University of Alberta

DYNAMIC SCHEDULING IN A DELAY-CONSTRAINT VEHICULAR NETWORK: A LYAPUNOV-OPTIMIZATION APPROACH

by

Qiang Guo

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Communications

Department of Electrical and Computer Engineering

Fall 2012
Edmonton, Alberta

Dedicated to

Kevin.

Abstract

Vehicular Network provides essential services to improve the road safety, extend the wireless connection coverage, and allow more reliable vehicle-to-vehicle communication. The key challenge of such networks is throttling the delay to satisfy certain Quality of Service (QoS). In this thesis, we employ the Lyapunov Optimizatin technique to derive an optimal system throughput, while at the same time meets the transmission delay constraint. The algorithm developed is implemented in a frame-based fashion: the decision is made only at the beginning of each vehicle arrival frame, which greatly simplifies the calculation overhead. The algorithm also yields strongly stable queue, which is always upper-bounded by a finite value.

Acknowledgments

Any non-trivial endeavor involves efforts of more than one individual. In the process of finishing my thesis, I owe greatly to many people who have come to my aid and support me wholeheartedly.

First of all, I would like to say a big Thank You to Dr. Hai Jiang, who has been guiding me through my academic course wisely and professionally. His glowing characteristics, integrities, and personalities make an ideal model for me to adhere to when I am setting my foot to the professional world.

Also, I would like to thank my intelligent colleagues and friends: Lijie Huang, Jie Gao, Rongfei Fan, and Zhou Zhou for their share of knowledge and support during my graduate study. When we all are to embark on a new world, I wish you the best. Good luck, fellows!

Last but not the least, I would like to thank Qian Cao, my loving wife and a fantastic mother of our ten month old son, Kevin. It is her dedication and unconditional support that help me through the darkest days when hope is frail and strength is fragile.

Today shouldn't be the same if I am without the help from those marvellous people. So finally, THANK YOU ALL.

Table of Contents

1 Introduction **1**

2 Literature Review **8**

2.1 Delay in Vehicular Network 8

2.2 Lyapunov Optimization Method 13

2.3 Research Motivation and Objective 16

3 Background Knowledge **18**

3.1 Renewal Process 18

3.2 Queues and Stability 19

3.3 Lyapunov Optimization 23

4 Dynamic Scheduling in a Vehicular Network **28**

4.1 System Model 29

4.2 Delay Constraint 32

4.2.1 Delay Description 32

4.2.2 Delivery Delay Constraint and Queue Formulation 33

4.3 Problem Formulation 37

4.4 Lyapunov Drift 38

4.5 Existence of an Optimal Policy 40

4.6 Optimum Policy by Drift-plus-Penalty 44
4.6.1 Vehicular Network Control Algorithm (VNC) 45
4.7 Discussion of Implementation of VNC 52

5 Simulation Results **54**

6 Conclusions and Future Work **62**

List of Figures

4.1 System Model 30

4.2 Vehicle Arrival Process 31

5.1 Converging of Throughput Rate 55

5.2 Throughput rate Versus Input Rate 56

5.3 Number of Violation of Transmission Delay Versus Input Rate . . . 56

5.4 Average transmission delay Versus Input Rate 57

5.5 Cumulative distribution functions(CDF)of queueing delay, transmission delay, and total delivery delay respectively 58

5.6 Maximum delay for different μ 59

5.7 Throughput for $V = 62, 52, 42$ 60

5.8 Comparison with heuristic algorithms 61

List of Abbreviations

Acronyms	Definition
GSM	global mobile communications
TDMA	time division multiplxing access
CDMA	code division multiplxing access
MIMO	multiple-input multiple-output
DTN	delay tolerant networks
QoS	quality of service
CDF	cumulative distribution function
RSU	road side unit

Chapter 1

Introduction

Wireless communication has become increasingly indispensable in our daily life. Nowadays people are shifting more and more work, entertainment, information retrieving, only to name a few, onto the mobile platform. The advantage of this revolutionary transition is that people are no longer restrained physically to a location in order to accomplish those tasks: you can browse the most updated news while on the go instead of sitting in front of a screen; you can download your favorite movies and music on your ipad without dragging inconvenient wires from a plug; you can even pay for your Starbucks via electronic bills issued from your mobile device, saving all monetary transactions. While we easily and happily adapt to this enormous advancement, the enabling technology behind the scene, the wireless communication, has come a long way.

It is far-fetching to elaborate on the ancient story in 1880 when Alexander Graham Bell conducted the world's first ever wireless conversation. We will directly land on late nineties last century where not only markets but also technologies of wireless communication gained their monuments and began their decades of thriving. The first commercial 2G cellular network standardized by Global System for

Mobile Communications (GSM) appeared in Finland in 1991. This event inaugurated the advent of digital networks that took advantage of the ever increasing speed of computer processors. Many technologies are "reinvented" such as Time Division Multiplexing Access (TDMA), Code Division Multiplexing Access (CDMA) to accommodate more users in smaller geographical region. In the meanwhile, many promising networks were emerging from universities and laboratories to satisfy bigger appetites of people. Here we enumerates a number of "hot" technologies that emerged in recent years to either supplement or upgrade current network technologies. Some of them are more visionary in nature than practical, while others already find their ways into becoming parts of international standards.

- **Multiple-Input-Multiple-Output Network** In a MIMO network, devices are equipped with multiple antennas to reap the benefits of *freedom of dimensions*. Combined with powerful space-codes, the MIMO network guarantees increase in channel capacity proportional to number of antennas [18],[19],[20]. This technology is revolutionary in the sense that it provided a tunnel to bypass the "Signal to Noise Ratio (SNR) curse" [37], which means that the channel capacity could not increase any more beyond a SNR threshold.

- **Ad Hoc Network** In certain extreme situations like battle fields and disastrous sites, constructing a network in traditional sense is impossible. Therefore, a temporary and self-managed network is necessary to tackle the communication problems in those emergent situations. The mobile nodes in such networks are capable of self-organizing and peer-to-peer communication. A lot of new problems arise in Ad Hoc networks such as designing new effective protocols, routings, etc. [21],[38],[39].

- **Relaying Network** Relay channel is a mysterious topic in Information Theory. The exact channel capacity of this kind of channel is still eluding scientists' understanding. However, assisted by time division technique and space coding, it is possible to gain the spatial diversity using one or more relay nodes to broadcast the information to the destination [17]. In the situation where a direct link between the source and the destination is absent, relaying is the only way for them to communicate.

- **Cognitive Radio Network** Radio spectrum were not considered as valuable until it was recently discovered that almost all spectrum bands are either registered or occupied [16]. The consequence of this is that new wireless services are blocked away not because of prematurity of technology but lack of suitable spectrums to work on. The advent of Cognitive Radio provides a clever solution to this problem by embedding new services into current spectrum hierarchy. When the license-holders are idle, cognitive users take over the spectrum band to carry out their activities, and when license-holders swoop in, cognitive users should evacuate immediately [22],[23].

Obviously the aforementioned networks strive to resolve the communicating problems in user-concentrated areas, i.e., cities, business buildings, etc. where crowded people demand faster and more reliable network connections. On the contrary, in much less populated places such as rural area and highways, the communication problem is almost equally challenging, if not more. Due to vast areas and scarce population, it is not economical to implement as many infrastructures in such regions as in populated locations. However, the desire for better communication is not dampened down by disadvantageous circumstances. Many recent efforts have been made to address this issue from different perspectives. One stream of research branched from the definition of Delay Tolerant Network (DTN) [40],[24]. Kevin Fall in [10] proposed this architecture in order to address the integration of existing TCP/IP based Internet services in the *challenging network*, whose characteristics include high latency, low data rate, and long queuing times. Compared with other networks, an end-to-end is impossible in such networks. Instead, end-to-end communication relies on time-dependent "contacts" to move packets from one node to another. And this particular transmission medium determines the long delays unique to DTN because: first, the "contacts" may not come for a while; second, the moving speed and distance of "contacts" are undetermined as well.

An advancement in DTN is the DakNet advocated by MIT Media Laboratory. The "Dak" in DakNet which means "postal" in Hindu reflects the purpose of the project: bring Internet connections to remote areas in developing countries in an economical way. In such areas, building conventional landline network backbones is financially impossible. A low budget solution is using vehicles to transfer data among separated "kiosks" such as hubs, post offices such that at the end of the day, all kiosks should obtain the same amount of Internet data. When the vehicles detect the existence of a kiosk in the vicinity, they will utilize Wi-Fi devices to upload/download tons of packets and carry them over to the next stop, synchronizing necessary packets then.

DakNet is merely a glimpse of the vast applications of vehicular communication. Many fields can be predicated to benefit significantly from vehicular communication. For under-connected regions such as highway roads and remote areas, vehicles can be adopted as signalling relays to make regions connected. Vehicles can also play a role as broadcaster of road conditions. For example, when an accident happens in one section of the road where road side unit is unable to send out the information due to lack of available connections, passing cars can forward this information to the next road side unit or dispose the information until meeting one central station that has the authority to do so. Besides broadcasting road condition to prevent further hazardous accidents, vehicle communication is also helpful for the purpose of post-accident investigations. The information received and transmitted is an authenticated image of the accident scene. As a matter of fact, this application is so important and practical that it has been motioned in IEEE 802.11p standard. A particular technology called vehicular ad hoc networking (VANET) [25] is accoladed as key to improve road safety. On the other hand, vehicles can also be the benefiter instead of contributor: in remote locations where Internet connection is poor, vehicles can take advantage of the road side unit (RSU) to perform certain low-traffic tasks such as sending out emails, text messages, or so. The application of vehicular communication is so fruitful that increasingly number of projects have been supervised by international standardization organization, universities, and multinationals. Such a list of projects consists of Network-on-Wheels (NOW) [41], Cooperative vehicles and road infrastructure for road safety (SAFESPOT) [42], and Wireless Access in Vehicular Environment (WAVE) [43].

Vehicular communication has distinct characteristics from other types of communication systems. This difference is introduced mainly by the mobility of vehicles. Some typical characteristics include:

- *Ample Power Supply*: unlike traditional mobile devices which are powered by batteries of short life time, vehicles are not in short of power energy. Another important component of vehicular network, RSU, also have sufficient power supply.
- *Predictable Mobility*: On a road it is clear which direction a vehicle is traveling to. This predictability alleviates system designers the formidable tasks of predicting mobility as in mobile networks.
- *Frequent Disconnection*: Due to the mobility of vehicles, the network will experience network disconnections from time to time. Especially in regions of low traffic, the intervals of arriving cars can be very large.

- *Long Delays*: Unlike traditional wireless networks in which the delay is in the magnitude of milli-second, the delay in a vehicular network can reach as high as seconds, minutes, or even worse, days (which is the case in the DakNet). Therefore, a different approach to address the delay is desired.

While we can merge as many as current wireless communication technologies into vehicular networks, it still takes soliciting efforts to address its some unique challenges.

- *Quality of Service* Because the vehicular network is notorious for its frequent disconnections and instability of communication environment, the issue of QoS is particularly thorny. Two factors contribute to the high latency of vehicular networks. First, the unpredictable waiting time: although cars on road arrive in sequential order, the intervals between sequential arrivals however vary. The network designers need to come up with a nice solution to cope with such disadvantage. Second, the packets take time to travel on roads with vehicles. The delivery duration, which used to be too small to worth concerning, has become an important system design factor.

- *Marketing* Frankly speaking, nowadays the demands for vehicular services have not grown enough to foster a mature market. The car manufacturers still need to unearth potential market-driven products to cater for customers' needs.

In our study, we strive to address the QoS aspect of vehicular communication networks. In detail, we consider the scenario in which vehicles volunteer to relay the packet from one point on the road to another. We divide the QoS into two separate entities: queue dynamics and transmission delay. The objective is to strike a balance between maintaining stable queues and achieving a high throughput of the system. In order to tackle this problem, we will borrow the tool-set from the Lyapunov optimization.

This thesis is broken into following parts: In Chapter 2, a literature review is given to cover various research progress in this field and lay out necessary theoretical background to usher in our development of the problem; In Chapter 3, the essential background knowledge will be introduced to make our presentation more theoretically sound. Our system model and problem formulation are shown in Chapter 4 and the solution based on Lyapunov optimization is given accordingly. Also, several Matlab simulation results are provided to solidify the theoretic results in Chapter 5. Finally, Chapter 6 gives the conclusion of the thesis.

Chapter 2

Literature Review

In this chapter, we introduce related works in literature. While existing literature in this field is bulky if not extensive, we only capture those most closely related to our work. Since our work can be abstracted as "applying Lyapunov optimization to the throughput maximization under transmission delay constraint problem in vehicular networks", it is natural to organize this chapter into two separate parts: In the first part, current progress relating to delay and throughput issues in vehicular networks is reviewed; in the second part, we give an extensive treatment for the current research status on Lyapunov optimization method.

2.1 Delay in Vehicular Network

Vehicular networks are mainly targeting at the emergency applications and road safety control. This kind of services have a high demand for low latency, e.g., the message about an ongoing accident should get through to the control station in a timely fashion[29]. However, as indicated in the previous chapter, vehicular networks suffer severely from frequent disconnections due to its dependence on passing vehicles as transmission mediums. Therefore, the delay problem is dealt with more attentively in vehicular networks than other types of communication networks. Actually, the delay analysis and its optimization are the mainstream topics in this field. Many research works in vehicular networks focus on connectivity and delay analysis. In [4], authors profile the mobility characteristics of vehicles in a vehicle-to-vehicle ad hoc network where two vehicles are labeled as being connected if their distance is within a certain threshold. It is assumed that vehicles are traveling along a multi-lane road and they have separate entry and exit points which are probabilistically distributed along the road. By applying a Poisson process model the authors derive the probability distribution of population and location distributions of vehicles in a certain geometric area. Although the system model represented in this paper is not general enough, it addresses the important issue of traffic modeling and connectivity defining [4]. [5] studies the dispatching of a time-critical packet, with a similar system model as [4], except that several physical-layer parameters, such as fading, path loss, and transmission power are also employed as the yardstick for a valid connectivity. In [5], the authors give the minimum number of vehicles necessary to successfully deliver a time-critical packet. [31] examines the multi-hop packet delivery delay in a vehicle-to-infrastructure communication network. In this network, no reachable direct link exists between the vehicle and the destination infrastructure and the "relay" RSUs are used to forward the packet for the vehicle. Using the theory of effective bandwidth and its dual and effective capacity, the authors derive the maximum distance between a moving vehicle and the infrastructure to meet the delay requirement [31]. The uniqueness of this work is that it presented the statistical variation of the disrupted network channels. [6] explores the spatial propagation of information in sparse and dense vehicle-to-vehicle ad hoc networks. The authors show that the propagation characteristics in such networks depend on various factors, for example, the population of vehicles on the road, average vehicle speed, and the relative movement of vehicles. One interesting finding of [6] is that the packet can have a smaller propagation delay if it is relayed to vehicles traveling backward. While this phenomenon is counter-intuitive, it stimulates some subsequent researches to take advantage of this characteristic. For example, Abdrabou in his work [29] studies the delivery delay between a vehicle and a RSU in sparse vehicle-to-vehicle ad hoc networks. The main contribution of the work is that it gives the closed-form CDF for delivery delay and finds the minimum distance between two RSUs to meet delay requirements numerically. [32] proposes a protocol of vehicle-to-vehicle communication for cooperative collision warning where quick broadcasting of packet is critical. A similar line of work in [33] proposes a two-layer protocol to deliver the safety message to other vehicles in the vicinity. In [32] and [33], latency of the network is divided into two parts: one is the waiting time of the message in the queue, and the other part is the transmission delay. This method of delay analysis will be adopted in this thesis.

