# Cross Layer Peer-to-Peer Video Sharing for Vehicle Ad-Hoc Networks (VANETs)

by

Hengheng Xie

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

# Abstract

Accompanying the increasing interest on Vehicle Ad-hoc Network (VANET), there is a request for high quality and real-time video streaming on VANET, for safety and infotainment applications. Video Streaming on VANET faces extra issues, comparing to the Mobile Ad-hoc Network (MANET), like the high dynamic topology. However, there are also benefits in VANET, like large buffer and battery capacity, predictable motion of vehicles and powerful CPU and GPU. Video streaming on VANET can be highly improved by these features. However the poor performance of wireless networks is an critical issue for video streaming in VANET. The high packet loss probability of wireless networks significantly reduces the quality of the transmitted video. An error recovery process is proposed in our research for high quality and real-time video streaming in VANET, which is call Multi-path Error Recovery Video Streaming (MERVS). The performance improvement of wireless networks is also considered in our research. The cross layer technique is adopted in our research, in order to increase the accuracy on the network condition monitoring and to guarantee the fairness on network resource distribution. Cross layer protocols on both the Media Access Control (MAC) layer and the Network layer are proposed to improve the performance collaboratively. The contribution of my researches are: 1) I proposed a MERVS, which provides high quality and real-time video streaming; 2) several improvement techniques are also designed to improve the performance of MERVS; 3) simulation results verifies that MERVS can have a higher quality on transmitted video comparing to the existing protocols in an acceptable delay.

# List of Publications

The following listed are the publications related to this thesis.

**Submitted/Accepted Journal Papers**

1. Xie, H.; Boukerche, A.; Loureiro, A., "MERVS: A Novel Multi-channel Error Recovery Video Streaming Scheme for Vehicle Ad-hoc Networks," *IEEE Transactions on Vehicular Technology* , vol.PP, no.99, pp.1,1, 2015.

2. Xie, H.; Boukerche, A.; Loureiro, A., "A Multipath Video Streaming Solution for Vehicular Networks with Link Disjoint and Node-disjoint," *IEEE Transactions on Parallel and Distributed Systems* , vol.PP, no.99, pp.1,1, 2014.

3. Xie, H.; Boukerche, A., "TCP-CC: cross-layer TCP pacing protocol by contention control on wireless networks", *Springer Wireless Network*, vol 21, no.4, pp.1-18, 2015.

**Published Conference Papers**

1. Hengheng Xie, Azzedine Boukerche, Robson De Grande, F. Richard Yu. Towards a Distributed TCP Improvement through Individual Contention Control in Wireless Networks. *IEEE ICC 2015*. Accepted.

2. Hengheng Xie, Azzedine Boukerche, Antonio A.F. Loureiro. Towards TCP Optimization in Wireless Networks by A Frame based Cross Layer Routing Metric. *IEEE Global Communications Conference (GLOBECOM13)* 2013: 4603 - 4608

3. Hengheng Xie, Azzedine Boukerche, Antonio A.F. Loureiro. TCP-ETX: A Cross Layer Path Metric for TCP Optimization in Wireless Networks. *IEEE International Conference on Communication (ICC13)* 2013: 3597 - 3601

4. Hengheng Xie, Azzedine Boukerche, Mohammed Almulla. A Novel Cross Layer TCP Pacing Protocol for Multi-hop Wireless Network. *IEEE Wireless Communications and Networking Conference (WCNC13)* 2013: 1428 - 1433

5. Mohammed Almulla, Hengheng Xie, Azzedine Boukerche, Abdelhamid Mammeri: A Fluid Dynamics based Distributed Cross Layer Quality Assurance Algorithm for Multi-hop Transmission in Wireless Networks. *9th ACM International Symposium on QoS and Security for Wireless and Mobile Networks* 2013: 107-114

6. Hengheng Xie, Azzedine Boukerche, Richard W. Pazzi. A Novel Collision Probability based Adaptive Contention Windows Adjustment for QoS fairness on Ad Hoc Wireless network. *IEEE Global Communications Conference (GLOBECOM12)* 2012: 5710-5715

7. Hengheng Xie, Richard W. Pazzi, Azzedine Boukerche. A Novel Cross Layer TCP Optimization Protocol over Wireless Network by Markov Decision Process. *IEEE Global Communications Conference (GLOBECOM12)* 2012: 5945-5950

**Acknowledgements**

First and foremost I would like to express my sincere gratitude to my supervisor Prof. Azzedine Boukerche for the continuous support of my Ph.D study and research, for his financial support, patience, motivation, enthusiasm, and invaluable guidance and advice. His guidance helped me in all the time of research and writing of this thesis. I feel very privileged to have had the opportunity to learn from, and work with him. He not only has always believed in me, he has always been there whenever I have needed advice. For that, I am forever grateful.

Furthermore, I would like to thank all my colleges throughout all the years who offered help in many aspects of what is presented here.

Last but not the least, I would like to thank my family who stood beside me, provided support and encouragement: my parents, Jingfen Xie and Qi Zhong; my wife, Qiyin Zhang.

# Contents

# List of Tables

# List of Figures

# Glossary of Terms

**ACK** Acknowledgement

**AODV routing** Ad hoc On-Demand Distance Vector routing

**CTS** Clear to Send

**CDMA** Code Division Multiple Access

**CW** Contention Window

**DCF** Distributed Control Function

**DSDV routing** Destination-Sequenced Distance Vector routing

**ETX** Expected Transmission Count

**GPS** Global Positioning System

**HTTP** Hypertext Transfer Protocol

**LTE** Long Term Evolution

**MAC layer** Media Access Control layer

**MANET** Mobile Ad-hoc Network

**MDP** Markov Decision Process

**MPEG** Moving Picture Experts Group

**OSI model** Open System Interconnection model

**PSNR** Peak Signal-to-Noise Ratio

**QoS** Quality of Service

**RSU** Road Side Unit

**RTP** Real-time Transport Protocol

**RTS** Request to Send

**RTT** Round Trip Time

**SSIM** Structural SIMilarity

**TCP** Transmission Control Protocol

**TDMA** Time division multiple access

**UDP** User Datagram Protocol

**VANET** Vehicle Ad-hoc Network

**V2I** Vehicle to infrastructure connection

**V2V** Vehicle to vehicle connection

**WCDMA** Wideband Code Division Multiple Access

# Chapter 1

# Introduction

In recent years, accompanying the innovation of mobile devices, the number of wireless services explodes dramatically [1]. Some of the mobile devices are integrated with the vehicle, which generates the Vehicle Ad-hoc Networks (VANETs). The concept of VANETs is part of the concept of Mobile Ad-hoc Networks (MANETs). Different from moving at random in MANETs, the motion of vehicle is predictable in VANETs; this is because the vehicle is restricted by the paved roads. The buffers and batteries of vehicles can be considered as large enough, because of the power provided by the engine and the possibility for extra cache. The Central Processing Unit (CPU) and Graphics Processing Unit (GPU) can also be considered more powerful than the traditional MANET nodes. These features enhance the capability of VANETs, comparing to the traditional MANETs. However, VANETs encounter some new problems, the high movement speed. The high movement speed of the vehicles decreases the time of the connection between the vehicles and the infrastructure (V2I), which lowers the chance for information exchange. The connections between the vehicle and the vehicle (V2V) are also fragile, because of the various speed of the vehicles.

The development of VANETs improves people's life by providing services on safety and infotainment. Traffic condition review can be provided on VANETs for avoiding the traffic jam for driving, especially for ambulances and fire trucks. Safely overtaking is also possible by enlarging the vision of the drivers. Also online movie and video conference are available inside the vehicle, when you are on the road. The main scenario considered in this research is that when a vehicle drives across the downtown area, the vehicle obtains a connection to the access point (AP). The vehicle can download the video to its buffer in order to maintain the video playing when it is out of the town. This downloaded

video is also can be shared among other vehicle or access point for other users. The key technique of these services are video streaming, which offers high quality and real-time video streaming services on VANETs. The issues of video streaming over VANETs is the poor performance of wireless networks, comparing to that of wired networks, which needs to be improved. One cause of its poor performance is the unstable environment of wireless networks, which includes fading, interference and collision. Another reason is that the existing network protocol and software are mainly designed and implemented based on the features of the wired networks. The poor performance of VANETs is also because of its short connection time caused by the high movement speed. Therefore, to well support the wireless environment, the existing protocol and software should be modified, or new protocol and software should be proposed.

## 1.1 Motivation

The challenges of video streaming on VANETs are: how to transmit the high quality video and how to reduce the time for video streaming. The quality of video streaming on wireless networks is very low, because of the high packet loss probability caused by the fading, interference and collision. The quality of the transmitted vide is measured by Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM). The quality of video streaming on wireless network is very difficult to control. It is even worse in VANETs, because of the fast changing network topology caused by the high movement speed. Most of the video streaming protocols do not have reliability techniques, because reliability is considered as useless in video streaming and some packet loss in video streaming is acceptable. However, certain reliability on important part of the video should be provided in order to increase the quality of the received video. TCP can be used to offer reliability in data transmission. However, its poor performance on wireless networks limits its usage on video streaming.

The real time video streaming is very sensitive to delay and jitter. The limited capability of wireless networks increase the time for video streaming, which eventually causes extra delay and jitter to video streaming. To improve the performance of wireless networks, two questions need to be consider: how to increase the throughput and how to improve the QoS fairness of the wireless networks. The first question attempts to create extra network resources, which is known as the performance management issue of Service Assurance. The second question attempts to optimize the utilization of the network resources, which is know as the Quality of Service (QoS) management issue of the Service

Assurance. Many problems on wireless networks arise accompanying these topics: How to improve the network performance with a high rate of packet loss probability? What is the optimal performance of a wireless network? How to reach the optimal performance? How to minimize the contention level of the wireless networks? How to achieve fairness over high contention level networks? However, solutions for problems addressed above poses great challenge, because of the Open System Interconnection (OSI) Reference Model [128], which isolates different network functionalities and simplifies the network structure of each layer. Network variable becomes hidden to certain functionalities, which increases the difficulty for network monitoring and performance estimation.

This research intends to provide a solution for video streaming over VANETs on IEEE802.11, which has lower cost and higher data rate comparing to other network protocols. The challenges of IEEE802.11 is its small transmission range, which leads to even shorter connection time in VANETs.

## 1.2 Problem Statement

This research concentrates on the challenges about the video streaming protocols and wireless improvement protocols of VANETs in the following aspects:

- Distributed Solution and Self-Organization: The communication cost maintenance is high for centralized system. Because of the limited resource of wireless networks, it is critical to reduce the overhead of the protocol. Self-Organized system is a better choice than the centralized system, in terms of low communication cost.

- High quality Video Streaming: The quality of transmitted video in wireless networks is very low, because of the high packet loss probability. The acceptable PSNR value for video streaming through wireless transmission are 20dB to 25dB [58, 102], provided the bit depth is 8 Bit. The typical PSNR value for image and video compression are between 30dB and 50dB [5]. If the PSNR can be increased for wireless video transmission to 30dB or higher, it is able to offer higher video quality service.

- Real-time Video Streaming: The main issue of real-time video streaming on wireless networks is the limited network resource. In order to offer better real-time video streaming over VANETs. The performance of wireless networks should be optimized.

- Performance Improvement: One of the main issues of wireless networks is its low performance. It is known that most of the multimedia software are implemented with Hypertext Transfer Protocol (HTTP) [37] and Transmission Control Protocol (TCP) [85], rather than Real-time Transport Protocol (RTP) [94] and User Datagram Protocol (UDP) [84], as presented in [69]. In order to reuse this software in wireless networks, the performance of TCP in wireless network will be one of the most critical issues. As it is known, most of multimedia software is sensitive to network performance; this is because the high requirements of the audio and video transmissions. This issue can be resolved by a performance Improvement on the TCP protocol or other protocols.

- Performance Monitoring and Estimation: Performance monitoring and estimation are one of the prerequisites for performance improvement. The improvement protocol intents to reach the optimal state of the networks, in which the optimal state of the networks is difficult to estimate in the OSI model. The strict rules of OSI model limits the communication interface between functionalities, in order to simplify the implementation of the network structure. However, the OSI model blind some protocols about the dysfunctionality of the lower layers. For example, TCP treats all segment loss as a result of congestion; this is because it believe the lower layer will deliver the packets properly. However, in high contention wireless networks, the probability of collision and signal interference is very high; and, it becomes the main cause of the packet loss rather than congestion on TCP. An approach for performance monitoring, which can pass through the limitation of OSI model, are required for performance improvement.

- QoS Assurance: Besides the performance improvement, which generates additional resource, another approach is to manage the network utilization. How to allocate the limited resource is another main issue of the performance improvement in wireless networks. QoS assurance attempts to reserve the resource for the services requests; and, the resource are shared based on certain mechanism. A good QoS assurance algorithm should be proposed to optimize the wireless resource utilization.

- Fairness: The mechanism for resource sharing should be optimized based on the concept of fairness. If the overall requested resource is lower than the network capacity, the network system should satisfy as many requests as possible; and,

assigning unnecessary resource to a request should be avoided. The best case is that each service obtains the exact resource as requested; and, none of the service is starving.

### 1.2.1 Challenges of Video Streaming on VANETs

**Network Feasibility**

| Protocol | Data Rate | Transmission Distance |
|---|---|---|
| IEEE 802.11a | 6-54 Mbit/s | 35 m (indoor), 120 m (outdoor) |
| IEEE 802.11b | 1-11 Mbit/s | 35 m (indoor), 140 m (outdoor) |
| IEEE 802.11g | 6-54 Mbit/s | 35 m (indoor), 140 m (outdoor) |
| IEEE 802.11n | 7.2-72.2 Mbit/s | 7 m to 140 m |
| IEEE 802.11p | 3-27 Mbit/s | 7 m to 140 m |
| WCDMA | 384 kbit/s | 12 km |
| CDMA | 144 kbit/s | 6 km |
| TDMA | 14.4 kbit/s | 35 km |
| LTE | 100 Mbit/s | 5-30 km |

Table 1.1: Comparison of wireless networks

Before this research, there are few questions should be considered: which network infrastructure will be used? and which video compression technique will be used? Table 1.1 lists the parameters of different network protocols. The feature of WI-FI techniques from IEEE802.11a-IEEE802.11n is high bandwidth but small coverage range. The 3G techniques, like CDMA and WCDMA, have large coverage range but small bandwidth. The 4G technique, like LTE, have both high bandwidth and large coverage, but the cost of 4G networks is very high.

To make a decision on the choice of network protocols in VANETs, it should also consider the parameters of VANET scenarios. There are two main kinds of scenarios in VANETs: the urban scenario and the highway scenario. In the urban scenario, vehicles are moving in the urban area, and the support of roadside infrastructure is available. In the highway scenario, vehicles are moving much faster than the vehicles in the urban scenario, and the roadside infrastructure is not available in most of the cases. The vehicle speed limit in different road is as follow:

- Urban: 50 km/h = 13.9 m/s
- Rural (local highway): 80 km/h = 22 m/s
- Highway: 100 km/s = 27.8 m/s

The connection of 3G and 4G networks are stable, because the coverage range of 3G and 4G networks are very large. The feasibility of IEEE 802.11 in VANETs needs to be confirmed due to its small coverage. For a V2I connection, in principle, a vehicle has a connection for 17.3 s when passing a roadside infrastructure in the urban scenario. In theory, this time is enough for an IEEE 802.11g network to transmit a 1-minute video in HDTV quality and a 10-minute video in VCD quality in full speed. Usually, the roadside infrastructure will be set up at each intersection in the urban scenario, and the vehicle can use other vehicles as intermediate nodes to extend the connection range of the roadside infrastructures. Therefore, an IEEE 802.11 network should be feasible for V2I connections in urban scenarios. Because the roadside infrastructure is not common yet in a highway scenario, the V2I connection in a highway scenario is not the main focus of this research.

