng, and Jianping Pan. Delay minimization for data dissemination in large-scale VANETs with buses and taxis. Technical report, Dept. of ECE, University of Victoria, 2015.

[31] Rongxi He, Humphrey Rutagemwa, and Xuemin Shen. Differentiated reliable routing in hybrid vehicular ad-hoc networks. In *IEEE ICC'08*, pages 2353–2358, 2008.

[32] Jerome Henry. *CCNP Wireless CUWSS Quick Reference*. Pearson Education, 2010.

[33] Xiaoxiao Hou, P. Deshpande, and S.R. Das. Moving bits from 3G to metro-scale WiFi for vehicular network access: An integrated transport layer solution. In *IEEE ICNP' 11*, pages 353 –362, 2011.

[34] Cheng-Hsin Hsu and Mohamed Hefeeda. Flexible broadcasting of scalable video streams to heterogeneous mobile devices. *IEEE Trans. Mobile Comput.*, 10(3):406–418, 2011.

[35] Cheng-Hsin Hsu and Mohamed Hefeeda. A framework for cross-layer optimization of video streaming in wireless networks. *ACM Trans. on Multimedia Comput., Commun., and Applications*, 7(1):5, 2011.

[36] Jin-Bum Hwang et al. Effective video multicast using SVC with heterogeneous user demands over TDMA-based wireless mesh networks. *IEEE Trans. Mobile Comput.*, 12(5):984–994, 2013.

[37] Kenta Ito, Kazuka Tsuda, Noriki Uchida, and Yoshitaka Shibata. Wireless networked omni-directional video distribution system based on delay tolerant network on disaster environment. In *IEEE IMIS'13*, pages 331–335, 2013.

[38] D. Kaspar, K. Evensen, P. Engelstad, and A.F. Hansen. Using HTTP pipelining to improve progressive download over multiple heterogeneous interfaces. In *IEEE ICC'10*, pages 1–5, 2010.

[39] Ingo Kofler, Robert Kuschnig, and Hermann Hellwagner. Implications of the ISO base media file format on adaptive http streaming of h. 264/svc. In *IEEE CCNC'12*, pages 549–553, 2012.

[40] Tomas Kupka, Pal Halvorsen, and Carsten Griwodz. Performance of on-off traffic stemming from live adaptive segmented HTTP video streaming. In *IEEE LCN'12*, pages 401–409, 2012.

[41] Nikolaos Laoutaris, Georgios Smaragdakis, Rade Stanojevic, Pablo Rodriguez, and Ravi Sundaram. Delay-tolerant bulk data transfers on the Internet. *IEEE/ACM Trans. Netw.*, 21(6):1852–1865, 2013.

[42] Yang Li, Farahnaz Naeimipoor, and Azzedine Boukerche. Video dissemination protocols in urban vehicular ad hoc network: A performance evaluation study. In *IEEE WCNC'14*, pages 2611–2616, 2014.

[43] Yong Li, Yurong Jiang, Depeng Jin, Li Su, Lieguang Zeng, and Dapeng Oliver Wu. Energy-efficient optimal opportunistic forwarding for delay-tolerant networks. *IEEE Trans. Veh. Technol.*, 59(9):4500–4512, 2010.

[44] Wan-Seon Lim, Dong-Wook Kim, and Young-Joo Suh. Design of efficient multicast protocol for IEEE 802.11 n WLANs and cross-layer optimization for scalable video streaming. *IEEE Trans. Mobile Comput.*, 11(5):780–792, 2012.

[45] KC Lin, Wei-Liang Shen, Chih-Cheng Hsu, and Cheng-Fu Chou. Quality-differentiated video multicast in multirate wireless networks. *IEEE Trans. Mobile Comput.*, 12(1):21–34, 2013.

[46] Kuang-Hao Liu, Lin Cai, and Xuemin Shen. Multiclass utility-based scheduling for UWB networks. *IEEE Trans. Veh. Technol.*, 57(2):1176–1187, 2008.

[47] Tom H Luan, Lin X Cai, Jiming Chen, Xuemin Shen, and Fan Bai. Engineering a distributed infrastructure for large-scale cost-effective content dissemination over urban vehicular networks. *IEEE Trans. Veh. Technol.*, 63(3):1419–1435, 2014.

[48] Tom H Luan, Xinhua Ling, and Xuemin Shen. MAC in motion: impact of mobility on the MAC of drive-thru internet. *IEEE Trans. Mobile Comput.*, 11(2):305–319, 2012.

[49] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw.*, 8(5):556–567, 2000.