While controlling excessive delays is essential for ad hoc wireless networks, the capacity is also an important system goal to achieve. In recent years, a good number of literature studies the throughput performance of a vehicular network. The first break-through came from the information theory community. In [7], the authors show that the mobility in ad hoc wireless networks (e.g., vehicle-to-vehicle ad hoc communication networks) actually increases the system capacity. Specifically speaking, a network consisting of N nodes in motion has a capacity $O(n)$ compared to $O(\sqrt{n})$ of that in a fixed network. An interesting work [8] also coming from information theory community, however provides an opposite insight into the relationship between mobility and throughput. It is found that too much mobility in ad hoc networks could impair the throughout because the overhead (e.g., increased channel uncertainty, network homogeneity, etc.) incurred by high mobility would overshadow the benefits that it brings.

Now it has been established that it is beneficial to employ passing vehicles to forward packets because of the boost of throughput, the problem of balancing between the throughput and delay are placed in front of researchers. In literature, various mathematical tools (such as Lyapunov optimization, stochastic optimization, etc.) are adopted to upper-bound the delay, and at the same time achieving an optimal system throughput. [9] examines the tradeoff between the delay and throughput by introducing redundant packets. Those redundant packets are relayed to multiple nodes in vicinity to carry to the destination, and the first one arriving at the destination will deactivate those arriving later. The strategy employed possesses similarity to the concept of diversity in MIMO: multiple carrying mediums provide multiple paths to the destination; the delay therefore will be reduced probabilistically according to the number of redundancies. It is shown that the redundancy cannot increase capacity, but the delay can be reduced significantly. The tradeoff obtained is given as: $\text{delay}/\text{throughput} \geq O(N)$. And two protocols are given operating on the boundary of tradeoff, yielding delays $O(\sqrt{N})$ and $O(\log N)$, respectively[9].

Even though fixed road-side infrastructures may be employed to relay the packet, all the above reviewed works focus on packet relaying for vehicles. In reality, however, the infrastructure itself may be the "sender" of the packet and under such circumstances, the vehicles will play the role of "relay". Due to its large applications in vast and scarce populated areas such as rural areas and highways, this idea of vehicle-assisted infrastructure to infrastructure communication has attracted many research efforts. DakNet [10] is a project led by MIT Media Lab to bridge the connectivity between wireless infrastructures in poor rural areas. In such areas, infrastructures are extremely deficient. To communicate packets between them, the vehicles traveling on the road are qualified and economic candidates. In this project, data generated in one roadside data post is carried to other posts by passing cars. And then when some other vehicles pass by these posts, they can get these data. After some delays (though could be large), all posts could have the same set of data to serve the areas. For poor-economic areas or sparsely populated geographic locations, the timely information is usually not the main target, and thus certain amount of delays are quite tolerable for such networks [10]. [11] comes up with a protocol called Vehicle-Assisted Data Delivery that uses multiple vehicles to forward packets between fixed data posts. The vehicles carrying the packet will forward it to the next vehicle moving in the vicinity. By exploring predictively of the mobility pattern and road layout, the minimum packet delay is achieved [11]. [34] explores the delay problem in vehicular networks under the umbrella of Mobile Ad Hoc Networks (MANETs) [44]. The author introduces an approach called Message Ferrying (MF) in which a group of mobile units (vehicles) are used to provide wireless connection service for the stationary nodes in the area. Compared to other approaches, MF utilizes the knowledge of the mobility pattern to reduce the randomness of the delivery process. In [1], vehicles help the Road Side Units (RSUs) to relay packet from one place to another. In this work, RSUs are in charge of selecting arrival vehicles according to their moving speed and current backlogs. The main problem can be abstracted in mathematical formulation as

$$\frac{1}{\mathbb{E}[\mathcal{I}]} \limsup_{k\to\infty} \frac{1}{k} \mathbb{E}\Big[\sum_{i=1}^{k} \big(X_i + \frac{Y_i}{V_i}\big) \big| X_1, V_1\Big], \tag{2.1}$$

where X_i is the queue backlog at moment i, Y_i is the decision whether to let the arrival vehicle ferry the packet, V_i is the speed of the arrival vehicle, $\mathcal{I}$ is the interval of vehicle arrivals, and $\mathbb{E}[\cdot]$ means expectation. This problem is a Markov decision problem that is not an easy task to solve. The authors derive a transmission policy that obtains minimum delay.

As far as we know, unlike the vehicle-to-vehicle ad hoc networks, only delay problems are tackled in vehicle-assisted infrastructure to infrastructure communication networks. There is no such work on throughput analysis, not to mention the tradeoff between throughput and delay in such networks. In this thesis, we will derive a scheme to achieve the optimal system throughput, while meeting the transmission delay constraint in such networks by employing the Lyapunov Optimization technique.

2.2 Lyapunov Optimization Method

Lyapunov is a well adopted method which models the stability of dynamic system and optimizes system performance. It is widely used in areas such as automatic control, dynamic system, and chaos theory. The first non-trivial application of Lyapunov Optimization to wireless context was attempted in [12], where authors consider the stability of a queueing network with independent servers. Although the Lyapunov drift (to be detailed in Chapter 3) is not explicitly presented, the same idea as Lyapunov drift is used to derive optimal routing and scheduling policies. Since then the group of Michael Neely made significant contributions to apply Lyapunov Optimization technique in other realms of wireless networks [2], [9], [13], [14]. [13] studies the dynamic power allocation and routing for a multi-node wireless network. The authors devise a joint routing and power allocation algorithm to not only maintain system queue stability (the definition of queue stability will be given in the next chapter) but also limit the whole delay under certain boundary. Besides controlling queue stability as in [13], Lyapunov Optimization also provides a convenient method to optimize system performance in relation to queue status. In [9], the authors derive a dynamic strategy that achieves both dynamic routing and optimal resource allocation. This strategy is performed at each node separately and independently. From [9], we can see two advantages of Lyapunov Optimization. First, compared with methods either targeting at minimizing queue size or maximizing system reward, Lyapunov Optimization can achieve both with little compromise, the tradeoff of which is within the control of system designers. Second, independent decisions at users lead to global-optimal result. A more general application of Lyapunov Optimization to wireless networks is studied in [14]. The authors develop a dynamic control policy to minimize the power expenditure and yet support adaptive transmission rates in a time-varying wireless network. In order to successfully apply the Lyapunov Drift and Drift-plus-Penalty (those lingo will be introduced formally in the next chapter), the power expenditure limit is transformed to a virtual queue. As long as this queue is stable, the power expenditure limit is assured to be satisfied. The benefit of such a policy is that it operates without knowledge of traffic rates or channel information. Yet its long-term result is arbitrarily close to the optimal value achieved by other algorithms, even those that are aware of future evolution of the dynamic system. Another example of using Lyapunov Optimization against dynamic scheduling and stabilizing system queues is [2]. In this work, authors investigate the resource allocation and access control for a group of secondary users wandering about in the vicinity of primary stations. Every secondary user has its own traffic generating process and seeks opportunity to obtain a vacant spectrum to communicate its packets to the central station. In order to coexist with primary users under the same roof, secondary users have to promise that their collisions with primary users are kept below a tolerable level. The authors proposed a scheduling algorithm is to achieve a sub-optimal throughput, the value of which is arbitrarily close to a optimal one. The problems in [2], [9], [14] are Markovian in nature because they build systems on the dynamics of queues, so they can be solved with Markov Decision Optimization where optimization took place over an infinite long time period in order to achieve global optimum. However, the Lyapunov Optimization stands strong contrast to Markovian Optimization because it offers a more lucid structure of the problem and it is very friendly for distributed implementation when the dimension of network grows large. An interesting comparison of Lyapunov methods with Lagrange multiplier is made in [3]. In this paper, the authors give an interesting remark that V in Lyapunov drift-plus-penalty acts much the same way as the Lagrange multiplier in Convex Optimization (V is a tradeoff parameter whose definition will be given in Chapter 3). This finding gives insightful perspective on the mechanism of tradeoff between the system reward and system constraints achieved by Lyapunov Optimization.

Other than aiming at maximizing the time-average quantities (e.g., throughput), Lyapunov optimization is also applicable to maximize functions of such quantities. [15] develops an optimal routing and adaptive scheduling algorithm to maximize instantaneous capacity, which is a function of time-averaged rate [15]. [36] offers an alternative solution to the traditional Markov Decision Problem using the Lyapunov optimization. It develops a dynamic scheduling algorithm that maximizes the *throughput utility*, stabilizes all queues and satisfies all delay constraints. While the optimal *throughput utility* achieved by this algorithm is within controlled proximity of real maximum throughput, the ease enjoyed by this algorithm is unparalleled: it does not need any channel information, predication, and other overheads. In order to satisfy the stochastic feasibility, it adopts an interesting technique called *forced renewal assumption* which forces the system state to "renew" itself at any moment. Although the optimal *throughput utility* achieved by this method is usually sub-optimal, the authors in [36] prove that the sub-optimum will converge to real optimum in polynomial time. The feature of this work is that its artificial creation of a renewal system greatly simplifies the scheduling problem with little sacrifice of performance.

In sum, the tradeoff analysis between maximizing performance and minimizing delay is becoming increasingly important and necessary in the vehicular context, especially for vehicle-assisted infrastructure to infrastructure communications.

Therefore, our effort to explore the joint of these two topics in such networks is natural and meaningful in its own way. In the meanwhile, we can see from the literature review that Lyapunov optimization is a good way to bypass the Markov's "dimension curse" [27] and to balance between the system performance and the delay, and its application in vehicular networks is seldom seen. In this thesis, we will spend considerable effort to discover the structure of our specific problem and see how it fit into the Lyapunov framework.

2.3 Research Motivation and Objective

Although the vehicular networks and relating technologies have been present for a while, there are still large demands for advanced techniques to make them more reliable and useful. Constrained by physical characteristics, the road environment provides a unique challenge for researchers to embody current development of wireless technologies and also invent new ones in order to sustain a satisfactory on-road communication experience for customers. While existing research efforts in other fields of wireless communication have been paid off with numerous mature toolsets, the migration to the vehicular context is still an on-going effort. The main challenge of this category of research is modeling the specific problem in the vehicular network and solve it using familiar methods, which are often available, but need non-trivial modification to adapt to this new environment. One pillar supporting our objective in this thesis is solving a new problem from an otherwise familiar approach.

Nowadays vehicular networks are not satisfied by just being able to link. More safety, larger throughput, and more reliability have become not mere design goals, but real requirements. However, with huge expansion of the amount of information, even on road it can be expected that the burst of information traffic would happen from time to time. Like in applications providing road accidents broadcast services, it is the last thing the system designers want to see that a sudden influx of traffics harms the system stability and therefore brings down the whole system. Thus, it is crucial for the current network to reliably handle this burst of incoming traffic. Except that the system needs to be stable, it is also desirable that its throughput is as high as possible. This purpose serves as the second pillar behind our research motivation. In order to achieve those two objectives, a traffic controlling and scheduling algorithm is needed to guide the road side units transmission policies. We in this thesis propose such a dynamic algorithm.

Chapter 3

Background Knowledge

In this chapter, we briefly list out some necessary techniques and knowledge that will applied later to solve our particular problem. Each section holds credibility for its own purpose; no artificial effort is paid to string them all together to form a logical presentation. They will be referred to whenever the later development makes use of them.

3.1 Renewal Process

In probability theory[26], *renewal process* is a generalization of *Poisson process*. Therefore, let's first define the Poisson process.

Definition 3.1. A Poisson process $N(t)$ is a stochastic process that counts the occurring times of certain event in time interval $(0, t]$.

The simplest type of Poisson process is the one called Levy process, which is characterized by a parameter λ indicating its "intensity". The number of events occurred during time interval $(t, t + \tau]$ is subject to Poisson distribution with a parameter $\tau\lambda$:

$$P\big[N(t+\tau) - N(t) = k\big] = \frac{e^{-\lambda\tau}(\lambda\tau)^k}{k!} \quad k = 0, 1, \ldots, \tag{3.1}$$

where $P[\cdot]$ means probability. The time intervals between any two consecutive events are independent and identically distributed (i.i.d.) random variables with an exponential distribution:

$$P\big[t(k+1) - t(k) = x\big] = 1 - e^{\lambda x}, \quad x \geq 0 \tag{3.2}$$

where $t(k), k = 0, 1, \ldots$ is the time instance when the kth event occurs. One important property of exponential distribution is *memoryless* which is coined by

$$P(T > s + t | T > s) = P(T > t) \quad \text{for all } s, t \geq 0, \tag{3.3}$$

where T is an exponentially distributed random variable. The intuitive behind this memoryless property is that no matter how long we have waited, the average further waiting time is the same. It seems like that it has been forgotten how long the waiting time has been.