In the following, I present some typical data rates for video applications:

- 128 to 384 kbit/s: business-oriented videoconferencing quality using video compression;
- 1.5 Mbit/s: VCD quality, using MPEG1 compression;
- 3.5 Mbit/s: Standard-definition television quality, with bit-rate reduction from MPEG-2 compression;
- 8 to 15 Mbit/s: HDTV quality, with bit-rate reduction from MPEG-4 AVC compression;
- 40 Mbit/s: Blu-ray Disc, using MPEG2, AVC or VC-1 compression;

In V2V connections, the famous two second rule while driving is a rule providing a recommendation of safe following distance at any speed. A driver should stay at least two seconds behind the vehicle directly in front of the current vehicle. Therefore, the recommendation distance between vehicles in urban scenario is 28m, and the recommendation distance between vehicles in highway scenario is 56m. The IEEE802.11 is feasible in both the urban scenario and the highway scenario on V2V connections.

Therefore all network protocols in Table 1.1 are feasible in both V2I and V2V connections. About the data rate, the 3G networks are only feasible for low quality video streaming, like videoconferencing. Only 4G networks and IEEE802.11 meet the requirements of high quality video streaming. By considering the cost of the techniques, it is decided to conduct the research on IEEE802.11 in VANETs. Because the lack of short range transmissions in highway scenario, it is only considered the urban scenario in this research. Meanwhile, IEEE802.11p is added the Wireless Access in Vehicular Environments (WAVE), which provides Intelligent Transportation Systems (ITS) on IEEE802.11 for vehicle communication.

### Video Compression

In this work, I consider the H.264/MPEG-4 standard. The H.264/MPEG-4 AVC compression technique [117] includes both temporal scalability and spatial scalability on video compression. Its Scalable Video Coding (SVC) extension is able to offer layered structure video streaming on different quality for different network capacities.

Temporal scalability indicates that the frame rate of the video is changeable. This feature is offered by the MPEG-4 compression technique. It compresses different types of frames as indicated below:

- I-Frame: The Intra-frame contains the full information about the image and can be decoded independently.
- P-Frame: The Predicted-frame improves the compression by removing the temporal redundancy in the video. P-frame contains only the difference of the I-frame or P-frame directly in front of the current P-frame.
- B-Frame: The Bidirectional-frame has to be decoded by two reference frames, where one is from the previous frame and the other one is from the future frame.

By eliminating the temporal redundancy information in the video, MPEG-4 reduces the size of the video, which is needed to be transmitted. Any received P-frame and B-frame will enhance the frame rate of the video.

Spatial scalability is supported by the layered structure video compression in the SVC extension. Each frame is split into several layers in SVC, where the extra layers can improve the resolution of the base layer. H.264/MPEG-4 AVC SVC is suitable for video streaming in wireless networks, because its quality scalability (temporal and spatial) allows video streaming in networks with different qualities.

### 1.2.2 Challenges of Performance Improvement

Because most of the wireless protocol is adopted from wired networks. It leaves us some challenges to improve the network protocols in wireless networks, as follow:

- OSI Model: The strict rules of OSI model limits the capability of each functionality. Different functionalities are impossible to cooperate with each other. High level network functionalities is impossible or slow to react to the network condition changes.
- The Implicit Assumptions of Existing Protocols: Some protocols are designed based on certain implicit assumption of the network condition. One example is mentioned above about that TCP treats all segment loss as a result of congestion. The reason of this TCP rule is that TCP assumes the lower layer network quality is high and the packet loss probability is low, as that is in wired networks.
- Poor Link Quality: The packet loss probability of wireless networks is much higher than that of wired networks; this is because the wireless networks operates in open environment. The communication is easy to interfere to each other. Because there is no limitation on the number of the connection plugs in the wireless routers, the number of connections in wireless network can be extremely large. All this factors leads to a very low link quality.
- Dynamic Network Topology: The connection of the wireless network is fragile and easy to establish in order to support the device mobility. The topology of the network changes frequently. This requires additional resource to maintain the network system.

## 1.3 Contributions

In my researches, I proposes a Multi-path Error Recovery Video Streaming (MERVS) protocol, which is able to provide high quality and real-time video streaming. Several cross layer improvement techniques are also designed on the network layer and the MAC layer. In order to verify the performance of our proposed protocol, simulations are conducted in the scenario of the downtown area of Ottawa, Canada. The simulation results show that our protocols perform better than the existing protocols.

## 1.4 Thesis Summary

The proposed research attempts to study the high quality and real-time video streaming on VANETs over IEEE802.11; and, a set of protocols will be provided. The quality scalability feature of H.264/MPEG-4 AVC SVC is adopted for video compression to provide scalable video quality. To further improve the performance of video streaming in VANETs, wireless transmission performance improvement protocols will be proposed for wireless transmissions. In order to improve the limitation of OSI model in wireless networks, the cross layer technique will be employed. Based on the cross layer technique, a network functionality can retrieve information from other network functionalities by violating the rules of OSI model. System structure is presented in Figure 1.1.

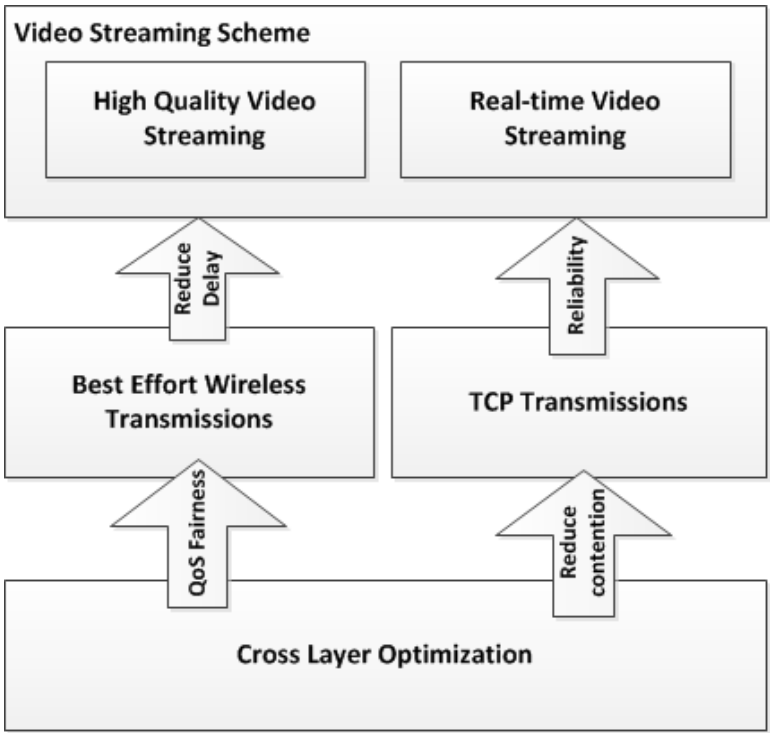



Figure 1.1: System Structure

The following chapters will be organized as follows: Chapter 2 presents the related works of our researches; Chapter 3 proposes the multiple channel solution for high quality and real-time video streaming; Chapter 4 demonstrates a multi-path improvement on the video streaming solution; Chapter 5 discusses the designed routing protocol for improvements on data streaming; Chapter 6 presents the designed MAC protocols for

improvements on data streaming; Chapter 7 indicates the conclusion of our researches and the possible future work of our researches.

# Chapter 2

# Related Work

To offer high quality and real-time video streaming on VANETs, the wireless networks capacity and video transmission approaches should be carefully considered [10, 11]. Because of the high packet loss rate in wireless networks, PSNR of video streaming in wireless networks is very high. Quality of the video streaming is difficult to be guaranteed. Some researches about the video streaming approaches have been proposed to solve this problem. Most of the recent researches are cross layer approaches.

Video streaming is one kind of data streaming. Besides protocols specially offered video streaming, protocols provided for data streaming improvement on wireless networks can also improve the performance of video streaming on VANETs. Service Assurance of telecommunication is one of the solution for data streaming service, which intends to ensure that the service providers guarantee the service quality for optimal user experience, by the methods of service quality degradation detection and resolution. The directions of Service Assurance that we are interested in are performance management, which focus on the performance optimization, and QoS assurance, which focus on the utilization optimization, over wireless networks. The approaches to improve the performance of protocols in wireless network can be classified as non-cross layer techniques and cross layer techniques [101]. Both techniques are going to be presented in this chapter; and, related researches will be discussed in order to compare the advantages and disadvantages of both techniques.

| Paper | Adaptive Scheme | routing | Error Recovery | Multicast | flow splitting |
|---|---|---|---|---|---|
| Asefi et al. [4] | Adaptive MAC retry limit | | | | |
| Xing and Cai [122] | video quality adaptation strategy | cooperative relay | | | |
| Kuo et al. [55] | Adaptive layered video | | | WiMAX multicast mechanism | |
| Soldo et al. [99] | | Distance based relaying | | MAC-layer multicasting | |
| Rezende et al. [90, 89] | | Receiver based relaying | | | |
| Wang et al. [113, 112] | | Multipath receiver based relaying | | | multipath forwarding schemes |
| Hua et al. [44] | | Helper Discovery and Relay Routing | | Scalable Video BroadCast/MultiCast Solution | |
| Tsai et al. [104] | | | Forward-looking forward error correction (FL-FEC) | | |
| Lin et al. [60] | Adaptive FEC redundancy rate | | Enhanced Random Early Detection Forward Error Correction (ERED-FEC) | | |
| Our proposal [121, 120] | | TCP-ETX | TCP retransmission mechanism | | node disjoint and link disjoint scheme |

Table 2.1: A comparison of existing work on flow splitting

## 2.1 Video Streaming

The techniques of video streaming in VANET significantly improve the driving experience. To improve the video streaming over wireless networks, a lot of researches are already proposed in Mobile Ad-hoc Networks (MANETs). The high dynamic topology of VANETs leads to new challenges in video streaming.

### 2.1.1 Video Streaming on VANETs

Video streaming is a well-known topic in Mobile Ad-hoc Networks (MANETs). However, the high dynamic topology of VANETs introduces new problems in video streaming, which MANET never faces. Many researches are conducted to resolve these problems. The survey [106] indicates that the approaches of video multicasting are adaptive schemes, cooperative relaying and Error Recovery Schemes. We also use this classification method in this section for video streaming on VANETs. Other protocols for three-dimension video is also available [80, 64, 65, 66]. Table 2.1 indicates the existing protocols and our proposal.

### Adaptive Schemes

By Improving the channel quality through adapted parameters, the quality of the video stream can be eventually improved. Asefi et al. [4] proposed a multi-objective MAC retransmission limit adaptation scheme for video streaming on VANETs. Their work analyzed the effect of the MAC retry limit to the packet loss of the wireless transmission, in order to estimate the accurate times of retries for video streaming. It handles the tradeoff about increasing the times for retries to ensure the transmission of the packet or decreasing the time for retries to avoid the buffer overflow. Xing and Cai [122] presented a video quality adaptation strategy, which decides the video quality by considering two key parameters: current download speed and receiver buffer level. They defined two threshold for the receiver buffer: the add threshold and the drop threshold. The value of the two thresholds is determined by the density of the vehicles, current speed of the vehicle and distance to two adjacent Road Side Units (RSUs). Oyman and Singh [77] evaluated the performance of HTTP adaptive streaming (HAS) on 3GPP LTE networks, which is a recent popular internet video delivery mechanism. Kuo et al. [55] proposed the Utility Envelope-Based Allocation for Layer-Encoded Multicasting (UE-LEM), an adaptive resource allocation algorithm by layer-encoded IPTV, which maximizes the total utility and the system resource utility.

The purpose of most of the adaptive video streaming protocols is to dynamically change the network parameters and the video encoding parameters to avoid packet loss. However, the packet loss is very common on wireless networks, because of the unstable network environment, especially in VANETs. Because of the high dynamic topology, vehicles are easy to be disconnected and packet losses may occur during the video streaming leading to an unrecoverable damage to the video clip. Even though adaptive schemes can enhance the transmission performance, the broken connection needs time to be detected and recovered. Without any error recovery mechanism, we are unable to recovery frames lost during that disconnection.

### Cooperative Relaying

Cooperative relaying techniques rely on the relay nodes along the path to forward packets to the destination [27]. Soldo et al. [99] proposed a Streaming Urban Video (SUV) protocol based on the path built by the relay nodes. Each node chooses its relay child during the video streaming, based on the distance between the two nodes. The solution is based on Time Division Multiple Access (TDMA), so each relay node has to follow the

schedule time to access the channel. Rezende et al. [90, 89, 91, 92] presented the VIRTUS protocol, which is a localization information-based and receiving-based relaying protocol for video streaming. After receiving the packet that needs to be forwarded, the relay node waits for a short period to check whether another node closer to the destination is going to forward the packets. Therefore, this protocol guarantees the shortest path for video streaming to the destination. Wang et al. [113] provided an extension solution to VIRTUS: LIAITHON, which offers multipath video streaming on VIRTUS. LIAITHON discovers two available paths to the destination, which also minimize the route coupling effect. The multipath transmission improves the data rate of the video streaming. Hua et al. [44] proposed a Scalable Video BroadCast/MultiCast Solution (SV-BCMCS), which is based on relay routing algorithms. The coverage area of the base station is separated into different areas for different video layers. The relay nodes will help deploy the layers to the other nodes to improve their video quality.

In most of the proposed relaying protocols, a common assumption is that the quality of the each relay node is the same. However, it is not always true because of the different quality and the different workload of the devices. The link quality should also be considered in the selection of the relay nodes.

**Error Recovery Schemes**

Vella and Zammit [106] discuss two kinds of error recovery techniques: Forward Crror Correction (FEC) and Automatic Repeat reQuest (ARQ). Most of the error recovery schemes would use only FEC or combined the FEC with ARQ [106], because each retransmission of ARQ adds at least one round trip time delay to the delivery. Bucciol et al. [29] proposed a FEC and an Interleaving Real Time Optimization (FIRO) technique to increase the quality of the communication. That work combined the joint effect of FEC, Interleaving, dynamic parameter update and dynamic update of the inter-arrival frequency of the receiver reports, which can provide service with a maximum allowed packet loss rate and tolerated delay. Qadri et al. [86] discussed an error resilience approach for ensuring the video quality in video streaming, by employing the Flexible Macroblock Ordering (FMO) technique to H.264 AVC. The Forward-Looking Forward Error Correction (FL-FEC) [104] mechanism selects non-continuous source packets in previous FEC blocks to generate FEC redundancy with the FEC block to avoid burst packetloss. Lin et al. [60] proposed an Enhanced Random Early Detection Forward Error Correction (ERED-FEC), which handles the FEC on the access point. The FEC rate is considered for the wireless channel condition and the network traffic load. FEC is a pop-

ular topic in error recovery schemes for video streaming, but ARQ is barely considered. The drawback of FEC is the extra bandwidth consumption caused by the redundancy data. The drawback of ARQ is the delay caused by the retransmissions.