[50] R.K.P. Mok, X. Luo, E.W.W. Chan, and R.K.C. Chang. QDASH: a QoE-aware DASH system. In *ACM MMSys'12*, pages 11–22, 2012.

[51] Farid Molazem Tabrizi, Joseph Peters, and Mohamed Hefeeda. Dynamic control of receiver buffers in mobile video streaming systems. *IEEE Trans. Mobile Comput.*, 12(5):995–1008, 2013.

[52] Christopher Mueller, Stefan Lederer, and Christian Timmerer. A proxy effect analyis and fair adatpation algorithm for multiple competing dynamic adaptive streaming over HTTP clients. In *IEEE VCIP'12*, pages 1–6, 2012.

[53] Farahnaz Naeimipoor and Azzedine Boukerche. A hybrid video dissemination protocol for VANETs. In *IEEE ICC'14*, pages 112–117, 2014.

[54] T. Ngo, H. Nishiyama, N. Kato, T. Sakano, and A. Takahara. A spectrum- and energy-efficient scheme for improving the utilization of mdru-based disaster resilient networks. *IEEE Trans. Veh. Technol.*, 63(5):2027–2037, June 2014.

[55] Pengpeng Ni, Ragnhild Eg, Alexander Eichhorn, Carsten Griwodz, and Pål Halvorsen. Flicker effects in adaptive video streaming to handheld devices. In *ACM MM'11*, pages 463–472, 2011.

[56] Joon-Sang Park, Uichin Lee, Soon Y Oh, Mario Gerla, and Desmond S Lun. Emergency related video streaming in VANET using network coding. In *ACM VANET'06*, pages 102–103, 2006.

[57] J. Reichel, H. Schwarz, and M. Wien. Joint scalable video model 11 (JSVM 11). *Joint Video Team, Doc. JVT- X*, 2007.

[58] C. Rezende, A. Boukerche, H. Ramos, and A. Loureiro. A reactive and scalable unicast solution for video streaming over VANETs. *IEEE Trans. Comput.*, PP(99):1–1, 2014.

[59] Cristiano Rezende, Abdelhamid Mammeri, Azzedine Boukerche, and Antonio AF Loureiro. A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection. *Ad Hoc Netw.*, 17:1–17, 2014.

[60] Somsubhra Sharangi, Ramesh Krishnamurti, and Mohamed Hefeeda. Energy-efficient multicasting of scalable video streams over WiMAX networks. *IEEE Trans. Multimedia.*, 13(1):102–115, 2011.

[61] Fabio Soldo, Claudio Casetti, C Chiasserini, and Pedro Alonso Chaparro. Video streaming distribution in VANETs. *IEEE Trans. Parallel Distrib. Syst.*, 22(7):1085–1091, 2011.

[62] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Trans. Netw.*, 16(1):77–90, 2008.

[63] T. Stockhammer. Dynamic adaptive streaming over HTTP –: standards and design principles. In *ACM MMSys'11*, pages 133–144, 2011.

[64] R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*, volume 28. Cambridge Univ Press, 1998.

[65] Wee Lum Tan, Wing Cheong Lau, OnChing Yue, and Tan Hing Hui. Analytical models and performance evaluation of drive-thru internet systems. *IEEE J. Sel. Areas Commun.*, 29(1):207–222, 2011.

[66] Ktawut Tappayuthpijarn, Thomas Stockhammer, and Eckehard Steinbach. HTTP-based scalable video streaming over mobile networks. In *IEEE ICIP'11*, pages 2193–2196, 2011.

[67] Alvaro Torres, Carlos T Calafate, Juan-Carlos Cano, Pietro Manzoni, and Yusheng Ji. Evaluation of flooding schemes for real-time video transmission in VANETs. *Ad Hoc Netw.*, 2014.

[68] Alvaro Torres, Carlos T Calafate, Juan-Carlos Cano, Pietro Manzoni, and Yusheng Ji. Evaluation of flooding schemes for real-time video transmission in VANETs. *Ad Hoc Netw.*, 24:3–20, 2015.

[69] Alvaro Torres, Pablo Pinol, Carlos T Calafate, Juan-Carlos Cano, and Pietro Manzoni. Evaluating h. 265 real-time video flooding quality in highway V2V environments. In *IEEE WCNC'14*, pages 2716–2721, 2014.

[70] Pierre-Ugo Tournoux, Jeremie Leguay, Farid Benbadis, John Whitbeck, Vania Conan, and Marcelo Dias de Amorim. Density-aware routing in highly dynamic DTNs: The rollernet case. *IEEE Trans. Mobile Comput.*, 10(12):1755–1768, 2011.