3.2 Queues and Stability

Queue is a physical form of Fist-In-First-Serve (FIFS) dynamic structure. Usually we use the following equation to represent a discrete time queue[27]

$$Q(t) = \max\{Q(t-1) - b(t), 0\} + a(t) \qquad t = 1, 2, \ldots, \tag{3.4}$$

where $Q(t)$ is called *backlog* at moment t, $a(t)$ and $b(t)$ are real valued random variables of some stochastic process. $a(t)$ amounts to the number of new work arriving at queue in time t. $b(t)$ represents the amount of work processed by the server of the queue at time instance t. It is assumed that both $a(t)$ and $b(t)$ are non-negative and they are independent if each other. The unit of $Q(t)$ is actually subject to the context. In this thesis, we use *number of packets* as the units of $Q(t)$. Other possible units include *bits, kilobits* or some other system-dependent unit. The dynamics of the queue can be alternatively written without $\max$ operator

$$Q(t) = Q(t-1) - b^{'}(t) + a(t) \qquad t = 1, 2, \ldots, \tag{3.5}$$

where $b^{'}(t)$ is the actual departure work of the queue. Note that when queue backlog is 0, there is no packets to be processed although the server has the capacity to. The definition of $b^{'}(t)$ can then be given as:

$$b^{'}(t) = \min\{b(t), Q(t-1)\}. \tag{3.6}$$

Next we introduce a variety of stability definitions of the queue. The most common constraint of queue stability is perhaps

$$\mathbb{E}\{a(t)\} \leq \mathbb{E}\{b(t)\}, \tag{3.7}$$

where $\mathbb{E}\{\cdot\}$ means expectation. The intuitive behind this is that as long as the arrival rate is less than or equal to the departure rate, the server should have capacity to process all the work in finite time, keeping the queue stablized. This definition is however not applicable to every situation. Sometimes the arrival rate or departure rate is escaping our understanding. For example, in our problem to be formulated in next chapter the departure rate (time average of throughput rate) is actually the optimization objective. Therefore its characteristics cannot be understood before solving the problem at the first place. As a result, we need other metrics to measure the queue stability. First, let's introduce the *rate stability* [27]

Definition 3.2. A discrete time queue process $Q(t)$ is *rate stable* if:

$$\lim_{t\to\infty} \frac{Q(t)}{t} = 0 \quad \text{with probability } 1. \tag{3.8}$$

A discrete time queue process $Q(t)$ is *mean rate stable* if:

$$\lim_{t\to\infty} \frac{\mathbb{E}\{|Q(t)|\}}{t} = 0. \tag{3.9}$$

In mathematics, a sequence of variables $\{X_n\}$ converging in probability to X ($X_n \xrightarrow{P} X$) means that for any positive real number ϵ

$$P(|X_n - X| < \epsilon) \to 1, \text{ as } n \to \infty. \tag{3.10}$$

The *rate stable* has some nice properties. $Q(t)$ is rate stable if and only if $a^{ave} \leq b^{ave}$, where a^{ave} and b^{ave} are defined as

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} a(\tau) = a^{ave} \tag{3.11}$$

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} b(\tau) = b^{ave}. \tag{3.12}$$

Thus, it can be concluded that as long as we know that $Q(t)$ satisfies (3.8), we can obtain the relationship between a^{ave} and b^{ave}.

Another property concerning the quantity of $Q(t)$ is as follows: If $a^{ave} \geq b^{ave}$, then:

$$\lim_{t\to\infty} \frac{Q(t)}{t} = a^{ave} - b^{ave} \quad \text{with probability } 1. \tag{3.13}$$

For the case where arrival rate or departure rate does not have well-defined limits, the following theorem presents a more general necessary condition for rate stability[27].

<u>Theorem</u> 3.1. Suppose $Q(t)$ is a discrete queue process, and $a(t)$ and $b(t)$ are both non-negative stochastic processes. Then:

1. If $Q(t)$ is rate stable, then:

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \big[a(\tau) - b(\tau)\big] \leq 0 \text{ with probability } 1 \tag{3.14}$$

2. If $Q(t)$ is mean rate stable and if $\mathbb{E}\{Q(0)\} \leq \infty$, then

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\big[a(\tau) - b(\tau)\big] \leq 0. \tag{3.15}$$

The *rate stability* is a kind of weak forms of stability because it only describes the long-term average rate of arrival rate and departure rate and their relationship in probability. There is no prediction of whether or when the queue backlog $Q(t)$ exceeds certain value. Next we introduce stronger forms of queue stability: *steady state stable* and *strongly stable*[27]

<u>Definition</u> 3.3. A discrete queue process $Q(t)$ is *steady state stable* if:

$$\lim_{B\to\infty} f(B) = 0, \tag{3.16}$$

where for each non-negative real number $B \geq 0$, $f(B)$ is given by

$$f(B) \triangleq \lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} P\big(|Q(\tau)| \leq B\big). \tag{3.17}$$

A discrete queue process $Q(t)$ is *strongly stable* if:

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\big\{|Q(\tau)|\big\} < \infty. \tag{3.18}$$

Strong stability implies all other sorts of stability. This fact is elaborated in the next theorem[27].

<u>Theorem</u> 3.2. Suppose $Q(t)$ is as described in (3.4) and $a(t)$ and $b(t)$ are both real valued stochastic processes. If $Q(t)$ is *strongly stable*, then:

1. $Q(t)$ is steady state stable;
2. If there is a finite constant C such that $b(t) - a(t) \leq C$ with probability 1 for all t, then $Q(t)$ is rate stable;
3. If there is a finite constant C such that $\mathbb{E}\big\{b(t) - a(t) \leq C\big\}$ for all t, then $Q(t)$ is mean rate stable.

According to the definition of strong stability, the queue backlog will always be finite on average. In our work, we desire this sort of stability over other forms because we need to assure that at each moment in system's life span the queue is stable, no "over-flow" should happen. For example, once the backlog exceeds certain threshold, the system will shutdown or for virtual queue case, the QoS degradation is not permitted. The strong stability guarantees that at any time we will have a queue length that is below the threshold value.

3.3 Lyapunov Optimization

In this section, we introduce the theory of Lyapunov optimization. The essence of Lyapunov Optimization is Lyapunov Drift. Assume a system of N queues denoted as a vector $\boldsymbol{Q}(t) = (Q_1(t), Q_2(t), \ldots, Q_N(t))$. Note that these queues include both actual queues and virtual queues (which behave exactly like actual queues but the underlying physical meaning is not the same as that of real queues. In the next Chapter, the construction of a virtual queue will be detailed). Note that all queue variables are non-negative. In order to measure the volume of $\boldsymbol{Q}(t)$, define a quadratic Lyapunov function of $\boldsymbol{Q}(t)$ as follows:

$$L(\boldsymbol{Q}(t)) \triangleq \frac{1}{2}\sum_{n=1}^{N} \theta_n Q_n(t)^2, \tag{3.19}$$

where $\theta_n, n = 1, 2, \ldots, N$ are the weight coefficients that tag the importance of different queues. $\theta_n's$ have to be positive values. Usually $\theta_n = 1$ since all queues are equally weighted in the system. We conclude that $L(\boldsymbol{Q}(t))$ is non-negative, and equal to zero if and only if all queues are zero. Based on this quadratic Lyapunov function $L(\boldsymbol{Q}(t))$, the one-slot conditional Lyapunov drift is defined as follows:

$$\Delta(\boldsymbol{Q}(t)) \triangleq \mathbb{E}\big\{L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t)) | \boldsymbol{Q}(t)\big\}, \tag{3.20}$$

where the expectation is taken over all possible states of $\boldsymbol{Q}(t)$ in one time slot. The Lyapunov drift has the following important theorem relating to queue stability[27].

<u>Theorem</u> 3.3. Consider the quadratic Lyapunov function $L(\boldsymbol{Q}(t))$ and the case $\mathbb{E}\big\{L(\boldsymbol{Q}(0))\big\} < \infty$. Assume there are constants $B > 0$ and $\epsilon \geq 0$ such that

$$\Delta(\boldsymbol{Q}(t)) \leq B - \epsilon \sum_{n=1}^{N} |Q_n(t)| \tag{3.21}$$

holds for all possible $\boldsymbol{Q}(t)$. Then we have:

1. if $\epsilon \geq 0$, then all queues $Q_n(t)$ are mean rate stable;

2. if $\epsilon > 0$, then all queues are strongly stable and:

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} \mathbb{E}\{|Q_n(\tau)|\} \leq \frac{B}{\epsilon}. \tag{3.22}$$

The above theorem gives condition of queue stability based on the bound of the Lyapunov drift defined in (3.20). This bound on the Lyapunov drift determines the stability of queues in the system. All algorithms desiring to effectively maintain stable queues should devise a mechanism to achieve this bound. This fact makes the framework of Lyapunov Optimization easy to recognize. Almost all algorithms utilizing Lyapunov Optimization can be divided into two parts: the first part strives to satisfy the condition of the above theorem, therefore stabilizing the queues, while the second part aims to achieve some system goals (throughput, power consumption, etc.) These two parts are however not totally separated, they are connected by some control variable that we are going to introduce now.

Suppose the system has a utility function $y(t)$ which is a stochastic process. $y(t)$ can be associated with the system queue vector $\boldsymbol{Q}(t)$ or other system parameters, such as channel conditions, power expenditure limits, etc. Assume that $y(t)$ has a minimum average value y_{min} which is a finite positive real number:

$$\mathbb{E}\{y(t)\} \geq y_{min}. \tag{3.23}$$

Then we have the Lyapunov Optimization theorem [27]:

<u>Theorem</u> 3.4. Assume that $\mathbb{E}\{L(\boldsymbol{Q}(0))\} < \infty$ and there exist constants $B \geq 0$, $V \geq 0$, $\epsilon \geq 0$, and y^* so that the following inequality holds for all time slots

$\tau \in \{0, 1, 2, \ldots\}$:

$$\Delta(\boldsymbol{Q}(\tau)) + V\mathbb{E}\{y(\tau)|\boldsymbol{Q}(\tau)\} \leq B + Vy^* - \epsilon \sum_{n=1}^{N} |Q_n(\tau)| \tag{3.24}$$

Then all queues are mean rate stable. Furthermore, we have the following consequences for system utility $y(t)$:

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{y(\tau)\} \leq y^* + B/V, \tag{3.25}$$

and for queue backlog $\boldsymbol{Q}(t)$:

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} \mathbb{E}\{|Q_n(\tau)|\} \leq \frac{B + V(y^* - y_{min})}{\epsilon}. \tag{3.26}$$

This is the single most important theorem in Stochatis Lyapunov Optimization theory, which establishes the tradeoff between the utility function $y(\tau)$ and queue backlog $\boldsymbol{Q}(t)$.For detailed proof of Theorem 3.4 please look in [27].

Remark 3.1. To sum things up, we give an intuitive explanation of how Lyapunov Optimization works in general. Unlike conventional optimization methods which strives to maximize the *system utility*, Lyapunov Optimization first transforms system constraints into queue stability constraints (i.e., the *Lyapunov drift*), then minimizes the *drift* ($\Delta(\boldsymbol{Q}(\tau))$ in the lefthand side of (3.24)) plus the *penalty* ($V\mathbb{E}\{y(\tau)|\boldsymbol{Q}(\tau)\}$ in the lefthand side of (3.24)). The result of this minimization is that both the system constraints is satisfied as in (3.26) and the penalty is minimized as in (3.25). Note that due to the coupling consideration of both the system constraints and penalty, a parameter $V > 0$ is introduced. V actually controls the "tradeoff" between the system constraints and penalty. This can be shown from

(3.26) and (3.25). We can see from these two equations that by relaxing the requirement on system stability (a larger V, and thus a larger $\frac{B+V(y^*-y_{min})}{\epsilon}$ in (3.26), more flexibility can be made towards optimal system penalty ($y^* + B/V$ is closer to y^* in (3.25)). This leaves us plenty of room to explore for different applications with different requirements.

Specific to Theorem 3.4, The Lyapunov optimization works as follows: For any parameter $V > 0$, we design an algorithm that would satisfy (3.24) for each slot τ, then the time average penalty achieved falls into the range of target value y^* no more than B/V. At the same time, average queue backlog $\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{n=1}^{N}\mathbb{E}\{|Q_n(\tau)|\}$ is controlled below $\frac{B+V(y^*-y_{min})}{\epsilon}$. Thus, the penalty is decreasing with V and the queuing backlog is increasing with V. It is at this point the tradeoff between penalty and system delay comes into play: If more emphasis is placed for minimizing the *penalty*, we should chose a smaller V; If more relaxation is required for *system constraints*, we should chose a larger V.

Chapter 4

Dynamic Scheduling in a Vehicular Network

Vehicular network encompasses three modes of communication: vehicle-to-vehicle, vehicle-to-RSU (road-side-unit), and RSU-to-RSU. Despite that all three modes have individual importance of their own, in this thesis, we mainly focus on addressing scheduling problem for RSU-to-RSU. This problem is particularly challenging because the vehicles employed by RSU-to-RSU link are inherently statistical. Their randomness in, for instance, speed, arrival moment, etc., poses difficulty on efficient scheduling policies for such scenarios. Furthermore, since we mainly focus on network layer where traffic admission and scheduling reside, RSU-to-RSU is a natural target to focus on.

With the evolution of vehicular technologies, nowadays many types of data packets can be carried over on vehicular networks. One of the many is the *real-time packets*. Real-time applications (such as broadcasting, video surveillance, and so on) have strict time deadlines for the packet delivery at the receiver. If the packets arrive at the receiver when the deadline has expired, they will be dropped by the receiver [30]. To deliver such kind of packets is particularly challenging for the vehicular network. When the vehicles are employed as transmission medium, the fluctuating speed of cars and the random arrival time of individual vehicles impose a dilemma for us. On one hand, if we keep waiting for fast cars which can satisfy the delivery deadline, the receiver could be starved to death for that it is possible all cars travel at a low speed and the packets will be piled up at the source; On the other hand, if we take every opportunity to transmit the packets, it is possible that large amount of packets will be discarded at the receiver due to the violation of deadline.

The objective of this chapter is developing a dynamic scheduling algorithm to guide the selection of vehicles in order to maximize the system throughput given that the packet delivery delay requirement imposed by real-time applicatins is satisfied. It is known that the packet delivery delay is naturaly divided into two parts: the waiting delay at the source and the transmission delay from the source to the receiver. It is difficult however to consider these two parts as a whole since they are coupled tightly together [1]. In order to tackle this obstacle, we adopt an approach that transforms the original problem into a sub-problem and thereafter obtain a sub-optimal throughput. The advantage of this methodolgy is two fold: first, the complexity of the original problem is significantly reduced; second, by dividing the delivery delay in parts we are allowed a chance to reveal the tradeoff between the system throughput and the delay. To our best knowledge, this characterization of the tradeoff is not seen in previous literatures. In next section, we will start establishing the system model and fomulate the corresponding problem.