The problem of FEC is the burst packet deliveries, which might lead to high contention level of packet transmissions in wireless networks. This is caused by the duplicated packets generated by the FEC mechanism and the limited resource of wireless networks. Wireless networks might not be able to absorb the extra generated traffic. This situation is more severe in high quality video streaming, because of its large bandwidth requirement. FL-FEC is able to avoid the burst packet loss, by deploying the FEC block separately. However, in real-time video streaming, there is only a small period of time to assign the transmission of a frame in order to minimize the frame delay. FL-FEC is not a good choice in high quality real-time video streaming.

FEC attempts to generate duplicated packet to ensure the quality of the video, but sometimes the duplicated packets might be unnecessary. In order to optimize the estimation of the accurate packet loss probability in a VANET, FEC has to be integrated with an accurate motion prediction of the current vehicle and the surrounding vehicles, the overall deployment of the access point on the road side, the wireless interference level prediction, etc. ARQ mechanism only generates the duplicated packet, when an error happens, which minimizes the number of packets transmitted in the network, with low cost techniques. The only problem of ARQ is how to minimize the delay of the transmissions.

### 2.1.2 Flow Splitting

Flow splitting is one of the efficient approaches to improve the performance of the video streaming. Flow splitting is able to split a large amount of video data into small data flows, which lead to a higher delivery ratio comparing to the one flow video streaming. Several studies [57, 54, 100, 127] have been proposed to multi-path video streaming. Li et al. [57] proposed a joint coding/routing optimization between network lifetime and video quality, based on the information theory to Wireless Visual Sensor Networks (WVSNs). By applying the Distributed Video Coding (DVC) and Network Coding (NC), the proposed protocol improves the performance limit for video streaming. A two-level optimization is also presented, in which the high level optimization problem is to update the coupling variable vector based on the optimal subgraph and the low level optimization problem attempts to solve the local optimization of the power, rate and distortion

| Paper | Encoding | Routing | Reliability |
|---|---|---|---|
| Tsai et al. [105] | | NP | |
| Li et al. [57] | DVC | NC | |
| Zou et al. [129] | SVC | NC | |
| Song and Zhuang [100] | | PGF | |
| Zhu et al. [127] | SVC | Cloud technique | |
| Volkert et al. [108] | | HRM | QoS insurance |
| Kserawi et al. [54] | | FAR | |
| Our solution | H.264/MPEG-4 AVC | TCP-ETX | TCP |

Table 2.2: A comparison of existing work on flow splitting

based on the selected coupling variable vector. The low-level optimization problem also can be decomposed into four cross-layer subproblems: rate control, channel contention, distortion control and energy conservation. Zou et al. [129] presented a similar study to [57], in which Scalable Video Coding (SVC) is used instead of DVC. Kserawi et al. [54] designed a Field-based Anycast Routing (FAR) protocol for real-time video streaming, which is based on the rapid routing dynamics of an electrostatic potential field model following a Poissons equation. An on-the-fly rerouting process is also designed to ensure the continuity of video streaming, when a node failure occurs. Different video streaming scenarios are considered to design the protocol and to measure its performance. Song and Zhuang [100] analyzed the performance of probabilistic multipath transmission of video streaming traffic over multi-radio wireless devices. The Probability Generation Function (PGF) is derived to evaluate the delay metrics, in which the minimum channel data rate to support a video flow/substream can be obtained. In order to minimize the overhead of the splitting flow, the multi-path streaming will be activated if the resource occupancy exceeds a threshold. By deploying the multi-path transmission in multiple networks, it results a high resource utilization comparing to single path transmissions. Zhu et al. [127] presented an efficient cloud-assisted Scalable Video Coding (SVC) video streaming with QoS-aware multi-path provisioning strategies, which improves the performance of SVC. The cloud-assisted scheme can provide better knowledge on the network environment of the requested client. A multi-path algorithm is also proposed in this research to provide video streaming. Tsai et al. [105] proposed a network proxy (NP) based multi-path video streaming technique. A mathematical model is derived to obtain the optimal total

throughput, and a packet scheduling scheme to reduce the impact of packet disorder at the receiver. Volkert et al. [108] discussed a Hierarchical Routing Management (HRM), which is a QoS-oriented routing scheme. The network view is organized in different abstraction levels, in order to create a tree structure. HRM can achieve QoS guarantees based on the pre-set transmission requirements. However, it still does not have an error recovery mechanism, whenever an error occurs. The existing flow splitting techniques are summarized in Table 2.2, which shows that most of the existing flow splitting techniques do not provide reliability. The QoS guarantee provided by Volkert et al. still can not recover from an error.

The aforementioned studies mainly consider the benefit of flow splitting of video streaming on the network layer. To further improve the quality of the transmitted video, the error recovery technique should be added to the multi-path solution. As mentioned above, the retransmission mechanism technique is chosen in this research to ensure the transmission of the video stream. The type of data to be ensured the transmission is also needed to be determined. In H.264, I-frames are more important than the other frames. Therefore, I-frames should be transmitted in a retransmission mechanism technique, and inter-frames should be transmitted without a retransmission mechanism technique to reduce the delay.

## 2.2 Non-Cross Layer Service Assurance for Data Streaming

Besides video streaming techniques, a lot of data streaming techniques are proposed for performance optimization. We classify them as non-cross layer techniques and cross layer techniques. Non-cross layer techniques strictly follow the OSI reference model shown in Figure 2.1. Most of the researches of non-cross layer Service Assurance focus on the improvement on the Transport layer, the Network layer and the Data Link layer, which define the process of the transmissions. Each layer implements different network functionalities. Data link layer offers reliable link connection between two directly connected nodes, which detects and corrects the possible errors; and, one of the famous protocols is IEEE802.11. Network layer provides route information from one node to all possible destination in the same network, by forwarding the data between directly connected nodes; and, the popular protocols for wireless networks in the network layer is Destination-Sequenced Distance Vector (DSDV) routing protocol [82, 83] and Ad hoc

On-Demand Distance Vector (AODV) routing protocol [81]. Transport layer provides end-to-end data transmission for applications; and, the example protocols in transport layer is User Datagram Protocol (UDP) [84] and TCP [85]. Based on the functionality that researchers attempt to improve, non-cross layer Service Assurance approaches improve the corresponding layer in OSI layered model.

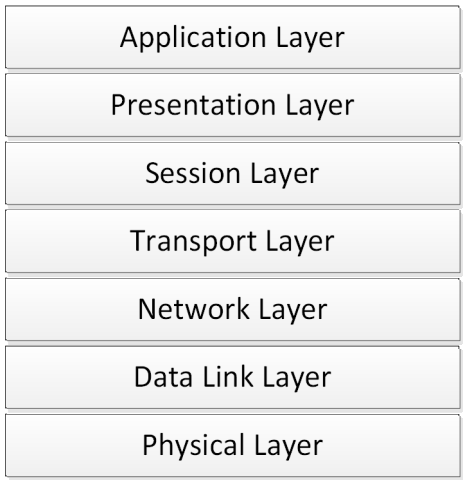



Figure 2.1: OSI Model

### 2.2.1 Performance Management on Wireless Transmissions

Performance management focus on the performance of the system in terms of different parameters. The parameter that we consider in this research is the throughput of the network. The throughput of the wireless networks is limited because the resource of wireless networks is restricted by the link quality and high volume of connections. There are two area that we are going to present in this section: one is the wireless transmission optimization; and, another one is TCP transmission optimization over wireless networks.

#### Wireless Transmission Optimization

Contention window mechanisms are designed to avoid collision on networks. If a terminal attempts to access one resource simultaneously, it will cause collision. When a collision occurs, the contention window mechanism forces the terminal, which has started the transmission, to wait for a random time for retransmission. Thus the retransmission has

a smaller opportunity to collide. A suitable contention window range can simultaneously absorb the increased system load and guarantee the maximum throughput

The Contention Window adjustment algorithms focus on maximizing overall throughput. A Determinist Contention Window Algorithm (DCWA) [53] is a mechanism by which to get a more specific contention window range. During the contention stage, it increases both the upper and lower bound of the contention backoff parameter through the predefined contention window range at this stage. The contention backoff is chosen from a lower bound to an upper bound, rather than from 0 to the upper bound. The contention window's decrease process is based on the current networks load and networks history. The study [110] proposed a GDCF mechanism, which is an improvement upon Distributed Coordination Function (DCF) on IEEE802.11, by limiting the contention window recovery only on continuously successful transmissions. The purpose of the contention window adjustment algorithm is to determine the proper contention window size for a certain moment.

### TCP Transmission Optimization over Wireless Networks

TCP suffers high level of degradation in wireless networks, which makes its throughput almost 10 times less than that of UDP. However, as it is mentioned in the Chapter 1, most of the multimedia software is still implemented with HTTP and TCP, rather than RTP and UDP [69]. The performance of TCP over wireless networks is a great limitation of the reusability of these software. Many researches are conducted in order to improve the TCP mechanism over wireless networks. The main drawback of TCP over wireless network is the congestion window mechanism.

The purpose of TCP congestion window is to reduce congestion over the network. Based on the TCP protocol, each packet loss, no matter the cause, leads to a decrease in the congestion window size. The reason for this is that TCP always treats TCP segment losses as a result of congestion. This assumption is only accurate for stable network environments, for example: wired networks, where packet losses on lower layers are rare. However, over wireless networks, packet loss is very common due to fading, interference and contention, which leads to a decrease in the congestion window size without any actual congestion in the network. This behavior reduces the overall TCP throughput, even though the channel condition is good. A number of research works have been proposed with the aim to improve TCP over wireless networks. Fast retransmission and fast recovery techniques in TCP try to retain a part of the current flow over the network when confronted with packet losses, rather than entering slow start, which would reduce the

congestion window size to 0 and close all current flow [98]. However, fast retransmission and fast recovery still only halve the congestion window size, which ultimately reduces current throughput, despite being unnecessary.

- Congestion Window Adjustment: To improve TCP, the first thought is to modify the congestion window mechanism, in order to make it adjust to the oscillation of the network quality. ElRakabawy and Lindemann [36] proposed a TCP with adaptive pacing techniques, which adaptively adjust the TCP sending rate. Marfia and Roccetti [69] proposed an improved Home Entertainment Center (HEC) architecture for handling different wireless connection. By integrating TCP Westwood into the architecture, there is a significant reduction in network degradation over wireless networks.

- Proxy and Coordinators: Several proxy systems and coordinator systems are proposed to handle the TCP transmissions. Ren and Lin [87] presents a TCP proxy, which dynamically adjust the sending window in the TCP proxy according to the changeable available bandwidth. There is a queue in the proxy system, which store the forwarding message. Normal TCP application only need to forward the TCP segment to the TCP proxy, without considering any TCP improvement. However, the improvement of proxy and coordinators are limited; this is because the high overhead of the centralized solution is not suitable in wireless networks.

- Delayed Acknowledgement (ACK): The concept of delay ACK is to reduce the number of acknowledgements during the TCP transmissions. After each TCP segment is sent, the sender will wait for the ACK of that segment from the receiver to confirm that the segment is successfully received. If the ACK is not received, the sender will send the segment again. In order to reduce the number of ACK transmissions, delayed ACK set one ACK to acknowledge more than one TCP segment. Shi et al. [96] proposed a dynamical delayed ACK algorithm, which dynamically delays the ACKs during the runtime based on the congestion window size and ACK types.

- TCP pacing: One of the main reasons leading to the poor performance of TCP over wireless networks is the TCP burstiness. Many observations indicate that bursty traffic caused by congestion control mechanisms degrade TCP performance [2]. The causes of TCP burstiness are as follows: slow start, segment losses, Acknowledgement (ACK) compression and multiplexing. It can be summarized in

two scenarios: 1) buffer overflow or 2) a large number of segments sent simultaneously. Therefore, some researchers focus on spreading a large number of TCP segments over the whole RTT with TCP pacing. [36] proposed a TCP with an adaptive pacing technique, which adaptively adjusts its sending rate to spreads the data segment transmission in the congestion window over a period of RTT. [76] introduces link RED, a dropping mechanism in 802.11; it increases the backoff timer of a node, while its number of retries is more than the minimum threshold. [63] described a pure rate based TCP for replacing the congestion window. An interval between two TCP segments is calculated based on RTT and SRTT. [42] presented a link RED and adaptive pacing algorithm for TCP pacing in order to set a suitable TCP congestion window for TCP transmission. Besides rate control on the senders, the receivers can also achieve an ACK rate control. [8] presented an ACK pacing technique by preventing the fixed host from timing out based on the network conditions; this in turn reduces the degradation of TCP throughput caused by the temporary network condition. [71] modifies the TCP receivers to adjust the transmission rate of the ACKs, in order to control the transmission rate of the TCP senders with the ACK clocking mechanism.

### 2.2.2 Quality of Service (QoS) Assurance on Wireless Transmissions

QoS assurance intends to reserve the resource for requests from subscribers, in order to guarantee certain quality requirements of the subscribers. Because the insufficient resource in wireless networks, QoS assurance is important for keeping the performance level of the services, especially for multimedia applications. The parameter of QoS that we are going to consider in this research is the throughput. The throughput of wireless networks is much less than that of wired networks, especially when facing the high volume of connections. It is critical to decide how to guarantee the throughput of each service request and how to avoid or reduce the contention with large number of simultaneous connections. A reasonable resource assignment mechanism should be proposed.

### 2.2.3 Contention Window Classification

The idea of resource distribution over wireless networks is contention window classification, which separate the service request in different contention window categories based

on their requested resource. IEEE802.11 includes a contention window mechanism, which in order to avoid the occurrence of contention. If two or more packets are transmitted to the same destination node simultaneously, the transmissions will be collided. The collided transmissions will be suspended; and, the retransmissions of the packets will be postponed based on the decision of contention window mechanism in order to reduce the collision probability of the retransmissions. The period of time to wait for the retransmission is difficult to determine; this is because if it is too small the collision probability is very high, and if it is too large the channel might be idle for no reason. In addition to this reason, contention window mechanism add extra time interval among transmissions, which will affect the data rate of the services. The analysis of collision probability is able to improve the networks performance by scheduling the packets based on the networks traffic estimation [52].

IEEE802.11 [68] includes the Enhanced Distributed Channel Access (EDCA), which is designed for QoS assurance over the general IEEE802.11. EDCA defines access categories (ACs) for different traffic based on their priorities. High priority traffic has a higher chance to send a packet than low priority traffic, because the smaller contention window size is assigned to the high priority category. The contention window is set based on the service that is expected in each category, for example video transmissions and voice transmissions. Besides EDCA, the Transmit Opportunity (TXOP) is also proposed to enable the service to access the media freely in a short period of a time. In contrast to the previous transmission protocols, which handle the transmission by one packet, TXOP provides a way to offer transmission service by a session. In this case, the service is able to reserve the resource for a period of time, which is not only able to guarantee the throughput but also the jitter or other parameters for multimedia transmissions. More researches are proposed in contention window classification area in order to offer QoS in wireless networks.