[71] Alexey Vinel, Evgeny Belyaev, Karen Egiazarian, and Yevgeni Koucheryavy. An overtaking assistance system based on joint beaconing and real-time video transmission. *IEEE Trans. Veh. Technol.*, 61(5):2319–2329, 2012.

[72] Alexey Vinel, Evgeny Belyaev, Olivier Lamotte, Moncef Gabbouj, Yevgeni Koucheryavy, and Karen Egiazarian. Video transmission over IEEE 802.11 p: real-world measurements. In *IEEE ICC'13*, pages 505–509, 2013.

[73] Miao Wang, Hangguan Shan, Rongxing Lu, Ran Zhang, X Shen, and Fan Bai. Real-time path planning based on hybrid-VANET-enhanced transportation system. *IEEE Trans. Veh. Technol.*, PP(99):1–1, 2014.

[74] S. Xiang and L. Cai. Transmission control for compressive sensing video over wireless channel. *IEEE Trans. Wireless Commun.*, 12(3):1429–1437, 2013.

[75] S. Xiang, L. Cai, and J. Pan. Adaptive scalable video streaming in wireless networks. In *ACM MMSys'12*, pages 167–172, 2012.

[76] Siyuan Xiang. *Scalable Video Transmission over Wireless Networks*. PhD thesis, University of Victoria, 2013.

[77] M. Xing. Wireless bandwidth traces. `http://www.ece.uvic.ca/~mxing/mdp/bdtraces.mat`.

[78] Min Xing, Jianping He, and Lin Cai. Maximum-utility scheduling for multimedia transmission in drive-thru Internet. Technical report, Dept. of ECE, University of Victoria, 2014.

[79] Min Xing, Jianping He, and Lin Cai. Maximum-utility scheduling for multimedia transmission in drive-thru internet. *IEEE Trans. Veh. Technol.*, accepted, 2015.

[80] Min Xing, Jianping He, and Lin Cai. Maximum-utility multimedia dissemination in large-scale VANETs. *IEEE Trans. Mobile Comput.*, submitted, 2015.

[81] Min Xing, Siyuan Xiang, and Lin Cai. Rate adaptation strategy for video streaming over multiple wireless access networks. In *IEEE Globecom'12*, pages 5745–5750, Dec 2012.

[82] Min Xing, Siyuan Xiang, and Lin Cai. A real-time adaptive algorithm for video streaming over multiple wireless access networks. *IEEE J. Sel. Areas Commun.*, 32(4):795–805, 2014.

[83] A. Yaver and G.P. Koudouridis. Utilization of multi-radio access networks for video streaming services. In *IEEE WCNC'09*, pages 1–6, 2009.

[84] Quan Yuan, Ionut Cardei, and Jie Wu. An efficient prediction-based routing in disruption-tolerant networks. *IEEE Trans. Parallel Distrib. Syst.*, 23(1):19–31, 2012.

[85] Deze Zeng, Song Guo, and Jiankun Hu. Reliable bulk-data dissemination in delay tolerant networks. *IEEE Trans. Parallel Distrib. Syst.*, 25(8):2180–2189, 2014.

[86] Lei Zhang, Boyang Yu, and Jianping Pan. Geomob: A mobility-aware geocast scheme in metropolitans via taxicabs and buses. In *IEEE INFOCOM'14*, 2014.

[87] Xiaolan Zhang, Jim Kurose, Brian Neil Levine, Don Towsley, and Honggang Zhang. Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing. In *ACM Mobicom'07*, pages 195–206, 2007.

[88] Yang Zhang and Guohong Cao. V-PADA: Vehicle-platoon-aware data access in VANETs. *IEEE Trans. Veh. Technol.*, 60(5):2326–2339, 2011.

[89] Yang Zhang, Jing Zhao, and Guohong Cao. Service scheduling of vehicle-roadside data access. *Mobile Netw. Appl.*, 15(1):83–96, 2010.

[90] Liang Zhou, Haohong Wang, Shiguo Lian, Yan Zhang, Athanasios Vasilakos, and Weiping Jing. Availability-aware multimedia scheduling in heterogeneous wireless networks. *IEEE Trans. Veh. Technol.*, 60(3):1161–1170, 2011.

[91] Yanyan Zhuang, Jianping Pan, Vidhoon Viswanathan, and Lin Cai. On the uplink MAC performance of a drive-thru internet. *IEEE Trans. Veh. Technol.*, 61(4):1925–1935, 2012.