4.1 System Model

Consider a vehicular network where two Road-Side-Units (RSUs), R1 and R2, with distance l are deployed without any physical connection as in Figure 4.1. Due to far

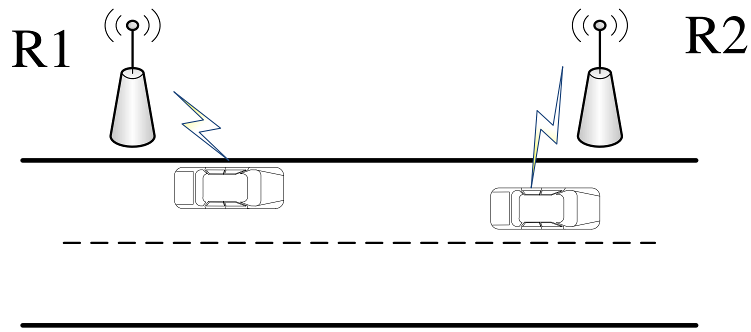

Fig. 4.1. System Model

distance, the radio signal from one side will fade away substantially before reaching the other end. Therefore, the only option for the two RSUs to communicate with each other is employing passing vehicles to forward the signals. In this work, without loss of generality, we assume that R1 intends to employ cars traveling toward R2 to courier the packet for R2. For convenience we assume that the RSUs are able to detect the vehicles once they come into the detectable range. We neglect the time delay incurred by establishing wireless links between RSUs and vehicles. Of course the detection and connection between RSUs and vehicles are by no means naive questions. In the field of vehicular communication, various efforts have been made [1] to tackle this problem. Since this topic is out of the scope of our work, the most ideal case suffices enough for our analysis.

We assume that the arrival times of vehicles at R1 $\{0, T_1, T_2, T_3, \ldots\}$. As in most circumstances, $\{T_k, k \geq 1\}$ can be approximately modeled as a renewal process with independent and identically distributed (i.i.d) inter-arrival times $\mathcal{I}_k = T_{k+1} - T_k,\ k \geq 1$. We call the time period $\mathcal{I}_k$ the kth frame. Figure 4.2 illustrates the time frames. We assume that the cumulative distribution function (CDF) of each frame is known at R1. This can be achieved, for instance, through long-term measurement of road traffic statistics. When the car is in the range of RSUs, we assume the RSUs can immediately become aware of the speed at which the car is traveling. In our

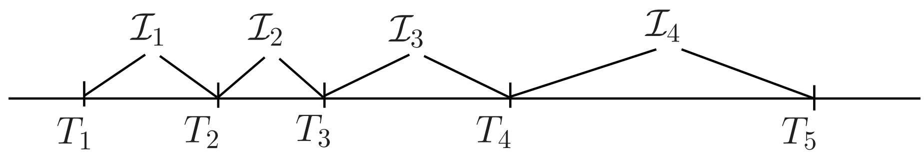

Fig. 4.2. Vehicle Arrival Process

context, we denote the speed of arriving cars as $\{v_1, v_2, \ldots\}$. With this knowledge, the traveling time of the kth car from R1 to R2 can be calculated as $d_k = \frac{l}{v_k}, k \geq 1$. This variable later will constitute an important factor in our decision-making process.

After laying out external settings, next we focus on building up the assumptions in RSUs. R1 maintains a queue in its network-layer to buffer the incoming packets. Since the only transmission opportunities are at instances $T_k,\ k \geq 1$ when cars arrive at R1, we therefore solely examine the queue dynamics and decision making at these precise epoches. For transmissions between cars and RSUs, the transmission mechanism could still be a slot-based one as usually assumed. To implement the slot-based packet uploading and downloading, the only requirement is that the cars need to synchronize with RSUs to begin transmission at the beginning of a slot. Considering the modest speeds on the road, it is not difficult for vehicles to catch this transmission window and therefore fulfill the requirement. With this concern dealt with, we denote $X_k,\ k \geq 1$ as the queue backlog at time instant $T_k,\ k \geq 1$. At T_k, R1 is at liberty to make a decision to transmit $Y_k, k \geq 1$ packets sitting on top of the queue backlog. We assume at most one packet can be transmitted at each turn. Therefore we have $Y_k = \{0, 1\}$[1], $\forall k$. As soon as the controller makes the decision to transmit, we assume, for simplicity, this one packet can be finished successfully.

[1]More scaled action space can be adopted easily into our system model. For example, $Y_k \in \{0, 1, 2, \ldots, N\}$.

The queue dynamic at each decision epoch can be written as

$$X_{k+1} = (X_k - Y_k)^+ + R_k, \tag{4.1}$$

where $(\cdot)^+$ means $\max[\,\cdot\,,0]$ and R_k is the packets admitted into the system by network layer admission control policy, which will be elaborated in Section 4.6.1. The admission policy will be detailed in the sequel section. We assume that the R_k can join the queue in the next frame. Obviously R_k must satisfy the following condition,

$$0 \le R_k \le A_k, \ \ \forall k, \tag{4.2}$$

where A_k represents the packets arrival in the kth frame $\mathcal{I}_k$. $\{A_1, A_2, \ldots\}$ is an i.i.d sequence with mean $\mathbb{E}[A]$ since A_k is independent of both X_k and Y_k. We assume that $\lambda = \frac{\mathbb{E}[A]}{\mathbb{E}[\mathcal{I}]}$ is the packet arrival rate. It deserves to be reemphasized that our analysis is *not* slot-based as in some literatures [2], but is conducted at every epoch when the vehicle arrives.

4.2 Delay Constraint

As discussed in literature review, the delay problem is the major concern in vehicular networks. Thus, in this section, we first specify the components of delivery delay in our system and then give the delay constraint with the goal introduced at the beginning of this chapter.

4.2.1 Delay Description

The delivery delay of the packets in the system consists of two parts. One is the queuing delay when the packets are buffered at R1, and the other is the time cost for cars to carry packets from R1 to R2, which we call transmission delay. In conventional scenarios such as cellular communications[28], queuing delay dominates because the transmission time in air is mostly negligible. However, with the introduction of vehicles as the transmission medium, the speed of delivery determines the transmission delay, and thus becomes a prominent factor determining the delivery delay. And sometimes this component of delay plays a larger part than the queuing delay. For example, in the realistic case discussed in DakNet[10], the data-sinks (with Internet connections) in the remote area can be separated far away and the commuters (usually buses) are rarely seen. Even if we assume that the commuters come often enough and the packets finally get transferred to the carrier, it may take hours or so for them to arrive at the destination. In this situation, the carrier's speed plays a major role. For example, to cover a distance of 100 km a car of speed 100 km/hour uses one hour while a car of speed 20km/hour uses almost 5 hours.

4.2.2 Delivery Delay Constraint and Queue Formulation

In this work, we bound the delivery delay by $\mathcal{D}$ through bounding the queuing delay and the transmission delay μ separately. In real-time applications, $\mathcal{D}$ is predetermined to be different values to meet different requirements [30]. The maximum queueing delay can be determined by X_{max}/λ, where X_{max} is the maximum queue length and λ is the packet arrival rate. We use this value to approximate the queueing delay. Because in our case, two queues(X_k and U_k) are tantem, which means one queue's departure is another's arrival, this approximation is therefore sensible and this method is also employed in [1]. And it can be seen in the simulations in later chapter that this approximation successfully bounds the queueing delay.The design principles of X_{max} will be given in Section 4.6. Given the delivery delay

$\mathcal{D}$ and maximum queueing delay X_{max}/λ, the transmission delay bound μ can be given as:

$$\mu = \mathcal{D} - \frac{X_{max}}{\lambda}. \tag{4.3}$$

In our system model, the cars arrive in sequence with speed $v_1, v_2, \ldots$. At an arriving moment k, if R1 decides to transfer ($Y_k = 1$), then the transmission delay incurred is $d_k = \frac{l}{v_k}$. As stated in the preceding paragraph, we need to impose some constraints on this delay. And the transmission delay constraint can be defined as

$$d_k < \mu. \tag{4.4}$$

If the delivery delay constraint $\mathcal{D}$ is not satisfied, the packet will be dropped. In realistic applications, certain drop rate is not only tolerable but also highly possible. For example, in a DTN network, the droping rate is usually set as 1%. To reflect such practical situation, we define ρ_μ as the upper bound of delivery-delay violation rate.

In this thesis, we control the delivery-delay violation rate only through μ, that is whenever a car fails to deliver the packet to the receiver within μ, it is considered as a violation of the delivery delay $\mathcal{D}$. This is reasonable because our bound on the queuing delay is conservative (We use X_{max} rather than the actual queue length to calculate the queuing delay) and we always use this queuing delay in the determination of transmission delay μ as shown in (4.3). We will also show the validity of this separate bounding in the simulation results. It is admitted that such a conservative policy is strict, however our concerns are practical in two ways. First, it is difficult to calculate the actual queueing time in practice. Actually, whether a packet violates the delay constraint is determined at the receiver by comparing the time stamps of transmission generation and transmission reception in practical applications. Second, a conservative policy is more flexible in handling unexpected situation more gracefully. The "redundancy" rooted in the conservative policy could counteract the effect resulted from, for instance, an improvised temporary speed constraint regulation on the road.

To define the drop rate, we first consider a random variable C_k standing for the event of the breach of delay constraint at the car-arriving moment k (implicitly indicating $Y_k = 1$):

$$C_k = \begin{cases} 1 & \text{if } Y_k = 1 \text{ and } d_k > \mu, \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

With this concept in mind, the transmission-delay violation rate can be written as

$$\lim_{t\to\infty} \frac{1}{\sum_{k=0}^{t-1} Y_k} \sum_{k=0}^{t-1} C_k. \tag{4.6}$$

It should be noted that the average is taken not on all car arriving epochs, but only those when a car arrives and R1 makes a decision to initiate the transmission ($Y_k = 1$).

To simplify further discussion, by multiplying $\frac{1}{t}$ in both denominator and numerator, we rewrite transmission-delay violation rate constraint (4.6) as

$$\frac{c_\mu}{\lim\limits_{t\to\infty} \dfrac{1}{t} \sum\limits_{k=0}^{t-1} Y_k} \leq \rho_\mu, \tag{4.7}$$

where $c_\mu = \lim\limits_{t\to\infty} \dfrac{1}{t} \sum\limits_{k=0}^{t-1} C_k$. Observe in (4.3) that , among two parts of the total delivery delay bound $\mathcal{D}$, the queuing delay X_{max} is related to the control parameter which will be later introduced. The violation on the deadline $\mathcal{D}$ is only allowed to happen through violating the transmission delay μ. Therefore, ρ_μ is as a matter of fact the droping rate at the receiver $R2$. Intuitively, ρ_μ is related to μ: the more stringent the delay constraint is, the less tolerable the system should be.

We re-arrange (4.7) as follows:

$$c_\mu = \lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t-1} C_k \leq \lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t-1} \rho_\mu \mathbf{1}_{Y_k}, \tag{4.8}$$

where $\mathbf{1}_{Y_k}$ is a simple indicator function of Y_k. From (4.8) we find that it resembles the form of a queue system where c_μ plays the role of arrival rate and the right most part the departure rate. Thus we can treat the dynamic entity in (4.7) as a queue. This resemblance is clearer if we write the following queue expression

$$U_{k+1} = (U_k - \rho_\mu \mathbf{1}_{Y_k})^+ + C_k, \tag{4.9}$$

where U_k is the "virtual queue". It is "virtual" because it in fact is not the real queue as X_k and it is "queue" because its dynamic evolvement can be mathematically modeled as a queue. It can be easily deduced that (4.8) is essential to stabilize the virtual queue U_k. In other word, as long as we maintain U_k as a stable queue it is guaranteed that the time-average delay(4.8) constraint is satisfied. This method of transforming system constraints into problem of queue stability was first introduced in [9] and saw its applications in [2].

<u>Remark</u> 4.1. Through the above analysis, a unification of two types of delay is obtained. Both the transmission delay and queuing delay constraints are converted to the queue length constraints. The difference is that one queue is real while the other is "virtual". This unification not only simplifies our analysis by allowing a consistent approach to the problem, but also integrates well into the Lyapunov analysis to be presented in the succeeding section.

4.3 Problem Formulation

In this section, we discuss the formulation of the main problem. The philosophy behind the problem is to achieve an optimized throughput rate, while at the same time satisfying the delay constraint introduced in Section 4.2.2.

Recall that in (4.1) R_k denotes the packets admitted into the queue during time interval $\mathcal{I}_k$. So the time-average rate is

$$r = \lim_{t \to \infty} \frac{1}{T_t} \sum_{k=0}^{t-1} R_k, \tag{4.10}$$

where T_t is the tth arriving epoch of a vehicle. Without harming generality, here we make the assumption that the expectation of $\mathcal{I}_k = T_{k+1} - T_k (k = 1, 2, \ldots, t)$ is bounded as

$$T_{\text{min}} \leq \mathbb{E}[\mathcal{I}_k] \leq T_{\text{max}}, \tag{4.11}$$

and also its second moment is also uniformly bounded by regardless of actions taken by the controller,

$$\mathbb{E}[\mathcal{I}_k^2] \leq D. \tag{4.12}$$

Our main problem then is given as:

$$\begin{aligned} & \max_{\pi} \quad r \\ \text{Subject to} \quad & R_k \leq A_k \quad \forall k \geq 1 \\ & Y_k \leq 1 \quad \forall k \geq 1 \\ & X \text{ and } U \text{ are stable,} \end{aligned} \tag{4.13}$$

where π denotes all possible policies ascribing the sequence of Y_k. This problem is a mix of integer maximization problem and Markov maximization problem. There are no straightforward solutions for such problems. In [1] the authors investigate a problem that bears resemblance to ours. In that work, the authors emphasize on minimizing queuing delay plus transmission delay. While that work originally proposes the tradeoff between queuing delay and transmission delay, it however fails to provide simple enough algorithm to solve the problem. Our work supplements it with two layers of additions. First, we attack the tradeoff problem from a different angle with a step-by-step observe-and-go strategy. Second, we strive to achieve an optimal throughput, which is not considered in [1]. The mechanism empowered us is Stochastic Lyapunov Optimization. From next section we will start with the derivation of Lyapunov Drift.