Besides the IEEE802.11e, a lot of other researches have been proposed to achieve QoS assurance in wireless network. A non-contiguous contention window range mechanism [72] was proposed in order to provide better relative performance between high priority class and low priority class. This method only support two priority classes and one total contention window range. Usually, there are more than two QoS classes for the differentiating application. The design should be able to support multiple priority classes. In order to guarantee the QoS of real time multimedia application on IEEE802.11, an Adaptive Contention Window Size Adjustment scheme (ACWA) [124] has been developed. A cross layer design was introduce to ACWA, in order to cooperate with the

Networks layer and MAC layer. The MAC layer was designed as a multiple priority queue for outgoing packets. The packet scheduler puts packets of a flow in the suitable queue, based on a flow requirements and channel state. A Multiplicative-Increase Linear-Decrease (MILD) [7] based on a Sliding Contention Window (SCW) was proposed, which provide a SCW per priority class. The contention windows changing on the contention stage is based on the MILD algorithm, with the aim to get higher performance under heavy system load conditions. This research focuses on how to manage the different QoS constraints among priority groups. The study [118] provided a Call Admission Control (CAC) design in order to satisfy the QoS fairness between high priority and low priority connections. Adaptive Contention Window Adjustment (ACA) is an additional mechanism provided in order to reduce the collision while the system load increases. An improved Distributed Coordination Function (DCF) scheme, DCF-MB, has been proposed in [31]. This research tries to solve the unfairness behavior of the DCF algorithm by multiple contention windows principle.

Even though most of the current QoS classification algorithms can partially solve the QoS problem on Ad Hoc Networks, they cannot achieve real QoS fairness over the networks for the same reasons as EDCA and HCCA, which ignore the external relation and the slight difference among service requirements. In order to offer better QoS assurance, each service requirement should be carefully considered. [111] pointed to a solution to the QoS fairness problem caused by DCF on IEEE802.11. This paper presents a non-linear function by which to measure the difference between the achievable throughput and the corresponding QoS requirements. A Neural Networks will be built based on this non-linear function. After learning the Neural Networks, the non-linear function can adaptively determine the suitable backoff for the specified QoS requirement. However, this approach need extra time for learning, while the network topology is changed. Unfortunately, dynamic network topology is one of the main challenges of wireless networks. A light weight solution should be proposed.

## 2.3 Cross Layer Technique over Wireless Networks for Data Streaming

Non-cross layer techniques are merely capable to patch up the problem occurring in wireless networks; however, they can not really solve the problem, unless they create a new protocol. One of the main causes of the poor performance of the protocols in

wireless networks is the OSI layered model, which restricts the exchanging information from layer to layer. OSI model merely create interface between adjacent layers for limited information exchange. Each layer is impossible to obtain extra information from adjacent layers; and, it is also impossible to retrieve information from other layer by crossing the adjacent layers.

Cross layer technique is an designing approach by violation of the reference layered model [101, 40]. This technique is suitable for wireless protocol design, because the strict OSI layered model is unable to react to the changeable quality of wireless networks. Cross layer technique allow an protocol to retrieve additional information about network condition, which simplifies the way to estimate the network quality and enhance the speed to react to the dynamical link quality.

### 2.3.1 Approaches for Cross Layer Design

To design a cross layer protocol, there are several different existing approaches:

#### New Interface

OSI layered model only offers interfaces between adjacent layers. New interfaces have to be designed in order to allow additional information exchange between two layers. Figure 2.2 presents the new interfaces between two different layers. The interface can be created from top to bottom, like the one from *Layer 4* to *Layer 1*. It also can be created from the bottom to the top, like the one from *Layer 1* to *Layer 3*. The creation of the interface is simple and flexible, but usually it is good to make it just one way and it is merely between two layers.

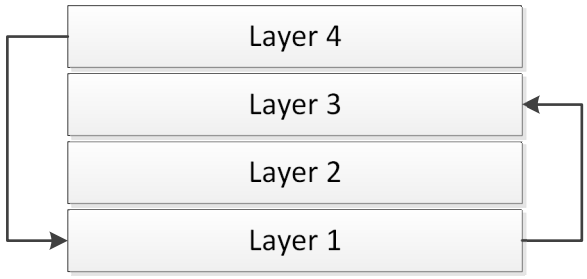



Figure 2.2: New Interface Between Layers

### A Shared Database

In order to allow the data sharing among multiple layers and unify the way of data management, a shared database can be used for data exchange among different layers. The shared database provides information storage and retrieval service to the connected layers. The advantage of a shared database is its pre-defined and unified data sharing and retrieval procedure, which simplify the development of new cross layer protocols. The shared database also help to manage the data sharing.

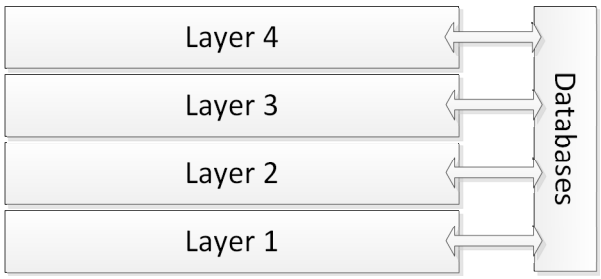



Figure 2.3: A Shared Database For Information Exchange

### Merging of Adjacent Layers

Some cross layer design may merge the adjacent layers in order to simplify the functionalities or to enhance the capability of the layer by creating a superlayer. The example is shown in Figure 2.4. Network-assisted Diversity Multiple Access (NDMA) [125] is an example solution by merging the physical layer and Media Access Control (MAC) layer, in order to solve the collision problem by improve the signal processing in Physical layer.

### Design Coupling

Design coupling method for cross layer design is different from the designing approaches discussed above. Without creating any extra interface and communication mechanism, design coupling approach achieve the cross layer idea at the design phase. The design of a layer is based on the functionality of a reference layer. Data exchange is not needed in design coupling, because the system is in integration design. The designed layer can be above the reference layer or below the reference layer.

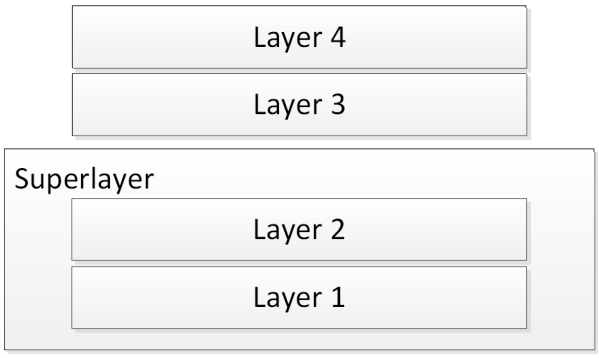



Figure 2.4: Merging of the Adjacent Layers

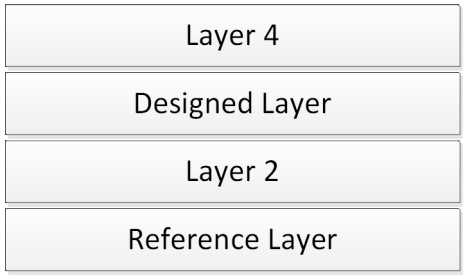



Figure 2.5: The New Layer is Designed Based on the Reference Layer

#### New Abstraction

The final goal of the cross layer design is to create a complete new network abstraction, which is different from the existing OSI layered model. To match the feature of wireless networks, a flexible network abstraction should be proposed, which react fast to the various network connection quality. Braden et al. [28] design a design a new network abstraction based on the concept of heap, which is not the protocol stack like OSI layered model. The functionalities of the networks is designed as *Role* in the heap; and, the interfaces between layers are replaced by the interactions among role, which is control by the Middle Boxes.

### 2.3.2 Cross Layer Design Service Assurance

Recently, the most common network protocols used for application development are UDP and TCP. In order to reuse the existing applications over wireless networks or to keep the application design in the same way, the performance of UDP and TCP has to be improved. As we discuss in the previous section, the non-cross layer technique is difficult to adjust to the rapidly changed network quality. The adoption of cross layer technique is very helpful to make the system flexible enough to react to quality change. There are many protocols proposed under the cross layer technique in order to improve the performance of both UDP and TCP. We classify these researches based on the layer that they interact with. There are two layers will be discussed in the following section: MAC layer and Network layer.

#### MAC Layer

MAC layer is a sub layer of Data Link layer, which provides access control for the shared media. The basic function of MAC in IEEE802.11 is link connection. Several mechanisms are created to enhance the network reliability. The backoff mechanism is proposed to resolve the problem caused by the simultaneous multiple accesses to the shared media. If two or more node send packets to the same destination, the packet will be collided, and the destination can not read the transmitted packets. The backoff mechanism postpone the retransmission of all transmitting nodes with a random time, in order to avoid the occurrence of the next collision.

The Request To Send (RTS) and Clear To Send (CTS) mechanism is also adopted in IEEE802.11, which confirm the channel is idle before the data transmission starts.

Before the packet is sent, the source sends a RTS to request a transmission and a waiting timer will be set. If the destination receive a RTS, it will reply a CTS if it is ready to do so. By receiving a CTS, the source knows the destination is ready to receive a packet; and it will schedule the transmission of the data packet. An acknowledgement will be sent, when the data packet is receive in the destination side. If the CTS is not received in the source side and the waiting timer is timeout, the source will retransmit the RTS until the times for retransmissions reaches the maximum number. The purpose of RTS/CTS mechanism is to establish reliable link between two connected nodes.

As we mention in the section above, the TCP congestion window treats all TCP segment loss as a result of congestion, which means TCP believes the buffer of receiver side is overflowed. Therefore the sender side narrows the congestion window and slows down the transmission. Research [41] shows that TCP segment loss is mainly due to contention on the link layer; and, contention loss usually happens before the TCP buffer overflows. Therefore, in addition to techniques on queue management, congestion window adjustment, and delayed acknowledgement, cross layer monitor and optimization are one of the most important techniques for TCP improvement over wireless networks. The way to decide if a TCP segment loss is caused by congestion or contention does not exist in normal TCP; this is because the lower layer of the network is hidden by OSI layered model. Therefore many researches have been proposed to solve this problem.

To estimate the RTT, Zhang et al. [126] split the RTT into two parts: congestion RTT and contention RTT. Contention RTT is the amount of delay caused by contention on the link layer, while congestion delay is the sum of the remaining delay, such as queuing delay, transmission delay and processing delay. This paper assumes that the transmission rate of successive segments is mainly determined by congestion RTT. With an estimation of contention status, a modified bandwidth-delay product (BDP) has been presented in order to illustrate the network capacity, which instructs the congestion window adaptation mechanism in order to establish a suitable congestion window size. Park et al. [79] studied the unfairness of wireless network, and concludes that the causes are TCP asymmetry and Media Access Control (MAC) asymmetry. This paper indicates that both TCP and MAC only take the sender side into consideration as it give less service opportunities to the receiver side. In considering fairness over wireless networks, this paper presents a cross layer feedback approach by investigating the interaction between TCP and MAC. This algorithm measures the channel access cost for TCP at the MAC layer in order to control the congestion window.

The aforementioned studies use a cross layer technique to monitor network conditions,

since it is able to access lower layer parameters. However, the cross layer technique can do more than monitoring and estimation. Because TCP segment loss is mainly caused by contention on wireless networks [41], contention control is another approach by which to improve the performance of TCP. Luo et al. [62] analyzed the relation between TCP throughput and channel conditions on cognitive radio networks, and models the channel as Partial Observable Markov Decision Process (POMDP) for solving the throughput optimization problem of secondary users. This paper indicates that the size of the frame on link layer is one of the factors that affect the overall throughput of TCP based on the derived equation. Therefore, the action space of the POMDP model determines which frame size will be chosen based on current network conditions. The direct reward of the POMDP model is the estimated throughput of current network parameters. Chen et al. [30] chose a similar way to solve the problem of TCP performance in cognitive radio network for secondary users. This paper models the channel as a Finite state Markov Chain (FSMC) model as the Rayleigh block-fading channel.

**Network Layer**

Network layer is responsible for packet forwarding through a selected route. The quality of the route will cause different effect to the protocol performance. In order to select the most suitable route for different protocols, the mechanism of those protocols should be considered carefully. Even through Network layer does not provide flow control mechanism, it can eventually affect the network performance by providing a suitable path for transmissions.

- Wireless Optimization: There are many existing routing protocols have been proposed to provide high quality routes. Expected Transmission Count (ETX) [32] is a path metric that estimates the corresponding number of transmissions needed to successfully transmit a packet, including retransmission. It probes all destinations in a network in order to retrieve accurate information regarding packet loss probability to all destinations. Based on the packet loss probability, it derives the corresponding number of transmissions to successfully transmit a packet. By lowering the transmission of packets, the performance of the transmission can be greatly increased as well. Krco et al. [51] presented a research on modification of neighbour detection of AODV for performance improvement. Wang et al. [114] discussed the possibility of using joint routing-and-scheduling in wireless networks. They propose a joint routing-and-scheduling algorithm for mesh networks, based

on the traffic information uncertainty.

- TCP: In order to find a high quality path for TCP, the link condition should be considered rather than the minimum number of hops. Nahm et al. [74] studied the TCP operation range with static routing, which results in a low bandwidth delay product in multi-hop IEEE 802.11. This work proposes two techniques to improve TCP performance: TCP fractional window increment (FeW) scheme and Route-failure using BUlk-losS Trigger (ROBUST) policy, where FeW is used to prevent link loss and ROBUST works as an on-demand routing protocol. ROBUST performs routing according to the following instances of link loss on the link layer: such as node mobility, congestion and channel noise. However, their proposed solution does not consider how to improve the TCP throughput by using these techniques. On the other hand, INSIGNIA signaling system [56] supported quality of service insurance over ad hoc networks. It offers a fast path reservation and the restoration of service by integrating QoS and routing protocol. IP packet is self-contained, so that reservation and restoration are fully maintained by the routing. INSIGNIA modifies the Distributed Control Function (DCF) of IEEE 802.11 to offer a set of services for INSIGNIA.

  Many modifications of ETX have been proposed to enhance its performance. Zhang et al. [123] investigated the existing data-driven estimation based on the ETX routing protocol and classified them into two groups: L-NT, that uses aggregate information, and L-ETX, which uses individual information. Zhang et al. show that L-ETX achieves a much more accurate and stable performance than L-NT, and L-ETX has a higher level of reliability and energy efficiency than L-NT. Koutsonikolas et al. [50] proposed a new efficient network coding based opportunistic routing (OR) protocol, CCACK, in which nodes can acknowledge the coded network traffic and inform the upstream node about loss rates. To lower the overhead of that protocol, they introduced a cumulative coded acknowledgment. Jung et al. [48] studied traffic load balancing over three-dimensional wireless mesh networks, and proposed a global load balancing scheme with only one hop information. The proposed scheme is tested on a testbed and compared with the Optimized Link State Routing (OLSR) ETX protocol. Huang et al. [45] introduced greedy forwarding into ETX given its simplicity and light weight. That study proposed a Topology Aware Routing (TAR) protocol, which stores the distance of nearby nodes. Besides integrating TAR with the ETX protocol, this particular solution

chooses the optimum path. By combining these techniques, TAR can achieve low cost routing, find less transmission path and forward packets that are geographically closer to the destination in the physical space. Javaid et al. [47] proposed an interference and bandwidth adjusted ETX (IBETX) routing protocol that uses cross layer techniques. IBETX calculates the interference on the MAC layer and the bit rate of all nodes in the same contention domain. After that, an ETX metric helps to choose the optimum path that bypasses heavy traffic areas.