4.4 Lyapunov Drift

In our system, there are two queues of concern: X_k which is the queue at R1 to receive incoming packets, and U_k which is the "virtual" queue symbolizing the restriction on vehicle speed for delivering packets. In the succeeding analysis, we denote $\boldsymbol{Q}_k = (X_k, U_k)$ as a vector of collection of these two queues at particular epoch k. The Lyapunov function is defined as

$$L(\boldsymbol{Q}_k) \triangleq \frac{1}{2}[X_k^2 + U_k^2]. \tag{4.14}$$

Note the coefficient $\frac{1}{2}$ means we place equal importance on two queues. Lopsided coefficients can reflect preference of certain queue over the other. For instance, $0.1X_k^2 + 0.9U_k^2$ conveys our emphasis on a more rigorous policy for transmission delay.

The Lyapunov drift can be written as

$$\Delta_k = \mathbb{E}\left\{L(\boldsymbol{Q}_{k+1}) - L(\boldsymbol{Q}_k)|\boldsymbol{Q}_k\right\} \tag{4.15}$$

By substituting (4.1), (4.9), and (4.14) into (4.15), we can extend the drift as:

$$\begin{aligned}
\Delta_k &= \frac{1}{2}\mathbb{E}\left\{ X_{k+1}^2 + U_{k+1}^2 - X_k^2 - U_k^2 \middle| \boldsymbol{Q}_k \right\} \\
&= \frac{1}{2}\mathbb{E}\left\{ \left[(X_k - Y_k)^+ + R_k\right]^2 - X_k^2 + \left[(U_k - \rho_\mu \mathbf{1}_{Y_k})^+ + C_k\right]^2 - U_k^2 \middle| \boldsymbol{Q}_k \right\} \\
&\leq \frac{1}{2}\mathbb{E}\Big\{ (X_k - Y_k)^2 + R_k^2 + 2R_k(X_k - Y_k) - X_k^2 \\
&\quad + (U_k - \rho_\mu \mathbf{1}_{Y_k})^2 + C_k^2 + 2C_k(U_k - \rho_\mu \mathbf{1}_{Y_k}) - U_k^2 \Big| \boldsymbol{Q}_k \Big\} \qquad (4.16) \\
&\leq \frac{1}{2}\mathbb{E}\Big\{ Y_k^2 + R_k^2 + 2X_k(R_k - Y_k) \\
&\quad + (\rho_\mu \mathbf{1}_{Y_k})^2 + C_k^2 + 2U_k(C_k - \rho_\mu \mathbf{1}_{Y_k}) \Big| \boldsymbol{Q}_k \Big\} \\
&\leq B + X_k \mathbb{E}\left\{R_k - Y_k \middle| Q_k\right\} + U_k \mathbb{E}\left\{C_k - \rho_\mu \mathbf{1}_{Y_k} \middle| \boldsymbol{Q}_k\right\},
\end{aligned}$$

where $B \leq \frac{Y_k^2 + R_k^2 + (\rho_\mu \mathbf{1}_{Y_k})^2 + C_k^2}{2}$. Considering that $Y_k \leq 1$, $R_k \leq A_k \leq A_{\text{max}}$[1], $C_k \leq 1$, and ρ_μ is a fixed threshold, B is a finite constant. The first inequality in (4.16) follows from the characteristic of $(\cdot)^+$:

$$\begin{aligned}
\left[(X_k - Y_k)^+ + R_k\right]^2 &= \left[\max\left\{X_k - Y_k, 0\right\} + R_k\right]^2 \qquad (4.17) \\
&= \left[\max\{X_k - Y_k, 0\}\right]^2 + R_k^2 + 2R_k \max\left\{X_k - Y_k, 0\right\} \\
&\leq \left(X_k - Y_k\right)^2 + R_k^2 + 2R_k X_k.
\end{aligned}$$

Next let's examine the drift-plus-penalty, which combines the concerns for both

[1]Place an upper bound on A_k is of physical meaning as well as its probabilistic underlying. For R1, there exists a maximum number of packets it can take due to physical restrictions.

queue drifts and "penalty" of the system. This is the place where the tradeoff comes into play. The "penalty" will be scaled by a factor V to reflect the intensity of emphasis on penalty. In addition, the "penalty" in our context, as matter of fact, really means "reward". Therefore the minimization of the penalty is the same as maximization of the reward. Let's modify (4.16) by adding the scaled version of reward:

$$\begin{aligned}\Delta_k - V\mathbb{E}\{R_k|\boldsymbol{Q}_k\} \le B - V\mathbb{E}\{R_k|\boldsymbol{Q}_k\} + X_k\mathbb{E}\{R_k - Y_k|\boldsymbol{Q}_k\} \\ +U_k\mathbb{E}\{C_k - \rho_\mu \mathbf{1}_{Y_k}|\boldsymbol{Q}_k\}.\end{aligned} \tag{4.18}$$

We can see from the above equation that V controls the significance of $\mathbb{E}\{R_k|\boldsymbol{Q}_k\}$. When $V \to 0$, all our concerns go to only queue drifts and we neglect the impact of system penalty total. On the contrary, the larger the V, the more prominent system penalty will be. By adjusting this threshold parameter, different strategies can achieve different tradeoffs.

4.5 Existence of an Optimal Policy

Before diving into the proof of existence of an optimal policy, let's first recap the gist of Lyapunov method. A normal approach to an average cost minimization (maximization) problem usually involves the labored path to a calculated final result. For a below-medium sized problem this approach is preferred because an exact result is not only optimal, but also beneficial for implementations. However, for large problems an exact result is usually obtained in sacrifice of other equally important factors such as complexity, speed, and etc. Not to even mention in some situations it is impossible to obtain an exact result. The beauty of Lyapunov method lies in its frugality of constraints. By satisfying some mild constraints, it can push the average cost as close to the optimal value as we want it. All we need to provide are conditions on which an optimal value might exist. In some sense, this idea parallels that in numerical optimization where Lagrange multiplier is inserted to adjust relationship between the utility and constraints. As a matter of fact, this implicit layer of relationship between Lyapunov's V and Lagrange's multiplier is discussed in [3]. Note during solving a Lyapunov problem, there is no need to know what the optimal result is. The mere acknowledgement of existence of an optimal solution is sufficient to allow us a policy to push the average cost infinitely close to the target. In this section, we will confront this problem by applying techniques similar to [27].

We would like to first define a family of random stationary algorithms for any given input rate λ inside or outside of system capacity region as in the following definition.

Definition 4.1. An Random Stationary Algorithm is the one which in each frame k, makes the decision Y_k independently and probabilistically chooses Y_k from a probability space.

Intuitively speaking, a random stationary algorithm is the one which can achieve a stationary throughput rate over long period of time by adopting a random policy. For instance, we can make $Pr(Y_k = 1) = Pr(Y_k = 0) = \frac{1}{2}$ for each step.

The system throughput in (4.10) is equivalent as

$$r = \lim_{t\to\infty} \frac{\sum_{k=0}^{t-1} R_k}{\sum_{k=0}^{t-1} \mathcal{I}_k} = \lim_{t\to\infty} \frac{\frac{1}{t}\sum_{k=0}^{t-1} R_k}{\frac{1}{t}\sum_{k=0}^{t-1} \mathcal{I}_k}. \tag{4.19}$$

Since frames are independent from each other due to its renewal nature, and also independent from the evolving queues (the case in which frames are determined somehow by the queue and decisions made during each frame is more complicated

[27]), by the law of large numbers, time average is equal to the expectation over one frame with probability 1. Thus we have:

$$\Pr\Big\{|\bar{\mathcal{I}} - \mathbb{E}(\mathcal{I})| > \epsilon\Big\} = 0, \tag{4.20}$$

where ϵ is any positive real number, $\bar{\mathcal{I}} = \frac{1}{t}\sum_{k=0}^{t-1}\mathcal{I}_k$ is time average of the renewal time intervals, and $\mathbb{E}(\mathcal{I})$ is the expectation of time interval over one frame. We conclude that the ergodic behavior of time interval $\mathcal{I}_k$ is promised by the renewal nature of arrival times. Now let's examine if the numerator of (4.19) presents the same ergodic characteristic. Note that in Definition 4.1 we designed the algorithm in such a way that it will independently and probabilistically choose the action according to the some constant distribution. As a matter of fact, this definition already pins down the ergodicity of the R_k. Therefore in the sequel investigations we will use the expectation representation form of (4.19).

Definition 4.2. The problem in (4.13) is feasible if there is an random stationary alogrithm that satisfies:

$$\lim_{t\to\infty}\frac{\sum_{k=0}^{t-1}C_k}{\sum_{k=0}^{t-1}Y_k} \leq \rho_\mu. \tag{4.21}$$

Note that ρ_μ is a pre-defined parameter describing the maximum tolerable ratio of delay constraint violations. And C_k as mentioned before, is a random variable in favor of decision variable Y_k and vehicle speed v_k. According to our assumption, both Y_k and v_k are i.i.d. respectively, plus their distributions are independent from each other, so we can rewrite the left hand side of feasibility condition in Definition 4.2 as:

$$\lim_{t\to\infty}\frac{\sum_{k=0}^{t-1}C_k}{\sum_{k=0}^{t-1}Y_k} = \lim_{t\to\infty}\frac{\frac{1}{t}\sum_{k=0}^{t-1}C_k}{\frac{1}{t}\sum_{k=0}^{t-1}Y_k} = \frac{\mathbb{E}\big\{C(\pi)\big\}}{\mathbb{E}\big\{Y(\pi)\big\}}, \tag{4.22}$$

where $C(\pi)$ and $Y(\pi)$ are random variables when the algorithm π is adopted. And the expectations are all taken over one frame. However, $C(\pi)$ is a composite random variable that is comprised of two independent variables $Y(\pi)$ and v. Note that v is the random variable of vehicle speed and hence has nothing to do with the random stationary algorithm π . Then we can derive (4.22) further that:

$$\begin{aligned} \frac{\mathbb{E}\{C(\pi)\}}{\mathbb{E}\{Y(\pi)\}} &= \frac{\mathbb{E}\{Y(\pi)\}\mathbb{E}\{\hat{v}\}}{\mathbb{E}\{Y(\pi)\}} \\ &= \mathbb{E}\{\hat{v}\}, \end{aligned} \tag{4.23}$$

where $\hat{v}$ is the event that $\frac{l}{v} > \mu$. To make our analysis simple yet without loss of generality, here we will pose certain constraints on the vehicle speed. First, the speed v has a maximum and minimum value, that is $v_{\text{min}} \leq v \leq v_{\text{max}}$. This constraint is plausible and reasonable because both for safety concerns and mechanic limits, vehicles traveling on roads, how remote they are, have some speed limits. Next, we assume v takes its value from a finite discrete set instead of from a continuous range, that is $v = \{v_1, v_2, \ldots, v_N\}$, where $v_n,\ n = 1, 2, \ldots, N$ is in ascending order. This constraint is acceptable since most cars travel in a relative fix speed and, for example, a speed of 60kmh and 60.5kmh differ little when applied to a medium distance such as suitable in our investigation. Therefore $\mathbb{E}\{\hat{v}\}$ can be written as,

$$\mathbb{E}\{\hat{v}\} = \sum_{n=1}^{N} \mathbf{1}_{v_n < l/\mu} Prob\{v = v_n\} \tag{4.24}$$

By Definition 4.2 the only prominent condition on which there exists a random stationary algorithm is $\mathbb{E}\{\hat{v}\} \leq \rho_\mu$. Therefore, for a reasonable demand on μ and ρ_μ this condition is easily satisfied. With this assurance, we present the following lemma [27]

Lemma 4.1. If there is an algorithm in Definition 4.1 that satisfies the condition stated in Definition 4.2, then there is an optimal random stationary algorithm

(ORSA) that satisfies

$$\mathbb{E}\{R^{ORSA}\} = \mathbb{E}\{\mathcal{I}\} r^* \tag{4.25}$$

$$\mathbb{E}\{C^{ORSA}\} \leq \mathbb{E}\{\mathcal{I}\} \rho_\mu, \tag{4.26}$$

where r^* is the infimum of $\frac{\mathbb{E}\{R(\pi)\}}{\mathbb{E}\{\mathcal{I}\}}$ over all stationary, random algorithms that meet the constraint in Definition 4.2.

Proof. By following the steps in [1], we can derive one optimal R^{ORSA}, which proves the existence of a ORSA. Here we omit the specific proof for it can be found in [1], [12], [13]. $\square$

In Lemma 4.1 r^* is the optimal throughput that can be achieved by any random stationary algorithm. We will regard this value as our target and our objective is to push the time average as close to this target as possible.

4.6 Optimum Policy by Drift-plus-Penalty

Now we have established the existence of an optimal random and stationary policy in the previous section, our main objective then becomes devising an algorithm to drive the Lyapunov drift-plus-penalty to this optimal r^*. Observe in (4.18) that in order to minimize the drift-plus-penalty on the left side, we can minimize the right side of the equation. By rearrangement, our approach to the minimization has been simplified to choosing at the begining of each frame, that is the arrival epoch of a vehicle, a sqeuence of $Y_k, k = 1, 2, \ldots,$ which will minimize the following expression,

$$\frac{\mathbb{E}\big\{R_k(X_k - V)\big\} + \mathbb{E}\big\{U_k C_k - X_k Y_k - \rho_\mu \mathbf{1}_{Y_k}\big\}}{\mathbb{E}\big\{\mathcal{I}_k\big\}}, \tag{4.27}$$

and update the virtual queue U_k according to (4.9) at the end of each epoch, which in fact is equal to the beginning of the next frame.

Note in (4.27) the denominator $\mathbb{E}\{\mathcal{I}_k\}$ is independent from any action taken, therefore is known in the system. As a result we only need to minimize the numerator of (4.27). By now, we have successfully screened out the renewal nature in our problem.

Remark 4.2. Our problem is simplified because the renewal process $(\mathcal{I}_k, k = 1, 2, \ldots)$ is independent from system's status and decision. This "fortunate" coincidence is however not artificial: in most vehicle communication scenarios, it is assumed cars arrival process is independent which is subject to certain distribution. Though the model adopted here is prevalent, we can still conjure up a scenario where $\mathcal{I}$ is actually partially decided by the policy. For example, when some constraints on queue length or some particular decision is made, the system will ignore the next coming car. Thus the length of inter-arrival time period is literally modified, depending on the action that will be taken for each step. For the remedies to such problems, the work of [27] has made interesting explorations.

By observing the structure of (4.27), it can be found that R_k and Y_k are totally decoupled. It means we can devise separate techniques to minimize each part containning them and as a consequence minimize the whole (4.27). Therefore, we formally state our dynamic scheduling algorithm as the following:

4.6.1 Vehicular Network Control Algorithm (VNC)

Remember that V is a control parameter, and let's denote the action decision made by VNC as R_k^{VNC} and Y_k^{VNC}. VNC is composed of two parts and the decision is made for each frame and only based on local information. The R_k^{VNC} and Y_k^{VNC} are determined according to following steps

- **Traffic Control Policy** At the beginning of a stage k[1], the allowable amount of packages is decided according to the policy:

$$R_k^{VNC} = \begin{cases} 0 & \text{if } X_k \geq V \\ A_k & \text{else } X_k < V \end{cases} \tag{4.28}$$

- **Transmission Control Policy** At each frame, minimize the following expression by choosing Y_k^{VNC} in its action space $\{0, 1\}$.

$$U_k C_k - X_k Y_k - \rho_\mu \mathbf{1}_{Y_k}, \tag{4.29}$$

  where X_k is the package queue length, U_k is the virtual queue length, and C_k is a simple composite random variable whose value is determined by Y_k and v_k, the vehicle speed that is available at the beginning through speed measurement. Because the action space of Y_k is finite, the minimum of (4.29) is well defined and easy to attain.

Algorithm 1 Vehicular Network Control Algorithm at R1

1: According to the total delivery delay $\mathcal{D}$ and vehicle speed on the road, choose μ.
2: Initialize V according to (4.3). Initialize $X_0 = 0$ and $U_0 = 0$.
3: When a vehicle arrives at time epoch k, compare X_k and V. If $X_k \geq V$, $R_k = 0$; otherwise $R_k = A_k$.
4: Calculate $U_k C_k - X_k Y_k - \rho_\mu \mathbf{1}_{Y_k}$ with $Y_k = 1$. If it is smaller than 0, transmit a packet to the vehicle; otherwise go to Step 5.
5: Update X_{k+1} and U_{k+1} according to (4.1) and (4.9).
6: Wait until time epoch $k + 1$ and go to Step 3.

---

[1] This policy can be carried out either all at once at the beginning or executed during the whole interval since the action does not affect current frame in any explicit manner

A formal description of the VNC algorithm is given in Algorithm 1. Before starting the analysis of the above algorithm, we make the following remark concerning the relationship between the randomly stationary optimal policy and the policy advised by our algorithm. It is noted that our algorithm makes decision on X_k, U_k, and v_k. The queue status variable X_k and U_k are histories of the system, therefore irrelevant to the control action in the current frame; v_k can be regarded as an independent random variable that evolves on its own (if we consider the measurement of v_k as an event before the frame k, v_k can also be ascribed as history). Since those information are defined by the history before frame k, and are not affected by the decision made by our algorithm, we make the following Lemma

Lemma 4.2. Concerning (4.18), we claim that the VNC minimizes the right-hand of the equation over all feasible algorithms including the ORSA stated in Section 4.5.

Proof. From the Algorithm 1, we can observe that for each arrival of vehicles, VNC minimizes the right hand side of (4.18). Then by the theorem of opportunistically minimizing an expectation stated in Chapter 3, it can be claimed that VNC minimizes over all feasible algorithms. Since ORSA is also feasible, therefore Lemma 4.2 is proved. □

In the following we denote all variables relating to our algorithm with a superscript VNC, and every variable concerning the Optimal Random Stationary Algorithm with a superscript ORSA. Therefore, by the prediction of Lemma 4.2 we have

$$\begin{aligned}\Delta_k^{VNC} - V\mathbb{E}\{R_k^{VNC}\} \leq B - V\mathbb{E}\{R_k^{\text{ORSA}}\} + X_k\mathbb{E}\{R_k^{\text{ORSA}} - Y_k^{\text{ORSA}}\} \\ + U_k\mathbb{E}\{C_k^{\text{ORSA}} - \rho_\mu \mathbf{1}_{Y_k^{\text{ORSA}}}\} \qquad (4.30)\end{aligned}$$

In order to reap the benefits of Lyapunov method, we first need to bound the right hand of equation (4.30). The technique we will use is borrowed from [2] which takes advantage of finite dynamics of a queue and converging of a functional of certain random variable. Let's examine a period of time from epoch moment $k-j$ to epoch moment k. During this period of time, the bounds of both queue X_k and U_k can be given by

$$\begin{aligned} X_{k-j} + jA_{max} \geq & X_k \geq X_{k-j} - j \\ U_{k-j} + j \geq & U_k \geq U_{k-j} - j\rho_\mu \end{aligned} \tag{4.31}$$

Obviously the upper bound is achieved when no departure but only arrival happens, while the lower bound is achieved when departure is the only occurrence in the system. Notice l, A_{max}, ρ_μ are all finite according to assumption, therefore the bounds are finite as well. Substitute (4.31) into (4.30) to obtain

$$\begin{aligned} \Delta_k^{VNC} - V\mathbb{E}\big\{R_k^{VNC}\big\} \leq & B + B_X + B_U + X_{k-l}\mathbb{E}\big\{R_k^{\text{ORSA}} - Y_k^{\text{ORSA}}\big\} \\ & + U_{k-l}\mathbb{E}\big\{C_k^{\text{ORSA}} - \rho_\mu \mathbf{1}_{Y_k^{\text{ORSA}}}\big\} \\ & - V\mathbb{E}\big\{R_k^{\text{ORSA}}\big\}, \end{aligned} \tag{4.32}$$

where

$$\begin{aligned} B_X &= jlA_{max}^2 + j \\ B_U &= j^2 + j\rho_\mu \end{aligned} \tag{4.33}$$

reflect the maximum fluctuation of queues during $(k-j, k)$. Now we still have two items in the right-hand side of (4.32) unbounded. In order to bound these two items, we first propose the next Lemma

Lemma 4.3. $X_{k-j}\mathbb{E}\{R_k^{ORSA} - Y_k^{ORSA}\}$ and $U_{k-j}\mathbb{E}\{C_k^{ORSA} - \rho_\mu \mathbf{1}_{Y_k^{ORSA}}\}$ are random variables that will converge exponentially fast to finite values according to j.

Proof. From the definition, the ORSA will only make decision based on observed vehicle speed information $v_1, v_2, \ldots$. Therefore, Y_k^{ORSA} is a function of converging random variable, by converging axioms of functional of random variables, we can deduce that Y_k^{ORSA} will converge to its steady state. Besides, X_{k-j} also has maximum queue length X_{max}. As a result, $X_{k-j}\mathbb{E}\{R_k^{ORSA} - Y_k^{ORSA}\}$ will eventually converge to zero given enough time for Y_k^{ORSA} to fall into its steady state. We will leave out the calculation of this converging time since it is of little reverence to our development. □

With all these bounds achieved, we are finally ready to give the equation that will promise us desired Lyapunov features.

$$\Delta_k^{VNC} - V\mathbb{E}\{R_k^{VNC}\} \leq B' - V\mathbb{E}\{\mathcal{I}\}r^*, \tag{4.34}$$

where $B' = B + B_X + B_U + \nabla$ and ∇ is the finite residue of $X_{k-l}\mathbb{E}\{R_k^{ORSA} - Y_k^{ORSA}\}$.

Now we propose two important properties resulting from our algorithm.

Theorem 4.1.

1. the time average throughput rate achieved by our algorithm is within $\frac{B'}{V}$ of optimal rate achieved by ORSA

$$\lim_{t\to\infty} \inf \frac{1}{\mathcal{I}_t} \sum_{k=0}^{t-1} \mathbb{E}\{R_k^{VNC}\} \geq r^* - \frac{B'}{V}, \tag{4.35}$$

   where $B' = \frac{B+B_X+B_U+\nabla}{V}$.

2. Queue X is of finite length. It is upper bounded by a finite value $X_{max} = V + A_{max}$. A_{max} is introduced before as the maximum arrival packet number.

3. Queue U is of finite length. It is upper bounded by a finite value $U_{max} = X_{max} + \rho_\mu + 1$.

Proof. (1) Assume in ORSA, Y_k starts converging at frame 0 for convenience. Then summing from frame 0 to t yields:

$$\mathbb{E}\{L(\boldsymbol{Q}_k)\} \le B^{'}t - V\mathbb{E}\{\mathcal{I}\}tr^* + \sum_{k=0}^{t-1} V\mathbb{E}\{R_k^{VNC}\}. \tag{4.36}$$

Rearranging (4.36) and dividing by $\mathcal{I}_t$ on both sides will give

$$\sum_{k=0}^{t-1} V\frac{1}{\mathcal{I}_t}\mathbb{E}\{R_k^{VNC}\} \ge \frac{\mathbb{E}\{L(\boldsymbol{Q}_t)\}}{\mathcal{I}_t} - \frac{B^{'}t}{\mathcal{I}_k} + V\frac{\mathbb{E}\{\mathcal{I}\}tr^*}{\mathcal{I}_t}. \tag{4.37}$$

Then let $t \to \infty$ and divide V on both sides, we have

$$\begin{aligned} \lim_{t\to\infty}\sum_{k=0}^{t-1}\frac{1}{\mathcal{I}_t}\mathbb{E}\{R_k^{VNC}\} &\ge \lim_{t\to\infty}\frac{\mathbb{E}\{L(\boldsymbol{Q}_t)\}}{V\mathcal{I}_t} - \lim_{t\to\infty}\frac{B^{'}t}{V\mathcal{I}_k} + \lim_{t\to\infty}\frac{\mathbb{E}\{\mathcal{I}\}tr^*}{\mathcal{I}_t} \\ &\ge 0 - \frac{B^{'}}{V} + r^*, \end{aligned} \tag{4.38}$$

where new $B^{'} = \frac{B^{'}}{\mathbb{E}\{\mathcal{I}\}}$, and in many places it uses the fact that $\lim_{t\to\infty}\frac{\mathcal{I}_t}{t} = \mathbb{E}\{\mathcal{I}\}$. And since this inequality holds for all frames t, we can further throttle the bound by take $\inf$ at the left side of equation:

$$\lim_{t\to\infty}\inf\sum_{k=0}^{t-1}\frac{1}{\mathcal{I}_t}\mathbb{E}\{R_k^{VNC}\} \ge r^* - \frac{B^{'}}{V}, \tag{4.39}$$

which completes the proof for the first part of theorem 4.1.

(2) According to assumption, $X_1 = 0$, by induction, assume $X_k \le X_{max}$, then consider two kinds of situations: first, $X_k \le X_{max} - A_{max}$, then we have $X_{k+1} \le X_{max}$; second, $X_k \ge X_{max} - A_{max}$, then by our Traffic Control Policy,

$X_{k+1} \geq V + A_{max} - A_{max}$, no packet is allowed in, so we still have $X_{k+1} \leq X_{max} = V + A_{max}$, which finishes proof for second part of the theorem.

(3) Again we employ the induction to prove this part of the theorem. By assumption, $U_1 = 0$. Then we assume the theorem holds for U_k, that is $U_k \leq U_{max}$. Very similar to the proof for part 2, consider two cases: first, $U_k \leq U_{max} - 1$, then $U_{k+1} \leq U_{max}$ since maximum arrival packet is 1 in the virtual queue; second, $U_k \geq U_{max} - 1$, then $U_{k+1} \geq X_{max} + \rho_\mu$. According to Transmission Control Policy, $Y_k = 0$, therefore $C_k = 0$. So we still have $U_k \leq U_{max},\ \forall k \geq 1$, which completes the proof for the third part of the theorem. $\square$

Finally we propose the following theorem concerning the queue stability of both queues X_k and U_k

<u>Theorem</u> 4.2. With Algorithm 1, the queue X_k and $U_k,\ k \geq 1$ are *strongly stable*.

Proof. A queue is strongly stable if $\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{|Q(\tau)|\}$. Substituting the upper bound of X_k and U_k into the definition, we have:

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{|X_\tau|\} \leq \lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} X_m ax \quad = \quad X_{max} \leq \infty \tag{4.40}$$

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{|U_\tau|\} \leq \lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} U_m ax \quad = \quad U_{max} \leq \infty. \tag{4.41}$$

Therefore both X_k and U_k are strongly stable. Note that the Lyapunov Optimization theorem only defines the condition from mean rate stable, our algorithm however guarantees the strong stability. This is attributed to our Traffic Control policy. The intuitive behind this is that the traffic control policy is rather “dramatic”: admit either maximum arrival or admit nothing at all. $\square$

4.7 Discussion of Implementation of VNC

Usually, the system designer needs to make a decision on μ in Algorithm 1 given that the total deliver delay is bounded by $\mathcal{D}$ as in (4.3). If we choose a particular μ, we can decide V to be

$$(\mathcal{D} - \mu)\lambda - A_{max}. \tag{4.42}$$

For different applications, we can select different μ, and thus different V which controls the tradeoff between delay and throughput as stated in Theorem 4.1. For example, if the vehicles on the road are fast in general, therefore capable of delivering the packet fast, then we can choose a small μ. Thus (4.42) gives a larger V, therefore we have a better system throughput. This phenomenon in the context of vehicular networks is reasonable and resembles the influence of transmission channels in wireless communication. If we take the vehicle speed as the "channel quality" in conventional wireless communication and that faster vehicle speed means better "channel quality", the above phenomenon is actually saying "better channel quality results in better system performance".

Although the theoretical analysis of interactive relationship between μ and V given a total delivery delay bound $\mathcal{D}$ still lacks, the intuitive speculation is supported by the numerical analysis in the simulation chapter. However, it should be noted that the ambition to improve throughput should not justify the effort to diminish μ unlimitedly. For example, if μ is so small that even the fastest vehicle on the road fails to deliver the packet successfully to the receiver, then this μ is invalid and should not be chosen.