Besides the high quality path, some researchers focus on different aspects of ETX. Dong et al. [34] proposed a secure routing protocol, by preventing compromised nodes to offer malicious routing information. Their algorithm is able to identify attacks and to restore the malicious information.Jakllari et al. [46] indicated that not only does the number of hops and link quality affect the TCP transmission, but the link position also has a very important impact on TCP traffic. They studied a simple model with a lower quality link on a different position, which resulted in totally different values of TCP throughput. The work in question proposed a path metric, Expected number of Transmissions On a Path (ETOP), which is a modified ETX. The authors' protocol models the network as a graph, and computes the smallest ETOP through the graph model. The main drawback of this protocol is that it needs a centralized decision maker for collection information from the network to complete the graph model. Its scalability might be a problem in large scale networks.

# Chapter 3

# Multi-channel Error Recovery Video Streaming on VANETs

The purpose of our researches is to propose a scheme for high quality and real-time video streaming on VANETs. Based on H.264/MPEG-4 AVC, both the P-frames and the B-frames have to be decoded by the reference frames, while the I-frame can be decoded independently. Therefore, most of the video information are save in the I-frame. The loss of a I-frame will cause chain effect on the information loss in the following P-frames and B-frames. Therefore, error recovery technique should be applied to the I-frame transmission for guaranteeing the transmission of the I-frames, in order to maximize the quality of the video. TCP transmission is used to achieve the error recovery on I-frame transmissions, by the ARQ mechanism of TCP.

## 3.1 Multi-channel Error Recovery Video Streaming on VANETs

The quality of a video streaming is measured by both temporal and spatial criteria. The usual temporal criterion is the frame rate, and the usual spatial criterion is known as the Peak Signal-to-Noise Ratio (PSNR). To improve the video streaming on VANETs, both criteria need to be considered. The video compression techniques H.264/MPEG-4 AVC SVC provide an scalable video format for video streaming. It provides both temporal scalability and spatial scalability for transmitting video on the networks with different quality.

Figure 3.1 shows the temporal scalability of H.264 by classifying the video frames as

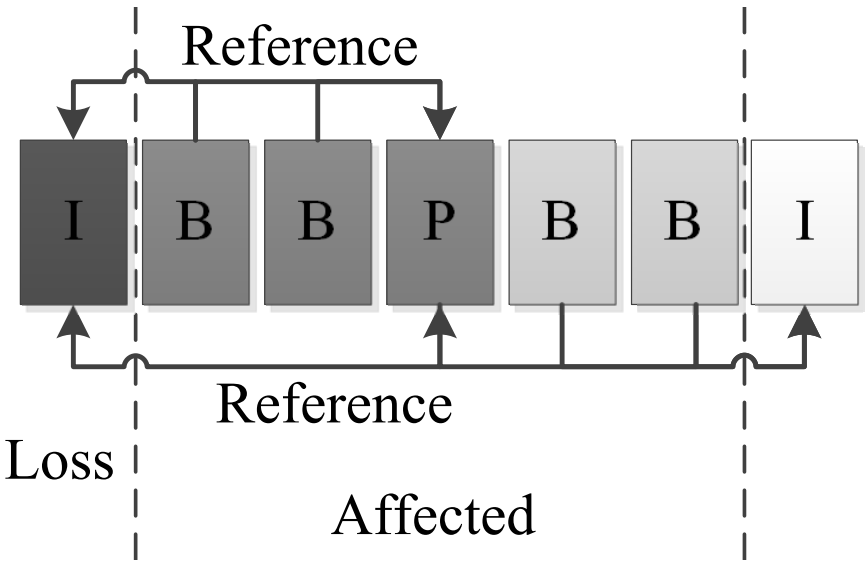



Figure 3.1: Effect of the loss of I-frame in video streaming

I-frame, P-frame and B-frames. I-frames contain the full information of the image, and P-frames and B-frames only contain partial information, which need to be decoded by other reference frames. As shown in Figure 3.1, a P-frame needs a reference information from the directly previous I-frame or P-frame to decode, because it eliminates the temporal redundancy of the video stream. Different from P-frames, a B-frame needs a reference information from one directly previous and one directly following I-frame or P-frame for decoding. In Figure 3.1, the loss of the first I-frame will affect the directly following P-frame, because it is the only reference frame of that P-frame. The damaged P-frame will affect the following P-frames and B-frames, because of the loss of redundancy information. The damage on the last few B-frames will not be that serious, because it also refers to the reference frame after it, but its quality is still affected. The effect of the lost I-frame only stops, when the next I-frame is received which contains the full information. The loss of an I-frame damages the whole Group Of Picture (GOP). Actually, the receiving of an I-frame will also benefit the previous GOP, which can be observed from the last B-frames decoded based on the I-frame received after it. Therefore, the information in an I-frame is very important for generating the visible frames. Any loss of an I-frame during the transmissions will cause a huge damage on the video, and the guarantee of the transmission of an I-frame can eventually improve the quality of the video streaming.

### 3.1.1 Multi-channel Video Streaming

Given that frames have different roles, they should be considered differently. Previous studies assign different priorities for different frames to achieve packet dropping in the gateway or intermediate nodes. When the queue of an intermediate node is full, low priority packets have a higher chance to be dropped. In this way, high priority packets have a higher chance to be transmitted. However, there is still a probability for the loss of important messages, because of fading, interference and collision. The loss of an I-frame will cause a serious chain effect in the video clip. On the other hand, the quality of the video clip will be highly improved, if the transmission of I-frames are guaranteed. Even though a P-frame can be a reference frame, it is not going to be assigned to a reliable channel, because the information carried by P-frames is not as important as the I-frames, and the capability of a reliable channel in wireless networks is limited as well. To minimize the information that is needed to be guaranteed, only I-frames are transmitted through the reliable channel, while P-frames and B-frames are transmitted in the unreliable channel.

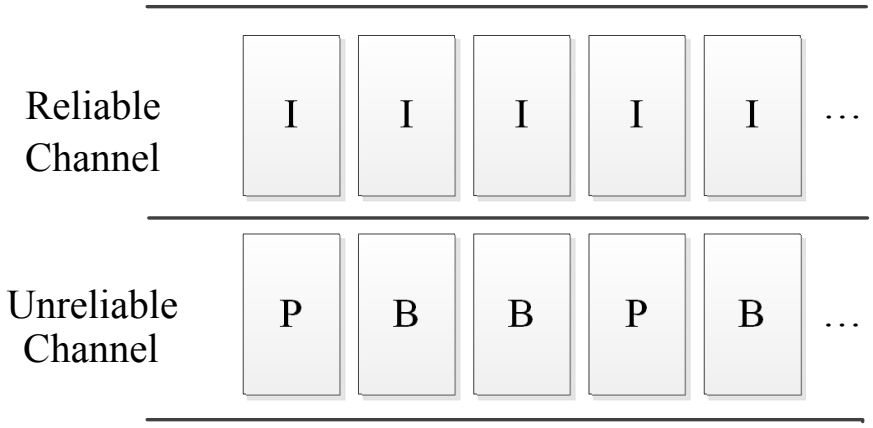



Figure 3.2: Transmitting different types of frame in different channels

The reliable channel, for the I-frame transmissions, uses the TCP protocol whereas and the unreliable channel uses the UDP protocol. The Automatic Repeat reQuest (ARQ) technique of TCP increases the chances of a message to be received. However, because of the degradation of TCP in wireless networks, TCP is barely considered in video streaming in a wireless network, even though most of the multimedia software is implemented using HTTP and TCP, rather than RTP and UDP [69]. The throughput of the TCP protocol is usually much lower than UDP in wireless networks. However, TCP

is still able to transmit the important information in video streaming with limited size. Therefore, TCP is used in our research to implement the reliable channel.

### 3.1.2 System Architecture

MERVS contains two channels, the reliable channel and unreliable channel. The reliable channel guarantees the packet transmission and the unreliable channel does not contain any reliability mechanism. Figure 3.3 depicts the architecture of our system. There is a frame filter in the sender side responsible for deciding the corresponding channel for the input frames. In our protocol, I-frames are assigned to the TCP protocol, and P-frames and B-frames are assigned to the UDP channel. The ARQ mechanism in TCP guarantees the transmissions and order of I-frames, generating a high quality base frame sequence. The P-frames and B-frames sent through the unreliable channel enhance the temporal quality by increasing the frame rate. The frame reassembling algorithm in the receiver side regenerates the video according to the order information in the packets.

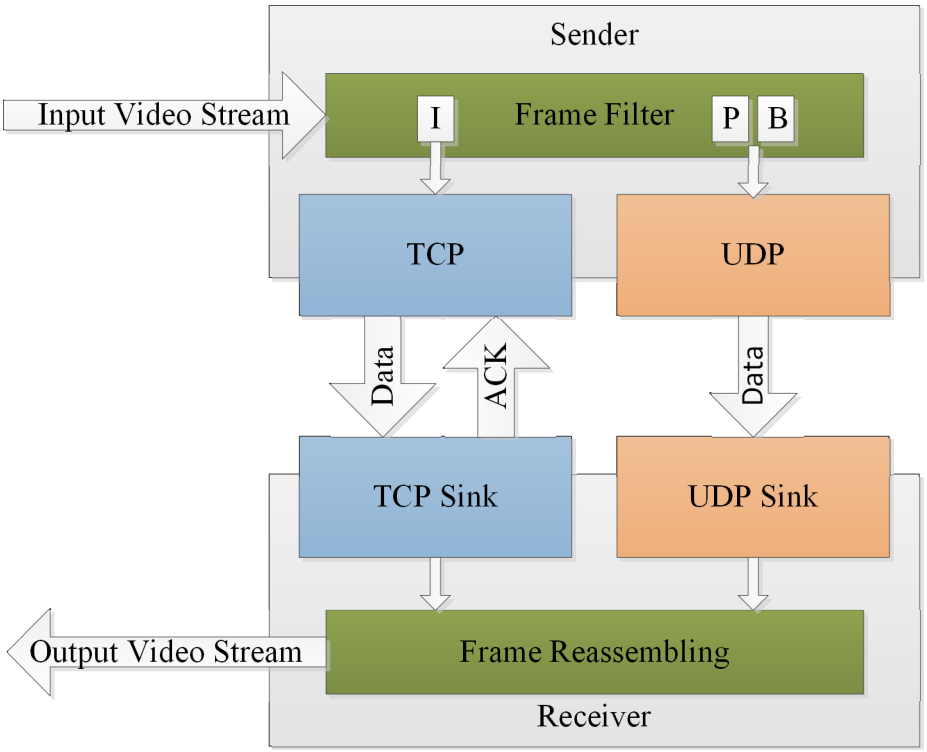



Figure 3.3: Process of multi-channel video streaming

As mentioned, TCP suffers degradation in wireless networks, especially in a VANET.

Therefore, a video streaming solution based on TCP is unrealistic in a VANET. On the other side, a video streaming solution based on UDP needs a reliability mechanism in the upper layer to improve the transmitted video quality. In this work, we propose combining both TCP and UDP protocols to achieve the desired video quality.

### 3.1.3 Computational Complexity

One concern about transmitting video in different channels is the computational complexity to separate the video stream into intra-frames and inter-frames and the computational complexity to combine the two channels into the video stream. The patent [103], published in 2013, describes a video frame parser able to separate the I-frame and inter-frame in a video stream. That patent proposes storing the I-frame data and inter-frame data separately in order to improve the performance of trick play. In that way, there is no extra computational complexity in our scheme, if this technique is used to preprocess the video stream.

A sequence ID should be assigned to each frame packet in order to regenerate the video stream. After the streams from the reliable channel and the unreliable channel are received, they will be sorted by the sequence ID first to simplify the video regeneration. The computational complexity of sorting is $O(n \log(n))$, which is the expected cost of the average case of Quicksort. Both the received I-frame and inter-frame sequences will be scanned and the sequence ID will be compared. The packet with a smaller sequence ID will be placed before the packet with a larger sequence ID in the video stream. The computational complexity will be $O(n)$. Therefore, the total computational complexity will be $O(n \log(n)) + O(n)$.

### 3.1.4 Simulation Result

The purpose of our simulations is to test the behavior of the multi-channel error recovery solution for future improvement. Multi-channel error recovery solution is tested under two scenarios: the V2V scenario and the V2I scenario. The V2V scenario is a two lanes traffic. Both lanes are going to the same direction. The distance between two adjacent vehicles is 100m, and the video streaming is tested in two hops transmission, which extends the range of the video streaming. The speed of the vehicle is assume to be stable and the same in our scenario to simplify the scenario, which is set to 13m/s. The video sample clip used in this simulation is with $352 \times 288$ resolution and 10s long, which is close to the quality of VCD. The simulation result in Figure 3.4 shows that multi-

**Algorithm 1** Regeneration of the Video Stream

**Require:** $Iframes.length \geq 0$ packets array of I-frames

**Require:** $PBframes.length \geq 0$ packets array of PB-frames

1: $Icur = 1$
2: $PBcur = 1$
3: $VideoCur = 0$
4: $Iframes[0] \rightarrow IF$
5: $PBframes[0] \rightarrow PBF$
6: **while** *running* **do**
7:  **if** $IF.ID < PBF.ID$ **or** $PBframes.length < PBcur$ **then**
8:  $IF \rightarrow Video[VideoCur]$
9:  **if** $Icur < Iframes.length$ **then**
10:  $Iframes[Icur] \rightarrow IF$
11:  **end if**
12:  $Icur++$
13:  $VideoCur++$
14:  **else if** $PBF.ID < IF.ID$ **or** $Iframes.length < Icur$ **then**
15:  $PBF \rightarrow Video[VideoCur]$
16:  **if** $PBcur < PBframes.length$ **then**
17:  $PBframes[Icur] \rightarrow PBF$
18:  **end if**
19:  $PBcur++$
20:  $VideoCur++$
21:  **end if**
22: **end while**

channel error recovery solution can offer higher PSNR comparing to merely use UDP for transmission. The several peaks of PSNR in multi-channel error recovery solution is caused by the successful receival of I-frames. The received I-frames even benefits the PSNR of the following frames and the previous B-frames. From the result, it can be seen that UDP rarely successfully receives a complete I-frame; thus, the UDP fails to transmit the high quality video clips. However, multi-channel error recovery solution suffers PSNR degradation in the first few frames; this is because the bursty transmission at the beginning of TCP collides and interferes with the UDP traffic. Therefore, the transmission of the UDP traffic is delayed by 1s, and multi-channel error recovery solution gets better PSNR at the first few frames.