In sum, our alogrithm VNC can be implemented in the following specific way. First, collect statistic data of vehicles on road for a period of time. For instance, calculate the vehicle speed pattern to determine the speed range and their corresponding probabilities. Then, system designer will choose an appropriate μ according to the discussion given above. Also, system designer is responsible to study the demands of data traffic in order to decide on an appropriate total delivery delay $\mathcal{D}$ that adapts to system requirement. Then the value of V will be determined by (4.42). With these three system parameters set up, the VNC algorithm can then be carried out at each time frame of vehicle arrival.

Chapter 5

Simulation Results

In this section various numerical simulation results are given to bolster the theoretical analysis in previous chapters. First, we will describe the simulating scenario and configuration parameters associated with it. We assume that the distance between RSU R1 and R2 is 1000m; the speed of vehicles spans over a discrete range $[20, 40, 60, 80, 100, 120]$ and the unit we use is km/h; the probability of those speed levels is $[0.1, 0.1, 0.2, 0.4, 0.1, 0.1]$; the transmission delay μ is set to $200s$ and the maximum rate of delay constraint breach is $\rho_\mu = 0.1$; During each frame, the maximum arrival packets is $A_{\max} = 3$; Vehicles arrive in intervals the distribution of which is subject to an exponential distribution with parameter $\lambda = 1$; Both queues are initialized to 0 at the beginning.

In Fig. 5.1 we explore the long-term throughput rate achieved by implementing our algorithm. In this figure, we mainly study the behavior of V in the analysis of algorithm. Therefore we neglect the total deliver bound $\mathcal{D}$. Only the tradeoff between throughput and the queue stability is studied. First of all it can be seen that over a long time period (in our simulation it is run 500000 times), the throughput rate r converges, which unveils two layers of observations: one is the throughput

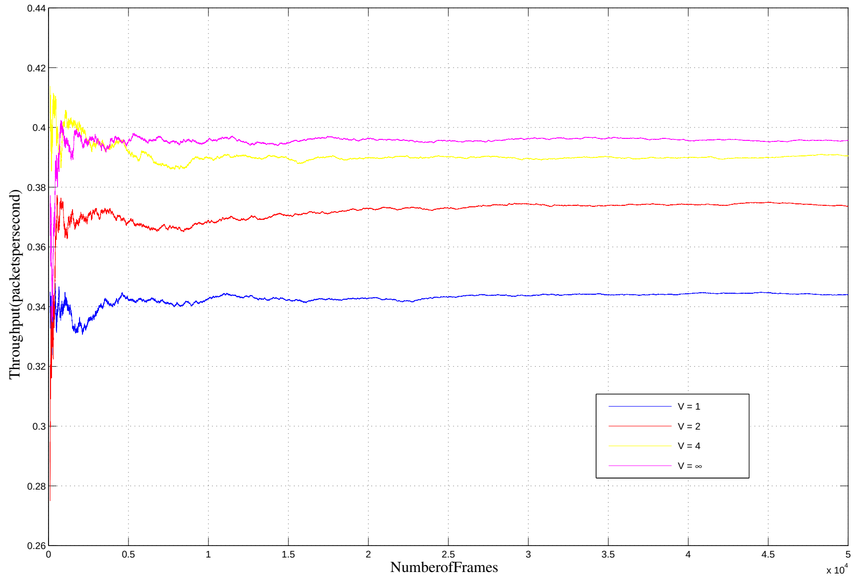

Fig. 5.1. Converging of Throughput Rate

converges relatively fast. Another observation is the leveraging effect of V: the larger the V, the more leverage is put on r and less on controlling the size of queues, therefore higher r can achieve. When $V = \infty$, *throughput* guaranteed by the random stationary optimal policy is achieved. Note that the small gap between $V = \infty$ and $V = 4$ is because the Lyapunov drift is small in our system, which means that it will diminish very fast along with the increase of V.

Fig. 5.2 depicts the relationship between the input rate λ and the optimal throughput rate. Again it can be observed from the graph that the general rule that larger V resulting in larger r still holds. However, r stops its growth at certain input rate and this threshold happens earlier for larger V. For example, in our experiment, the one with $V = 20$ stops increasing as early as $\lambda \approx 1.1$ packets per second, while the one with $V = 4$ halts its increasing until λ reaches 1.8. The larger the V, the faster the throughput rate reaches its optimal. This is the benefit of having an upper bound of arrival packets, which in our system is denoted as A_{max}. Whenever the

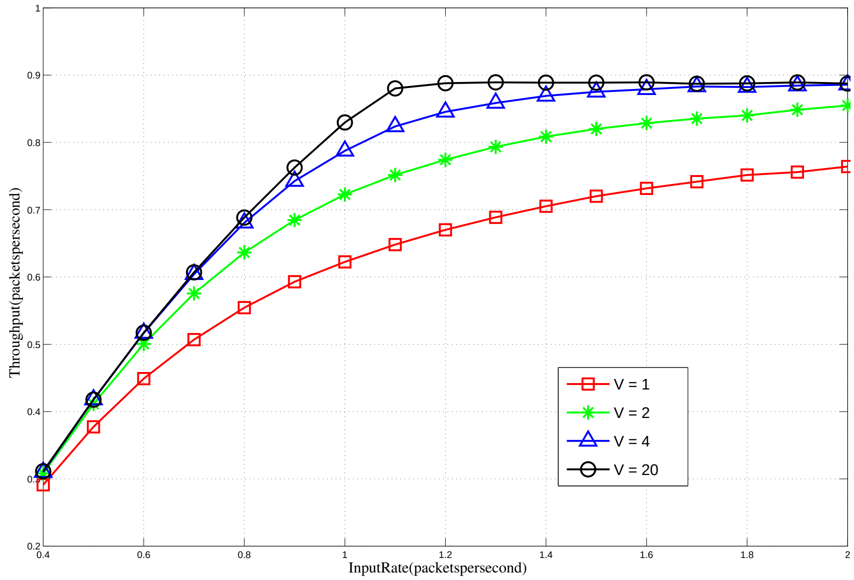

Fig. 5.2. Throughput rate Versus Input Rate

number of arrival packets exceeds A_{max}, the residual part is simply dropped.

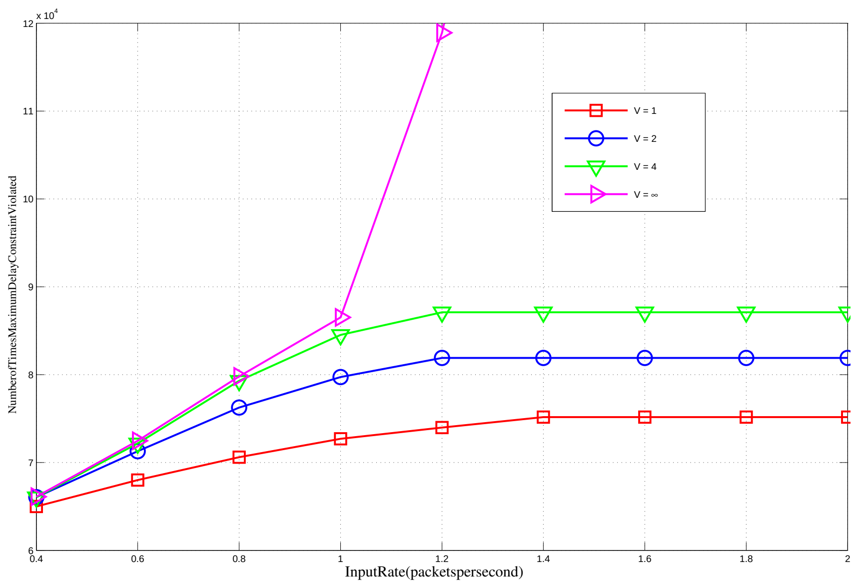

Fig. 5.3. Number of Violation of Transmission Delay Versus Input Rate

In Fig. 5.3 we examine the behavior of virtual queue. We plot the number of violation of transmission delay constraint according to the change of input rate λ. It is noted that when $V = \infty$, no concern for queue stability is spared any more. U therefore grows free of bound and the system network capacity is achieved and this value is roughly 1.2. And it can be seen that this is approximately the point where the throughput got saturated in Fig. 5.2. In Fig. 5.4 we continue exploring the

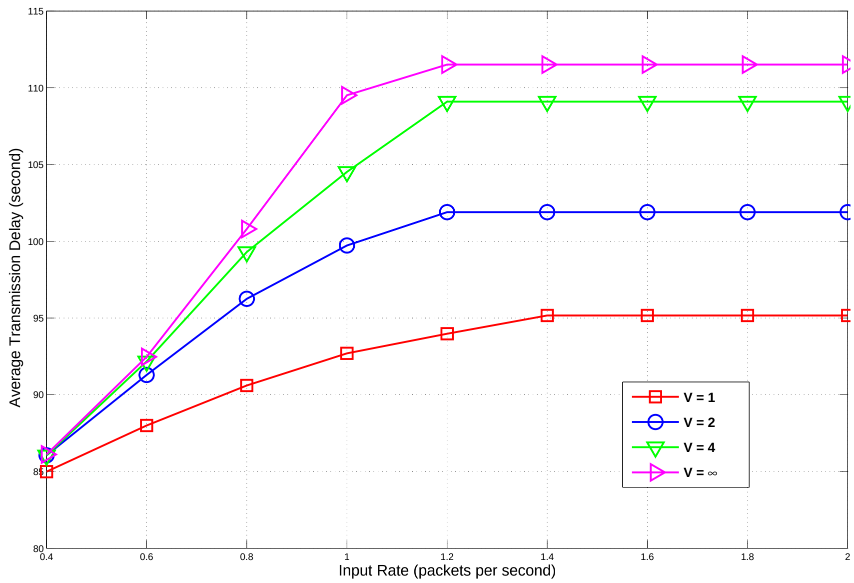

Fig. 5.4. Average transmission delay Versus Input Rate

tradeoff reflected through V. As we have seen from Fig. 5.2 and Fig. 5.3, increasing V will result in increasing throughput and also the number of dropped packets at the destination R2. In Fig. 5.4, we show the average transmission delay experienced by delivered packets. As predicated by the analysis, the average transmission delay, which is affected by V, increases with increasing V. By comapring this figure with the Fig. 5.2 we discover that V is a tradeoff parameter in the system.

In Fig. 5.5, the CDFs for the queueing delay, transmission delay, and the total delivery delay are given to validate that our algorithm VNC bounds them successfully. In the simulation, we set speed of vehicles as 90, 95, 100, 105, 110, 115, 120

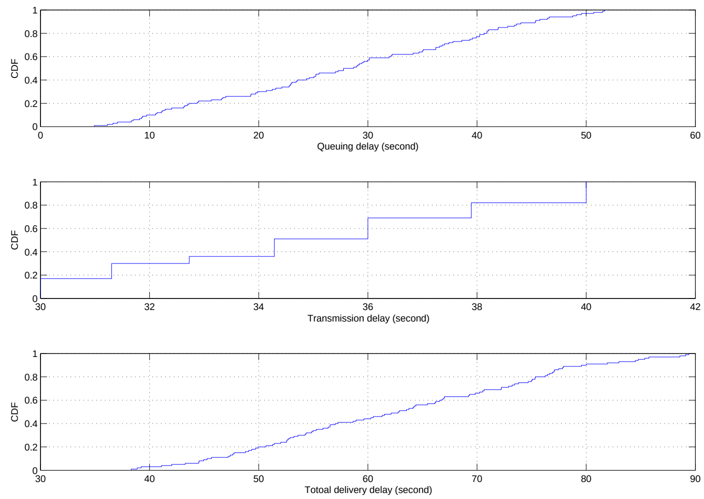

Fig. 5.5. Cumulative distribution functions(CDF)of queueing delay, transmission delay, and total delivery delay respectively

km/h with uniform probability. We set $\mathcal{D} = 100$s, $\mu = 45$s and queueing delay bound as 55s. From the result of Fig. 5.5 it can be observed that both queueing delay bound and transmission delay bound are achieved. And finally the total delivery is successfully bounded below $\mathcal{D} = 100s$. Furthermore, our notion about the conservativity of our algorithm is verified also by Fig. 5.5.

In Fig. 5.6 we show the maximum transmission delay, queueing delay, and therefore total delivery delay experienced by each successful packet for different μ and V as in (4.3). In the simulation, we set the total delivery delay bound $\mathcal{D} = 100s$ and let μ take values from $35, 45, 55$s. Consequently, the values of V is calculated from (4.3) and Theorem 4.1 to be $62, 52$ and 42 respectively. It can be observed from Fig. 5.6 that the transmission delay is well bounded below μ and the total delivery delay is also bounded under the given $\mathcal{D} = 100s$. It can also be found that our alogirthm results in a conservative delivery delay bound. There is relatively large room for us to maneuver the μ. This is due to our special treatment for de-

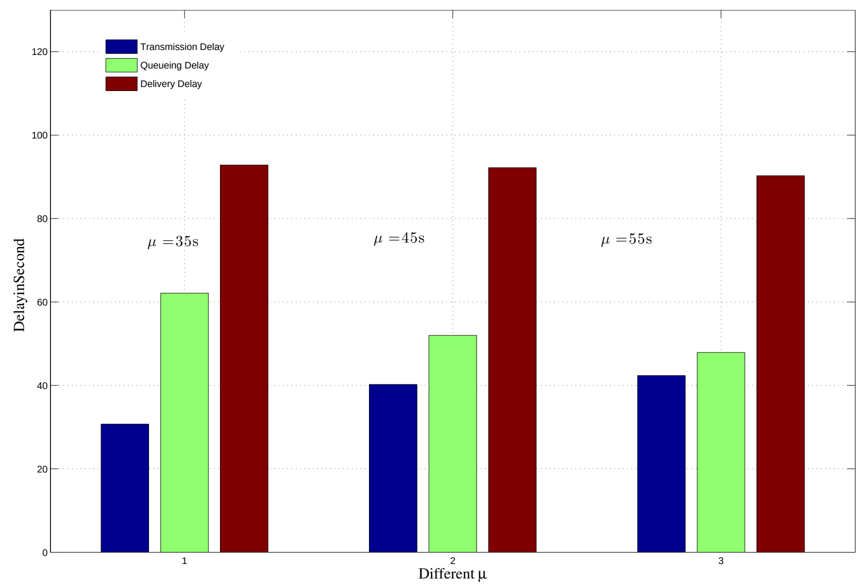

Fig. 5.6. Maximum delay for different μ

coupling two different delays and bound them separately. Most vehicles satisfying the transmission delay actually deliver the packet faster than μ. Also, our bound on the queueing delay $X_{\max}$ is also conservative. In real applications, we can always expect a smaller queue size instead.

In Fig. 5.7 we examine the throughput performance corresponding to three cases in Fig. 5.6. It shows that the larger the V, the better performance we will get. This is guaranteed by Theorem 3.4. This simulation provides a guidance to us on how to choose μ and V given the total delivery bound $\mathcal{D}$ is required. Generally, in situations where the vehicles travel in a faster speed, we can choose relatively smaller μ because vechiles on the road are more capable to deliver the packet within the transmission delay. And then we can choose a larger V to achieve higher throughput, which is reflected in Fig. 5.7. Although the theoretical analysis of the interaction between μ and V given a total delivery delay $\mathcal{D}$ still lacks, an intuitive explanation however can be given: Similar to the channel quality for conventional

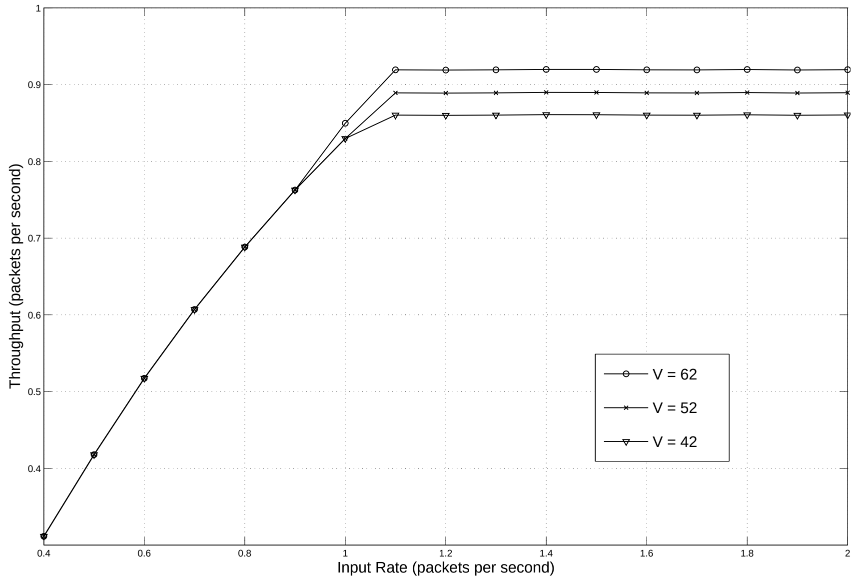

Fig. 5.7. Throughput for $V = 62, 52, 42$

wireless channels, the vehicular speed in the Vehicular Network plays the same role of "channel quality". The faster the vehicles move, the better the "channel quality" is, therefore the larger throughput can be achieved.

Finally, Fig. 5.8 compares the throughput performance of our algorithm with several heuristic ones. The heurisitc algorithm 1 (HA1) works likes the following: at the RSU side, a finite buffer of size $B = V$ is used to hold incoming packets and the packets will be refused admission if the finite buffer is full; the arrival vehicles will be chosen according to the criteria that the vehicle can deliver the packet within the delay constraint. And the HA2 is a simple one: whenever a vehicle arrives, RSU relays the packet to the vehicle. Note that we do not consider the situation where packets already violate the total delay constraint while still waiting in the queue. The justification behind this assumption is that once packets are admited into the queue, it is difficult to actually calculate the waiting time in the queue. It will not only complicates the implementation of queues, but also violates the

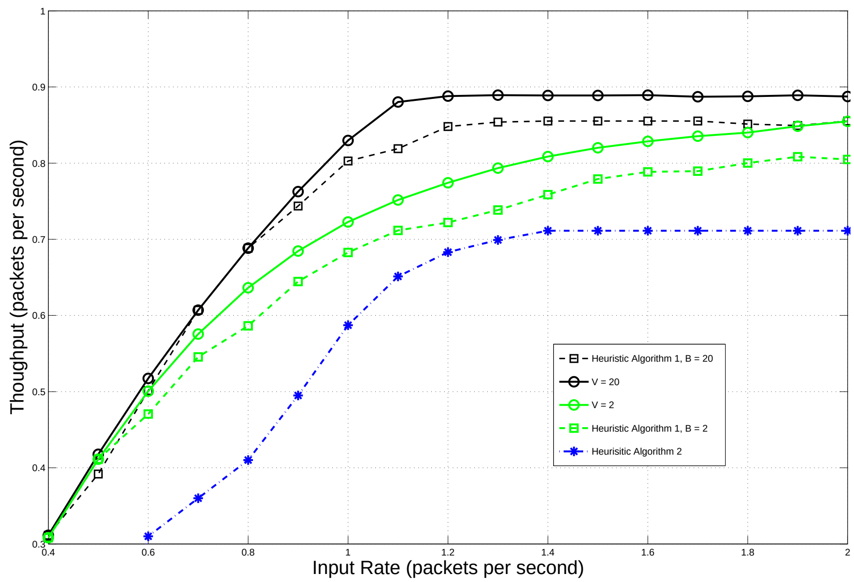

Fig. 5.8. Comparison with heuristic algorithms

separation of different layers in the network. From Fig. 5.8 we discover that HA2 will achieve the lowest throughput despite the fact that it delivers the packet to each passing vehicle. The intuition behind this low throughput is that HA2 wastes the opportunities. Especially in the light traffic region, it is of large probability that it will miss the vehicles of decent speed by transferring packets to slower ones. Our algorithm out-performs the HA1 regardless of the choice of buffer capacity.

Chapter 6

Conclusions and Future Work

Vehicular communication network has become a popular topic in recent years. It not only enhances the road safety by providing ways to communicate emergency information, but also improves the wireless connection coverage on the roads and in the remote areas. In less populated regions and poor countries, cheap vehicular communication sets and road side units provide a viable substitution of expensive mobile communication infrastructures. The implementation of such networks will effectively benefit those regions and countries. Vehicular network is a type of Delay Tolerant Network, which requires special treatment of delay. Unlike other sorts of communication networks, the main packet carrier, vehicles, are not always available in time, thus causing random behavior of packet delivery process. Another challenge of such network is the difficulty of balancing between the system reward (throughput) and delay. The problem presented in this thesis is motivated by these two concerns.

In this thesis, we consider a vehicular network where a data post (RSU) selectively employs the passing vehicles to relay the packets to the destination, which is another data post down the road. The packets to be delivered demand satisfaction of certain delay constraint to maximize quality of service. Also, the packets are not supposed to saturate the waiting queue for the sake of maintainability of the data post. On top of the requirement stack, we aim to achieve an optimal time-average throughput under the assumption that all other system constraints are not violated. To tackle this problem, we adopt the Lyapunov Optimization technique to construct the problem and solve the problem by deriving the Lyapunov drift-plus-penalty expression. At the end, an algorithm is developed to push the time-average throughput arbitrarily close to the optimal value, which is proved to be achieved by a random stationary optimal policy. At the same time the transmission delay of the packets from one data post to the next is controlled to be within certain threshold in some probability that the system can tolerate. Finally, the queue is strongly stable as long as the algorithm is carried out. The advantage of the algorithm developed in this thesis is that it is implemented in each decision epoch of the system and only simple calculations are involved. It generates much less calculation overhead compared to other alternative methods such as Markov decision technique. Yet, its result still falls as near to the optimal value as we can control. In the end, numerical results are given to verify our algorithm. It is seen that we do obtain a stable time average throughput rate and also the queues are well controlled under designed level.

The future directions of work include: In the thesis, the vehicle arrival times are assumed to be independent from the backlog status and system decisions. Though this greatly simplifies the development and captures the reality, it omits the case in which car arrival process is subject to more complex distribution than simple Poisson distribution. Consider a multi-lane road environment where cars travel in different lanes. The data post now is blessed with the option to employ more than one vehicle to do its job. Resource allocation then can be added to such situation. For example, the data post faces the option whether to forward packet to all cars arrival at the same time or select the fastest one among them. Another dimension of the problem becomes emerging when the power of the data post spends in transmitting packets to vehicles is limited and therefore demands optimization; In this thesis, we totally neglect the channel quality between the vehicles and the road side post. For instance, we can scale the channel quality into different levels, and for each level different decision Y_k will be made instead of only two choices are available which is the case in the thesis. For poor channel, transmission from the data post to the vehicle could also introduce noticeable delays that can be considered as part of the optimization target.

References

[1] V. Ramaiyan and E. Altman and A. Kumar, "Delay optimal scheduling in a two-hop vehicular relay network," *Journal Mobile Networks and Applications*, vol. 15, no. 1, pp. 97–111, Feb. 2010.

[2] R. Urgaonkar and M.J. Neely, "Opportunistic scheduling with reliability guarantees in cognitive radio networks," *Opportunistic scheduling with reliability guarantees in cognitive radio networks*, vol. 8, no. 6, pp. 766–777, Jun. 2009.

[3] L. Huang and M.J. Neely, "Delay reduction via lagrange multipliers in stochastic network optimization," *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 842–857, Apr. 2011.

[4] M. Khabazian and M. Ali, "A performance modeling of connectivity in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2440–2450, Jul. 2008.

[5] S. Panichpapiboon and W. Panichpapiboon, "Connectivity requirements for self-organizing traffic information systems," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3333-3340, Nov. 2008.

[6] H. Wu, R.M. Fujimoto, G.F. Riley, and M. Hunter, "Spatial propagation of information in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 58. no. 1, pp. 420–431, Jan. 2009.

[7] M. Grossglauser and D.N.C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477-486, Aug. 2002.

[8] S.A. Jafar, "Too much mobility limits the capacity of wireless ad hoc networks," *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 3954-3965, Nov. 2005.

[9] M.J. Neely and E. Modiano, "Capacity and delay tradeoffs for ad hoc mobile networks," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1917-1937, Jun. 2005.

[10] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78-83, Jan. 2004.

[11] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910-1922, May. 2008.

[12] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

[13] M.J. Neely, E. Modiano, and C.E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89-103, Jan. 2005.

[14] M.J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915-2934, Jul. 2006.

[15] M.J. Neely and R. Urgaonkar, "Cross-layer adaptive control for wireless mesh networks," *Ad Hoc Networks*, vol. 5, no. 6, pp. 719-743, Aug. 2007.

[16] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201-220, Feb. 2005.

[17] J.N. Laneman and G.W. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2415-2425, Oct. 2003.

[18] G.J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41-59, 1996.

[19] A. Goldsmith, S.A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 5, pp. 684-702, Jun. 2003.

[20] D. Gesbert, M. Shafi, D. Shiu, J.P. Smith, and A. Naguib, "From theory to practice: an overview of MIMO space-time coded wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 281-302, Apr. 2003.

[21] A.B. McDonald and T.F. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1466-1487, Aug. 1999.

[22] R. Fan, H. Jiang, Q. Guo, and Z. Zheng, “Joint optimal cooperative sensing and resource allocation in multichannel cognitive radio networks,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 722-729, Feb. 2011.

[23] L. Laim, E.H. Gamal, H. Jiang, and H.V. Poor, “Cognitive medium access: exploration, exploitation, and competition,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 239-253, Feb. 2011.

[24] T. Spyropoulos, T. Turletti, and K. Obraczka, “Routing in delay-tolerant networks comprising heterogeneous node populations,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1132-1147, Aug. 2009.

[25] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, “A stable routing protocol to support its services in vanet networks,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337-3347, Nov. 2007.

[26] D.P. Bertsekas and P.G. Gallager, *Data Networks*, Prentice Hall, 1992.

[27] M.J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan and Claypool Publishers, 2010.

[28] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005

[29] A. Abdrabou, B. Liang, and W. Zhuang, “Delay analysis for a reliable message delivery in sparse vehicular ad hoc networks,” in *Proc. IEEE GLOBECOM*, Anaheim, CA, Dec. 2010, pp. 1-5.

[30] J. Chen and V.C.M. Leung, “Applying active queue management to link layer buffers for real-time traffic over third generation wireless networks,” in *Proc. IEEE WCNC*, New Orleans, LO, Mar. 2003, pp. 1657-1662.

[31] A. Abdrabou and W. Zhuang, "On a stochastic delay bound for disrupted vehicle-to-infrastructure communication with random traffic," in *Proc. IEEE GLOBECOM*, Honolulu, HA, Dec. 2009, pp. 1-6.

[32] X. Yang, L. Liu, N.H. Vaidya, and F. Zhao, "A vehicle-to-vehicle communication protocol for cooperative collision warning," in *Proc. MOBIQUITOUS*, US, Aug. 2004, pp. 114-123.

[33] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Vehicle-to-vehicle safety messaging in DSRC," in *Proc. ACM VANET*, Philadelphia, PA, 2004, pp. 19-28.

[34] W. Zhao, M. Ammar, and R. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. ACM MobiHoc*, Roppongi Hills, Tokyo, 2004, pp. 187-198.

[35] M.J. Neely, E. Modiano, and C.P. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 1723-1734.

[36] M.J. Neely, "Stochastic optimization for Markov modulated networks with application to delay constrained wireless scheduling," in *Proc. IEEE CDC/CCC*, Shanghai, China, Dec. 2009, pp. 4826-4833.

[37] K.S. Beyer, J. Goldstein, R. Ramakrishan, and U. Shaft, "When Is Nearest Neighbor Meaningful? ," in *Proc. ICDT*, London, UK, 2009, pp. 217-235.

[38] S.R. Das, R. Castaneda, J. Yan, and R. Sengupta, "Comparative performance evaluation of routing protocols for mobile, ad hoc networks," in *Proc. IEEE ICCN*, Lafayette, Louisiana, Oct. 1998, pp. 153-161.

[39] T.K. Moseng, O. Kure, "QoS architecture in ad hoc networks," in *Proc. IEEE CHINACOM*, Shanghai, China, Aug. 2007, pp. 1010-1014.

[40] D. Niyato, P. Wang, and J. Teo, “Performance analysis of the vehicular delay tolerant network,” in *Proc. IEEE WCNC*, Budapest, Hungary, Apr. 2009, pp. 1-5.

[41] “The network on Wheels (NOW) project,” NOW, June 2009.

[42] “The cooperative vehicles and road infrastructure for road safety (SAFE-POST) project,” SAFESPOT, 2012.

[43] “The wireless access in vehicular environment (WAVE) project,” IEEE802.11p, 2012.

[44] “The Mobile Ad Hoc Networks (MANET) project,” MANET, 2012.