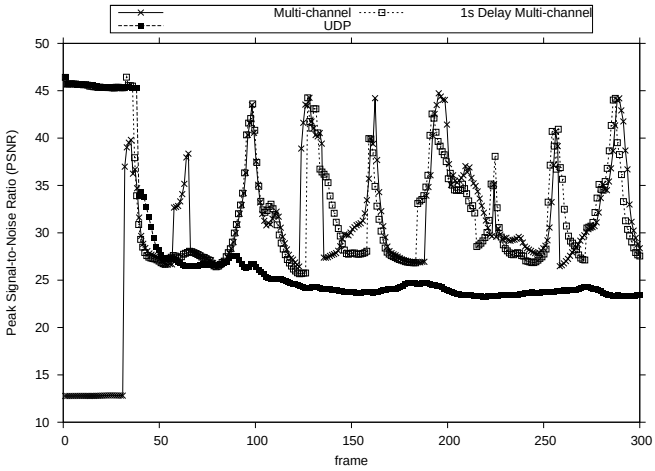



Figure 3.4: Video streaming simulation result on V2V scenario

| Protocols | Received Frames | Average PSNR | STDV | Transmission Time [s] |
|---|---|---|---|---|
| UDP | 236 | 26.4 | 7.04 | 10.680588 |
| multi-channel | 281 | 31.22 | 4.78 | 15.959039 |
| 1s Delay multi-channel | 283 | 31.65 | 6.31 | 14.718831 |

Table 3.1: Video streaming parameter of V2V scenario

The only problem of multi-channel error recovery solution is the time needs for video streaming. In Table 3.1, it shows that even though multi-channel error recovery solution give a higher PSNR, the transmission of multi-channel error recovery solution needs about half of the time more than the transmission of UDP. The V2V scenario is tested

again by restricting the time for transmission to 10s, which is the play time of the video. The simulation result is shown in Figure 3.5. Multi-channel error recovery solution and 1s delay multi-channel error recovery solution does not successfully transmit one whole I-frame, but the 2s delay multi-channel error recovery solution is able to successfully deliver one whole I-frame. Therefore, it can be seen that the problem of our research is the TCP transmission is delay by the UDP transmission, because of the collision and interference. The extra time of multi-channel error recovery solution transmissions is cause by the TCP transmissions, which wait for the UDP transmission to finish when UDP transmission occupy the route.

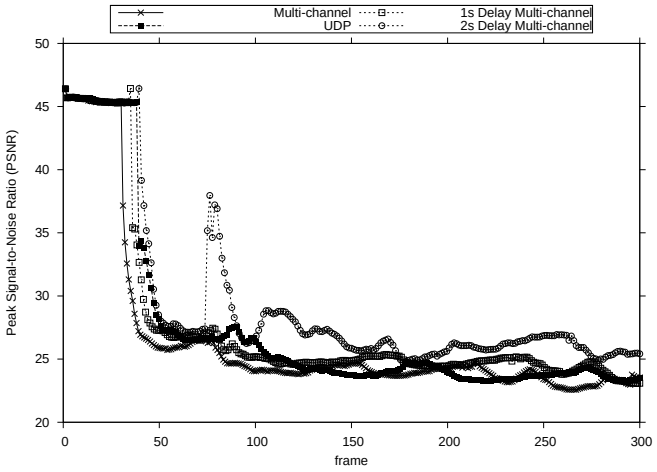



Figure 3.5: Video streaming simulation result on V2V scenario with limited time

Our V2I scenario is set up in the similar two hops scenario. A vehicle is moving towards the RSU and a connection is maintain by one previous vehicle which is closer to the RSU. When the vehicle move inside the range of the transmission, the video streaming starts. Figure 3.6 shows the simulation result of the V2I scenario. It gives similar result in the V2V scenario. The transmission still suffers a degradation at some frames because of the busty transmission, but the I-frames get successful delivered after. 1s Delay multi-channel error recovery solution can avoid the bursty at the beginning, but it still suffers burty later because of the limited route in this scenario. In Table 3.2, multi-channel error recovery solution still needs extra time for video streaming, comparing to the UDP transmission.

If the transmission time is restricted to the play time of the video clip, the similar result will be retrieved in Figure 3.7. The transmission of I-frame is still fail for multi-channel error recovery solution and 1s delay multi-channel error recovery solution. The

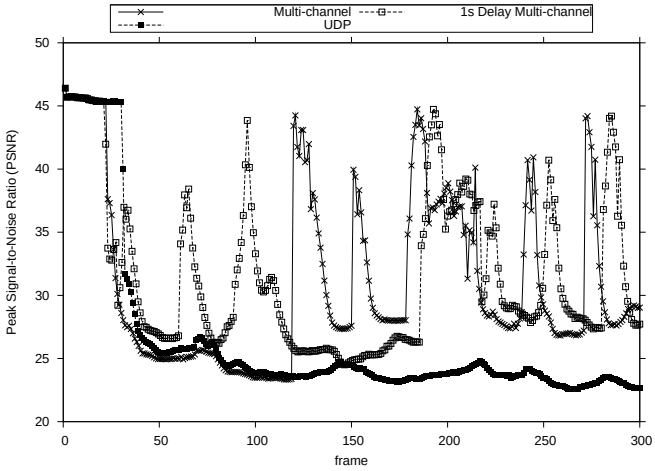



Figure 3.6: Video streaming simulation result on V2I scenario

| Protocols | Received Frames | Average PSNR | STDV | Transmission Time [s] |
|---|---|---|---|---|
| UDP | 299 | 26.4 | 6.59 | 10.680588 |
| multi-channel | 290 | 31.22 | 4.78 | 13.268961 |
| 1s Delay multi-channel | 290 | 31.64 | 3.34 | 13.178452 |

Table 3.2: Video streaming parameter of V2I scenario

interference and collision between the UDP and TCP should be considered in the future work.

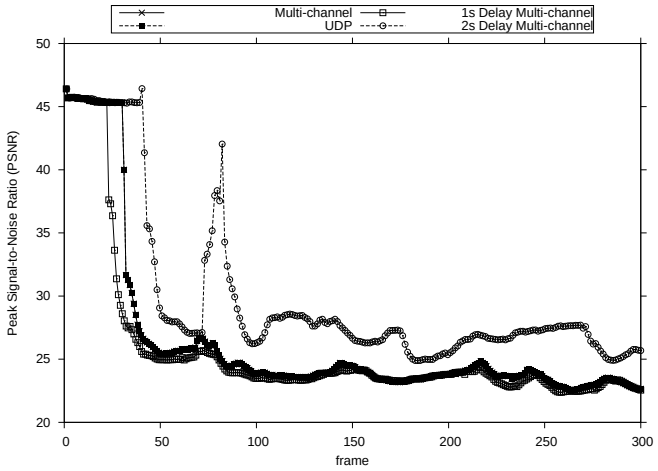



Figure 3.7: Video streaming simulation result on V2I scenario with limited time

## 3.2 Delay Improvement Techniques of Multi-channel Error Recovery Solution

A problem of MERVS is the delay caused by the reliable channel caused by the degradation of TCP in wireless networks. A failure of any TCP segment will increase the delivery delay by one Round Trip Time (RTT). In a VANET, this situation is much worse because of the fast changing topology and the fragile connections. The segment delivery ratio is lower than in regular wireless networks. TCP suffers extra delay because of the low delivery ratio. The interaction of the reliable channel and the unreliable channel will also delay the reliable channel because of the disorder when both channels access the shared medium. The cause of the delay is as follows:

1. **Channel coupling between TCP channel and UDP channel**: both TCP and UDP ignore any reliability mechanism of the lower layers because it is hidden in the protocol stack. Therefore, TCP and UDP transmit packets to the lower layers without noticing if such mechanism is available or not. Because the IEEE 802.11 standard sends one frame at a time in the Medium Access Control (MAC) layer, there should be a waiting queue to store the messages generated by TCP and UDP.

There is only one queue to store the generated messages, so both TCP and UDP segments will be put in the same queue. The segment order becomes a problem for TCP and UDP because of the confirmation mechanism of each protocol (confirmed in TCP versus unconfirmed in UDP). UDP segments occupy the queue when TCP segments are waiting for their acknowledgements.

2. **Low efficiency of the TCP channel**: the TCP mechanism tries to avoid congestion during the network transmissions. Congestion control prevents the source from sending more messages than the destination's capacity, causing a receiving buffer overflow. However, given a reasonable large buffer, message loss in wireless networks is mainly due to contention, rather than congestion [41]. Another problem is that the TCP congestion control mechanism treats all delivery failures as a result of congestion, which is incorrect in an unstable network condition, such as in a VANET. Congestion control slows down the TCP transmissions during the detected congestion, which might be a false alarm.

3. **Video packet scheduling for TCP Channel and UDP Channel**: given that the size of an I-frame is much larger than that of an inter-frame, the workload in MERVS might not be well separated since the reliable channel requests might exceed the resource limit of the TCP channel. The way to balance the workload separation of both the TCP channel and the UDP channel should be carefully considered.

In the following sections, we explain in more details the aforementioned problems and present solutions for them.

### 3.2.1 Priority Queue

TCP transmissions need to wait for the acknowledgement of previous messages to be ready to transmit, as defined by its congestion window mechanism. Different from TCP, UDP does not constrain the message transmissions. The coexistence of these two different behaviors may cause a negative effect on the transmissions, as depicted in Figure 3.8 that shows a message sequence of a video stream, in which the message ID varies from 1 to 20. Messages 1, 2, 3, 11, 12 and 13 are in grey box, representing the I-frames and to be sent through the TCP channel. The other messages are going to be sent through the UDP channel. We assume that the TCP congestion window size has size three and messages 1, 2 and 3 are transmitted at the same time. The ACKs of messages 1, 2 and

3 are received after starting the transmission of messages 11, 12 and 13. Therefore, the transmission of messages 11, 12 and 13 is postponed because of the delay to receive the ACKs. In the waiting queue, messages will be pushed into the queue one by one, like **Queue A**. Because of the delayed ACK, messages 11, 12 and 13 are not ready to be transmitted because of the congestion window mechanism. However, the UDP channel will not wait for the TCP channel and keep pushing the new messages to the waiting queue, like **Queue B**. The ACK is received after message 18, and messages 11, 12 and 13 become ready to be transmitted at the same time. Therefore, they are pushed into the waiting queue, like **Queue C**. However, the queue is already occupied by UDP messages, which should be transmitted after messages 11, 12 and 13. In this case, the transmissions of TCP segments are postponed. This delay adds to the following TCP transmissions, because TCP transmissions need to wait for the ACKs of the previous congestion window ready to be transmitted. The correct transmission schedule should be the same as **Queue D** in this scenario.

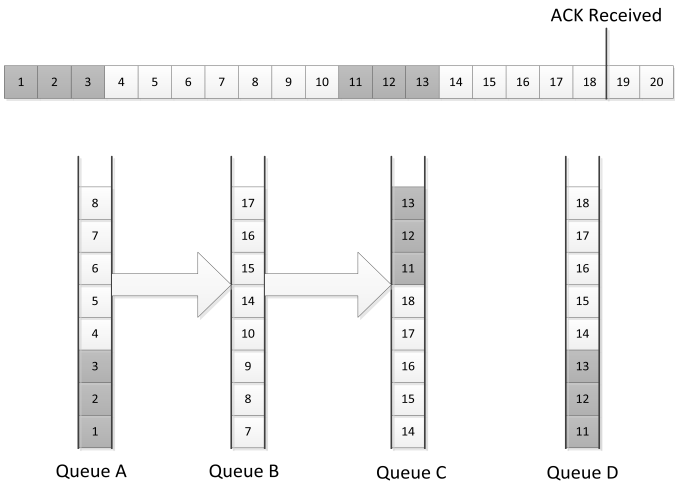



Figure 3.8: Negative effect caused by different behaviors of TCP and UDP

The disorder of both TCP and UDP channels is caused by the priority issue. Despite the fact that TCP and UDP messages have the same priority in the waiting queue, TCP messages are slower than UDP messages most of the time. Video messages should have a priority based on their sequence ID to maintain the correct video sequence. The different mechanisms of TCP and UDP confuse the order of video messages in the waiting queue. Therefore, a sorting algorithm should be used to correct that order. The priority queue that we proposed has the same cost of the sorting algorithm. There-

fore, the computational complexity of the priority queue for the insertion performance is $O(n \log n)$, implemented by the Quicksort, and the computational complexity of the pull performance is $O(1)$.

### 3.2.2 Quick-Start for TCP

The degradation of TCP over wireless networks is caused by the assumption that all delivery failures are the result of congestion, which means the sender transmits too fast and the receiver carries too much data in buffer. This assumption is correct in wired networks in most of the cases, because of the stable network environment. However, this assumption is incorrect in wireless networks, because the causes of the packet loss vary, such as fading, interference, and contention. In a VANET, another cause of the delivery failure is the fast motion of vehicles, which leads to network disconnection.

Different implementations of TCP act differently, when a delivery failure occurs. TCP treats both timeout (TO) and triple duplicated ACK (TD) as TCP segment losses. Each host sets initially the congestion window size to a multiple of the local maximum segment size, and TCP begins the slow start phase. In our analysis, we assume that the initial congestion window size is 1. In the slow start phase, even though the congestion window size exhibits an exponential growth when receiving successful ACKs in sequence, it still needs to take $\log_2(N)$ Round Trip Times (RTTs) to let the congestion window size become $N$ segments. During the recovery of the congestion window, TCP segments are delayed because of the small congestion window. Some implementations treat TD similarly to TO, like TCP Tahoe. Other implementations halve the congestion window size and begin the congestion avoidance phase. The increment of the congestion window size is linear, which is slower than the slow start face. Therefore, it also takes time to recover the congestion window size back to its initial size before a delivery failure.

If the delivery failure is not caused by congestion, the TCP congestion control is unnecessary. There is an optional Quick-Start mechanism [38] for TCP, in which the corresponding entities, i.e., sender, receiver and cooperated routers, negotiate the start transmission rate together. Sender can request a higher transmission rate, which should be approved by the routers and receiver. Routers and receiver decide the suitable transmission rate for each Quick-Start request, based on their capability. The transmission rate in Quick-Start respond should be small or equal to the transmission rate in Quick-Start request. If the Quick-Start request is approved, TCP will transmit up to the approved sending rate for a given data window. If the request is declined, TCP will

enter the slow start phase. With the Quick-Start mechanism, the TCP transmissions can be started in a negotiated sending rate rather than the initial sending rate. In this case, it can avoid some of the false congestion detection in TCP, by negotiation among sender, receiver and cooperative routers.

### 3.2.3 Scalable Reliable Channel (SRC)

The disorder of TCP channel and UDP channel in queue management is presented in Section 3.2.1. The priority queue is a kind of palliative treatment for the disorder. The real reason of the disorder is the limited resource of the TCP channel in wireless networks and the unbalance size of I-frames and inter-frames. To solve this problem, we should analyze the correct proportion of messages that should be transmitted in TCP channel.

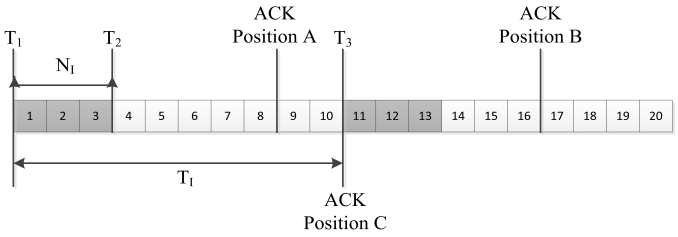



Figure 3.9: Analysis of the stream balance of TCP and UDP

#### Analysis of TCP capability

In Figure 3.9, there is a sequence of video stream messages with IDs varying from 1 to 20. Messages in a grey box are going to be sent in the TCP channel, whereas the other messages in the UDP channel. The first I-frame, in messages 1, 2 and 3, will start the transmission at $T_1$. The following inter-frame messages, from 4 to 10, will start the transmission at $T_2$. The next I-frame, in messages 11, 12 and 13, will start the transmission at $T_3$. If the ACKs of messages 1 to 3 are received at **ACK position A** before $T_3$, the TCP channel will be empty after receiving the ACKs, because messages 11 to 13 have not started the transmissions yet. In this case, the resource in TCP channel is wasted. If the ACKs of messages 1 to 3 are received at **ACK position B** after $T_3$, the transmissions of messages 11 to 13 will be delayed, like the analysis in Section 3.2.1. Therefore, in order to efficiently utilize the TCP channel, the correct position to receive the ACKs of messages 1 to 3 is **ACK position C**, which is right at $T_3$. In this case, when the ACKs are received, messages 11 to 13 are ready to transmit at the same time.

If from $T_1$ to $T_2$ there are $N_I$ I-frames that need to be transmitted in TCP channel, the time to transmit those messages, $T_I$, should be equal to $T_3 - T_1$, which is the time of the Group of Picture (GOP), $T_{GOP}$, in Eq. 3.1. Based on this analysis, we can decide how many messages can be transmitted in one $T_{GOP}$, with the $RTT$, congestion window size $CWND$ and message delivery ratio $p$. When the delayed ACK timer is small, the time between the departure of the first TCP segment and the arrival of its ACK is close to one $RTT$ [39]. Because in one $RTT$, the ACK of the first segment in a window will be received, the next window will start the transmission at the same time. A round of a window can be approximately measured as one $RTT$. The number of rounds can be conducted in one $T_{GOP}$ can be calculated by $\frac{T_{GOP}}{RTT}$. We assume that the congestion window size $CWND$ is almost the same in one $T_{GOP}$, because Quick-start maintains the sending rate, even though delivery failures occur. Therefore, $CWND \times p$ can calculate the number of the TCP segments that can be successfully transmitted in one window. The unsuccessful transmitted segments will be retransmitted in the next window. Eq. 3.2 shows how to calculate the corresponding number of packets that can be transmitted in on $T_{GOP}$.

$$T_I = T_3 - T_1 = T_{GOP} \tag{3.1}$$

$$N_I = \frac{T_{GOP}}{RTT} \times CWND \times p \tag{3.2}$$

**Scalable Reliable Channel**

The number of TCP segments that can be transmitted in one $T_{GOP}$ is decided by the $RTT$, $CWND$, $p$ and $T_{GOP}$. However, the number of messages in one I-frames is fixed. To scale the stream transmitted in the TCP channel, one solution is the spatial scalability of Scalable Video Coding (SVC) of H.264 [117], which separates a frame into different quality layers. If an extra layer is received, it will improve the quality of the video frame. Theoretically, spatial scalability of H.264 can support different resolution ratio between the consecutive layers, which can produce suitable video size to be transmitted in the TCP channel. However, the encoding complexity of spatial scalability is simplified only in the case of traditional dyadic solution, only when the resolution of two consecutive layers exhibits a 2:1 ratio. Another resolution ratio between subsequent layers may need a more complicated mathematical operation. In other words, the computational complexity will be higher, if the resolution ratio of consecutive layers does not present a 2:1 ratio [61].

Therefore, in order to propose a low computational complexity algorithm to decide the stream that should be transmitted in a TCP channel, instead of processing the spatial separation of I-frames, our solution, Scalable Reliable Channel (SRC), processes the video messages directly. Figure 3.10 indicates that there is a message sequence from 1 to 13, which is the encoded message for an I-frame. Based on Eq. 3.2, the number of messages $N_I$, which can be transmitted in a TCP channel, can be calculated. $N_I$ messages will be selected from this message sequence to be sent in a TCP channel. The selected messages will be uniformly distributed in order to maintain the certain quality of the video.

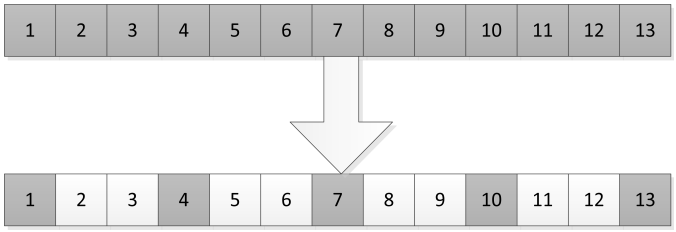



Figure 3.10: Selection of the packets transmitted in TCP channel

SRC has a very low computational complexity, because the number $N_I$ can be easily calculated and the message is selected only from I-frames. In contrast, spatial scalability of SVC needs to process the whole video stream, which increases its encoding delay.

### 3.2.4 Summary

In this section, we analyze the causes of the delay of MERVS, and a set of solutions is proposed. A priority queue is proposed to solve the disorder of both TCP and UDP channels in the waiting queue. The Quick-Start mechanism is integrated with the TCP to maximize the throughput of the TCP channel by eliminating the negative effect of the congestion control. A Scalable Reliable Channel (SRC) solution is presented to the video message scheduling and assignment to both TCP and UDP channels, in order to balance the transmission of the two channels. SRC and Quick-Start are the main treatments for the disorder of the two channels and the delay of MERVS, and the priority queue solution ensures that TCP can access a network correctly even if an disorder occasionally happens.

## 3.3 Performance Evaluation

In order to measure the performance of our protocol in the scenario of VANETs, several simulations are conducted on NS-2 [75]. We compare the following protocols for video streaming: RTP/UDP, FEC, MERVS, MERVS with Priority Queue and Quick Start and the MERVS with Priority Queue, Quick Start and SRC. The parameters of the simulation are presented in Table 3.3. The simulation is conducted in NS-2 with EvalVid, a video quality evaluation tool-set, and SUMO, a simulation for urban mobility. EvalVid is responsible for generating the video stream for the transmission and evaluation of the quality of the transmitted video stream. SUMO is responsible for generating the urban mobility model for NS-2. The generated mobility model is based on the downtown area of Ottawa, Canada. The SUMO mobility model includes street capacity, traffic light and speed limit. AODV is chosen to be the routing protocol. Two scenarios are examined: the V2I scenario and V2V scenario. The metrics we measure are the Peak Signal-to-Noise Ratio (PSNR), total time for video transmission, jitter and receiving data rate. PSNR and jitter are calculated to measure the quality of the transmitted video. The total timer for video transmission is measured to indicate the delay of the protocols. The quality of the video streaming is defined by the data rate of the receiver, and, thus, we measure the receiving data rate to compare the performance of each protocol. The scenario we consider in the simulation is a real-time video streaming, with a fixed transmitting data rate for the selected video clip.

In the V2I scenario, the vehicle requests a video transmission from a wired node in the network. Therefore, the video stream is transmitted through the access point to the vehicle that starts the request. There are 16 access points deployed in the area based on a uniform distribution. Figure 3.11 show the simulation result. When the signal gets weaker, the video quality of the UDP transmission is decreased, but the video quality of MERVS is still very stable because the transmissions of I frames are guaranteed. If the delay improvement algorithms is integrated, like the Priority Queue and Quick Start, it can maintain a video quality more than 40 dB in terms of PSNR in almost the whole video. The increment on the video quality of all transmissions at the last part is caused by the stable connection to a new access point. In this scenario, the transmission time of all protocols are the same, because most of the transmissions are in the wired networks and most of the wireless transmission are in one hop connection in the urban scenario. If the network is stable in the V2I scenario, the performance of all protocols are the same. However, if the network condition gets worse, the performance of MERVS with the delay

| Parameter | Value |
|---|---|
| Number of Vehicles | 100 |
| Area | 2300 m × 2000 m |
| Video play time | 139 s |
| Video Resolution | 352288 |
| Measured metrics | Peak Signal-to-Noise Ratio |
| | Structural SIMilarity (SSIM) index |
| | Total time to transmit the video |
| | Jitter |
| | Receiving data rate |
| Scenarios | V2I |
| | V2V |
| Comparative algorithms | MERVS |
| | UDP |

Table 3.3: Simulation parameters of the VANET scenarios

improvement algorithm has the best result in terms of PSNR. The average PSNR of the whole video is shown in Table 3.4, which is a summary of Figure 3.11. Table 3.4 clearly shows that MERVS with Priority Queue and Quick Start can achieve a much higher average PSNR compared with UDP and MERVS.

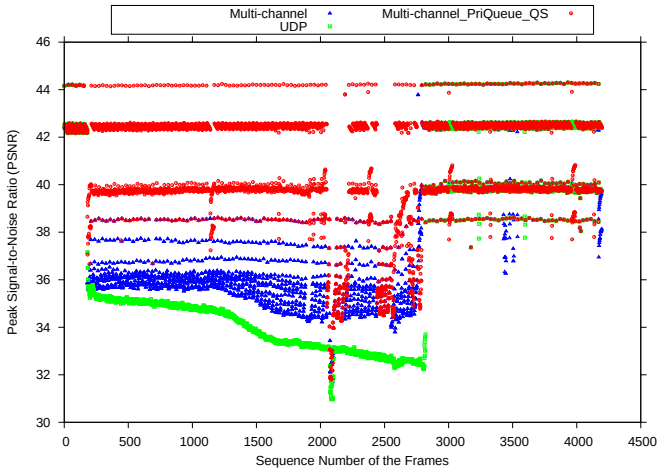



Figure 3.11: Simulation results for V2I scenario

| Protocol | Average PSNR (dB) |
|---|---|
| UDP | 36.67 |
| Multi-channel | 37.92 |
| Multi-channel with PriQueue QS | 40.61 |

Table 3.4: Average PSNR of the V2I scenario

In the V2V scenario, a vehicle requests a video transmission from another vehicle, which carries the video in the buffer. There are 36 access points deployed in the area using a uniform distribution allowing a better connection for the video transmission. All protocols are tested in this scenario, and the results are presented from Figure 3.12 to Figure 3.14. Figure 3.12 shows the resulting PSNR of the four protocols: RTP/UDP, MERVS, MERVS with Priority Queue and MERVS with both Priority Queue and Quick Start. The result of UDP keeps decreasing because of the less quality control on the video transmission. Only three fourths of the video are transmitted by UDP. The result of MERVS is not good as well, which is mainly caused by the delay of the transmission. It will be discussed in the delay measurement. After applying the Priority Queue technique, the PSNR of the transmitted video gets a certain improvement. However, the improvement is still limited by the performance of congestion control of TCP. With the Quick Start technique, the total quality of the transmitted video is highly improved, because of the number of the unnecessary decrements of the congestion window is dramatically decreased. The PSNR of the transmitted video can reach more than 35 dB, which is generally considered to be good . Table 3.5 shows the average PSNR of the whole video. MERVS with both Priority Queue and Quick Start can achieve an average PSNR closed to 35 dB, which is considered to be a good quality.

| Protocols | Average PSNR (dB) |
|---|---|
| UDP | 22.77 |
| Multi-channel | 24.76 |
| Multi-channel PriQueue | 27.07 |
| Multi-channel PriQueue QS | 34.33 |

Table 3.5: Average PSNR of the V2V scenario

The problem of using FEC in high-quality and real-time video transmissions is dis-

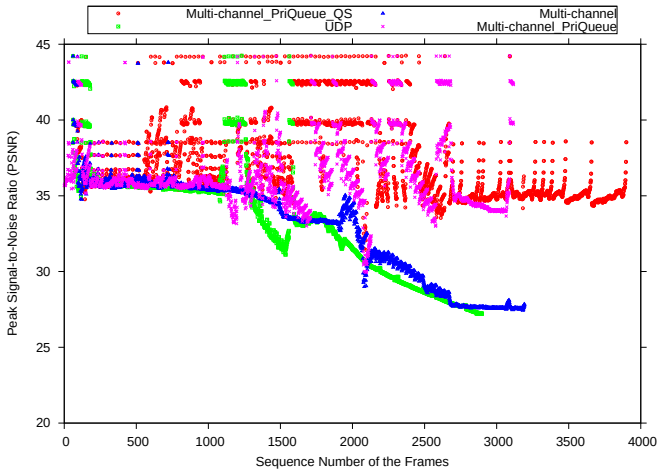



Figure 3.12: Simulation results for V2V scenario

cussed in Section 2. The comparison simulation is conducted in our research as well. We implement a FEC mechanism, which adds redundancy to the I-frames only. Because I-frames are important for decoding, if the transmissions of I-frames are confirmed, the quality of the transmitted video should be improved. Figure 3.13 presents the simulation results comparing FEC and MERVS with both Priority Queue and Quick Start. Even the FEC mechanism should have a better result compared with UDP, which maintains a PSNR about more than 33 dB. However, the total quality of the video is still lower than that of MERVS with delay improvement. Similar to UDP, FEC only transmits successfully about three fourths of the video. The probability of the successful transmitted I-frames should be higher in FEC. However, before about the 1700th frame, no fully transmitted I-frames are observed. The reason of the limited performance of FEC is the burst of messages, and the situation is worse in high-quality video streaming, as discussed in Section 2. The average PSNR of the whole video is presented in Table 3.6, summarizing the results in Figure 3.13. FEC has a higher video quality compared with RTP/UDP and MERVS. However, its quality is still very low compared with the MERVS with both Priority Queue and Quick Start mechanism.

| Protocol | Average PSNR (dB) |
|---|---|
| FEC | 25.12 |
| Multi-channel PriQueue QS | 34.33 |

Table 3.6: Average PSNR of the V2V scenario

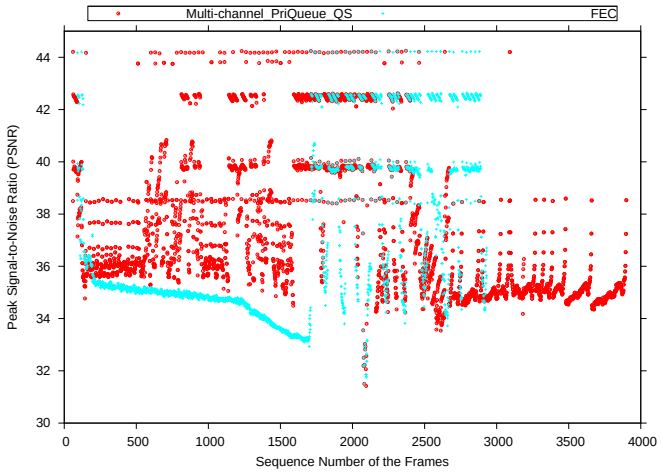



Figure 3.13: Simulation results for V2V scenario of FEC and Multi-channel Solution

Even though the quality of the transmitted video is good enough in MERVS with both Priority Queue and Quick Start, but the transmission delay is still very high. Figure 3.15 presents the total transmission time of the whole video of different protocols. The transmission time of TCP is presented to illustrate that if the whole video is transmitted in fully TCP, the delay will be huge. MERVS reduces the delay, because part of the video is transmitted in the UDP channel instead. The Priority Queue and Quick Start techniques almost halve the time to transmit the whole video compared with MERVS. However, it still doubles the time transmitted in UDP. To further improve the delay, the Scalable Reliable Channel (SRC) is proposed in this research, which adjusts the stream that needs to transmit in the TCP channel. Even though SRC improves the delay, but it still needs to be able to ensure the quality on the video transmission. Figure 3.14 shows the PSNR results of MERVS with SRC and without SRC, and both protocols are integrated with Priority Queue and Quick Start techniques. It is obvious that MERVS without SRC can have higher PSNR in most of the cases, because more messages with I-frames are transmitted in the TCP channel. However, after about the 2500th frame, SRC can achieve a better quality compared with non-SRC. The reason of this situation can be explained in Figure 3.15. The confidence intervals are also plotted at a confidence level of 95%. SRC needs less time to transmit the video. Therefore, SRC avoids a certain degradation of the network, which might be caused by the interference or connection failure. In Figure 3.15, the time to transmit the whole video with UDP and SRC is very similar. However, the quality of the transmitted video is much higher than that transmitted through UDP. SRC also performs better than FEC in terms of quality in our

simulations. The average PSNR of the whole video is presented in Table 3.7, summarizing the results of Figure 3.14. SRC has a higher average PSNR than the MERVS with both Priority Queue and Quick Start mechanism. This is because SRC achieves a higher video quality after the 2500th frame by avoiding a certain network performance degradation.

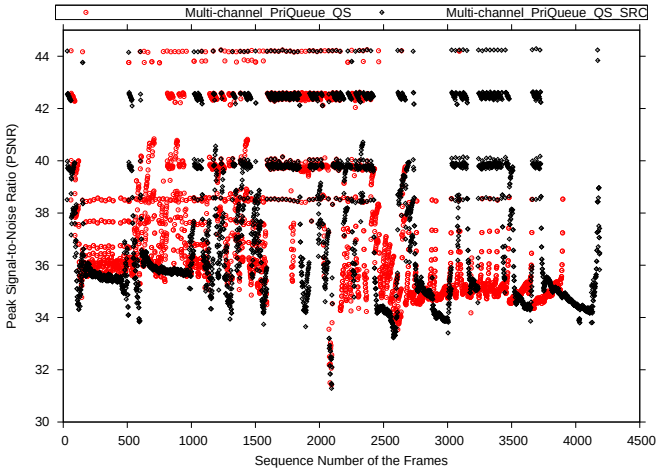



Figure 3.14: Simulation results for V2V scenario of SRC and non-SRC protocol

| Protocol | Average PSNR (dB) |
|---|---|
| Multi-channel PriQueue QS SRC | 37.20 |
| Multi-channel PriQueue QS | 34.33 |

Table 3.7: Average PSNR of the V2V scenario

In the following, we discuss the jitter of each protocol. Table 3.8 presents the average jitter of each frame in a video streaming. SRC has the smallest jitter on the frame, which is even lower than the jitter of UDP. Therefore, the message scheduling of SRC does not just improve the delay of the TCP channel, but also further improves the delay of UDP. Table 3.9 shows the maximum jitter of each protocol, in which the maximum jitter of SRC is much higher than that of UDP. This means that in some cases, SRC might still have a large jitter on some frames. However, all the other frames present a low jitter, which maintains a small average delay in the overall video transmission.

Figure 3.16 presents the receiving data rate of each protocol. The confidence intervals are also plotted at a confidence level of 95%. Based on the result, MERVS with Priority Queue, Quick Start and SRC presents the best receiving data rate in our simulations,

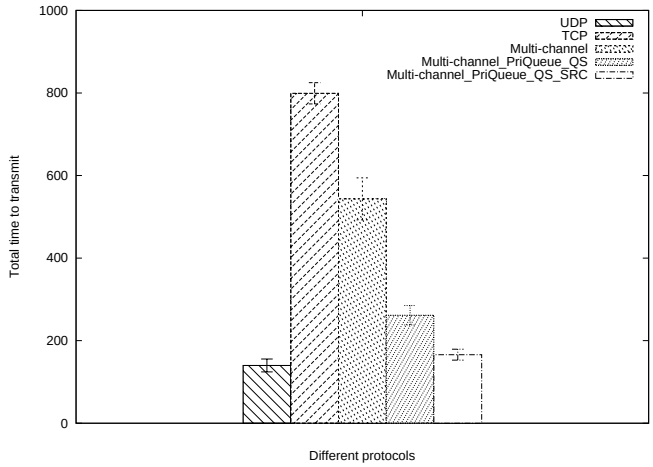



Figure 3.15: Total time to transmit the video

| Protocols | Average jitter [s] |
|---|---|
| Multi-channel PriQueue QS SRC | 0.035 |
| UDP | 0.046 |
| FEC | 0.099 |
| Multi-channel PriQueue QS | 1.668 |
| Multi-channel | 2.960 |

Table 3.8: Average jitter of each protocol

| Protocols | Maximum jitter [s] |
|---|---|
| Multi-channel PriQueue QS SRC | 12.524 |
| UDP | 2.078 |
| FEC | 4.598 |
| Multi-channel PriQueue QS | 117.812 |
| Multi-channel | 200.363 |

Table 3.9: Maximum jitter of each protocol

because it is able to properly schedule the transmission of the video messages and also provides a reliable transmission on I-frames. RTP/UDP also gives a high receiving data rate, because the delay of the RTP/UDP is very low. However, most of the successful received data is not from I-frames, which might be not important. The receiving data rate of FEC is not very high, because it generates message burstiness by creating redundant information. MERVS and MERVS with only Priority Queue and Quick Start have a low receiving data rate, even though their PSNR and successful received frame are high. This is because of their delay on video transmission.

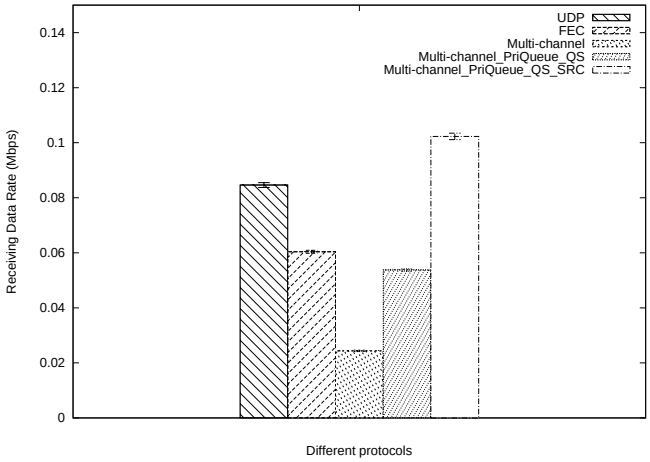



Figure 3.16: Receiving data rate of each protocols

To analyze the quality degradation of the gaussian blur [116] we evaluate the efficiency of PSNR. There is also a problem in PSNR when it compares a noise-free transmitted video frame to the original video frame, because the Mean Squared Error (MSE) is zero. The Structural SIMilarity (SSIM) index [115] shows a higher sensitivity degree to image degradation compared with PSNR. Therefore, SSIM is also measured for each simulation in order to generate a more precise and comprehensive quality measurement of this research. Figures 3.17 to 3.19 present the SSIM results of the simulations of the V2V scenario. Figure 3.17 shows that SSIM of MERVS with Priority Queue and Quick Start is higher and stabler than UDP, MERVS and MERVS with only Priority Queue, which is consistent with the result of PSNR in Figure 3.12. Figure 3.18 discusses the results of SSIM index of FEC and MERVS with Priority Queue and Quick Start, which is also similar to the PSNR result in Figure 3.13. There is an observation of the increment on video quality of FEC between the 1100th frame and 1300th frame, which is not observed in the result of PSNR. Figure 3.19 also demonstrates a similar result of

Figure 3.14, in which MERVS with Priority Queue and Quick Start can achieve a higher video quality compared with SRC in the 3000th frame, and after that SRC behaviors better because it compresses the transmission time and avoids the degradation of the network performance.

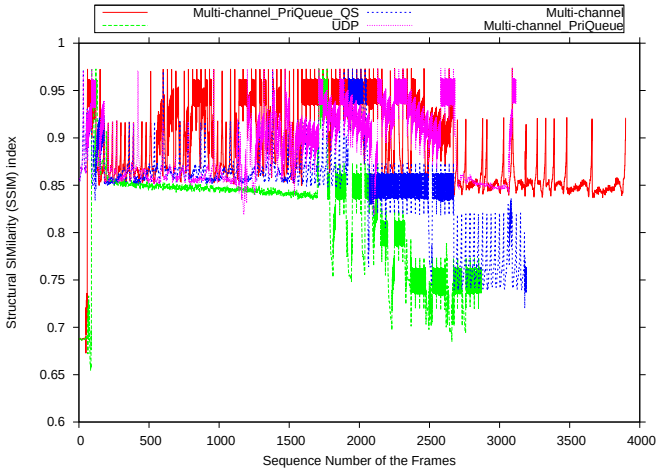



Figure 3.17: SSIM for V2V scenario

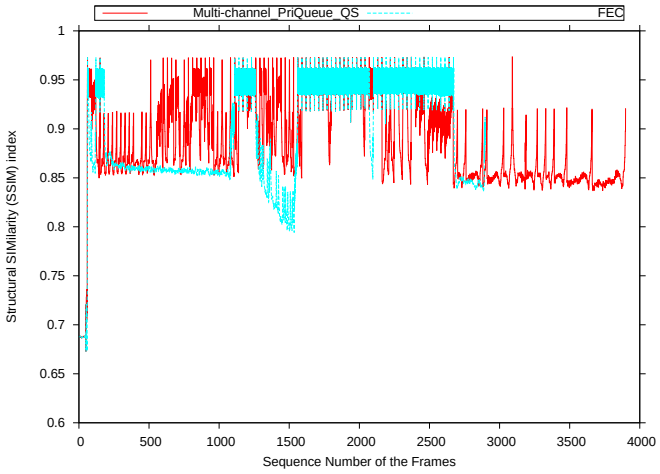



Figure 3.18: SSIM for V2V scenario of FEC and multi-channel solution

In summary, the simulation results show that MERVS presents a delay problem in Tables 3.8 and 3.9. Therefore, several delay improvement techniques are integrated: Priority Queue, Quick Start and Scalable Reliable Channel (SRC). The results shows that the delay improvements do not only enhance the delay, but they also improve the

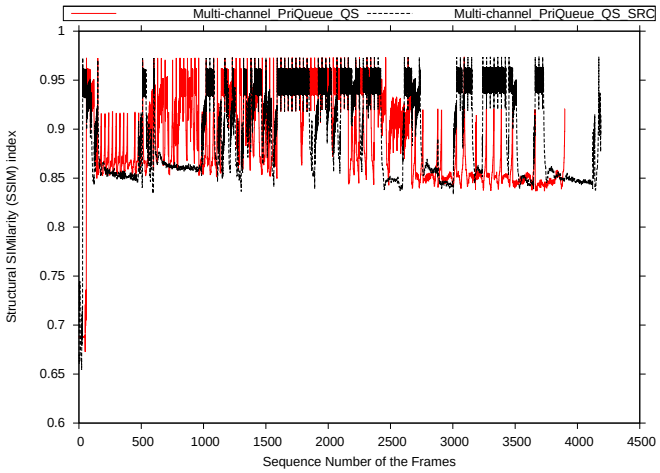



Figure 3.19: SSIM for V2V scenario of SRC and non-SRC protocol

quality of the transmitted video. FEC is also tested in our scenarios. However, the performance of FEC is not as good as that of MERVS with delay improvement. SRC shows a very low average delay on video streaming, but the transmitted video quality is slightly lower than that of non-SRC protocol, because of the filter on the I-frame messages. The degradation of the video quality of SRC is still accepted, because the PSNR is still above 35 dB.

# Chapter 4

# Multi-path Error Recovery Video Streaming (MERVS) on VANETs

A multi-path solution with disjoint algorithm is proposed to minimize the interference and contention, which leads to a higher transmission rate and an acceptable delay. In this research, Only I-frame are transmitted through TCP protocol, and the inter-frames are transmitted through UDP protocol. To improve the delay of TCP transmissions, a TCP-ETX protocol are integrated to select the suitable path for TCP transmissions. A more detail TCP-ETX routing protocol will be explained in Chapter 5. Simulations are conducted and several results are presented by comparing to other protocols. Based on the simulation results, the designed multi-path solution protocol can provide higher quality of video with a reasonable delay than that of the other protocols.

## 4.1 Introduction

Accompanying the development of the wireless networks and the mobile devices, there is a growing demand to improve the driving experience in the vehicles. A Vehicle Ad-hoc Network (VANET) creates connections among vehicles, in order to provide safety and infotainment services. Because of the equipped on-board techniques, the communications of VANET supports both Vehicle-to-Vehicle (V2V) connections and Vehicle-to-Infrastructure (V2I) connections. The On-Board Unit (OBU) on the vehicles are more powerful than the normal mobile devices and is comprised of Central Processing Unit (CPU), Graphics Processing Unit (GPU) and battery supply. However, the highly dynamic topology is a new challenge for the design of VANET services. The V2V video

transmission is mainly considered in this work, because the V2V connection is more common in both urban and highway scenarios. Besides, the V2V connection presents a higher delay and resource limitation when compared to the V2I connection.

One of the most valuable VANET services are the video streaming, which can provide a more comprehensive and appealing information to the users. The transmitted video can be related to the driving safety, such as the accident ahead or animals crossing the road. It can also be related to on-board communications, such as the vehicle-to-vehicle or vehicle-to-office video conference. On-board infotainment can also be available, such as the advertisement provided by shops along the road or other vehicles. Therefore, the video streaming service can significantly improve the on-board experience in vehicles.

The challenges of the video streaming in VANETs are the limited resource of the wireless networks, the highly dynamic topology and the large amount of video data. These problems become worse on high quality video streaming, because of the even larger amount of video data. Therefore, in this research, a protocol providing high quality video streaming will be proposed. To ensure the quality of the transmitted video, most of the recent studies are based on Forward Error Correction (FEC), like [106, 29, 86, 104, 60, 95, 119]. The concept of FEC is to create duplicated packets for the part of the data, which is required to ensure the transmissions. The problem of the FEC mechanism is the increased amount of data transmitted through the networks. Because the amount of video data is already large, the extra video data might exceed the network capability. Therefore, in this research, the retransmission mechanism technique is used to ensure the video quality. The concept of retransmission mechanism is to retransmit a packet when a transmission failure is detected. Therefore, the extra retransmissions only occur when the transmission fails.

In this research, the ARQ retransmission mechanism of TCP is chosen to insure the transmissions of the key parts of the video. The problem of the retransmission mechanism is the delay caused by the retransmissions. To minimize its the delay, only the significant video data will be transmitted using a retransmission mechanism technique. The H.264/MPEG-4 AVC compression technique [117] is chosen in this research, which include both temporal scalability and spatial scalability on video compression. The visible video frames can be encoded as I-frames, P-frames or B-frames. An I-frame contains the full information of a visible frame, which can be decoded independently. P-frames and B-frames are inter-frames, which only includes partial information of the visible frames and, thus, need the reference frames to be decoded. Therefore, most of the frames are decoded directly or indirectly by the I-frames. Therefore, the transmissions of

the I-frames should be ensured to improve the quality of the whole video stream. In this research, the Transmission Control Protocol (TCP) [85] is used for I-frame transmissions. The retransmission mechanism mechanism of TCP can ensure the transmission of the I-frames, which leads to an improvement of the video quality. The inter-frames are transmitted through User Datagram Protocol (UDP) to reduce the delay of the video stream.

Most of the recent routing metrics do not consider the difference between TCP and UDP transmissions. Therefore, the performance of TCP is not optimized in most of the cases. Expected Transmission Count (ETX) [32] is one of the most popular routing metric to evaluate the quality of a path, by estimating the number of total transmissions along a path. However, it fails to evaluate a path for TCP transmissions, because it does not consider the retransmission of the TCP mechanism. In our research, the TCP-ETX is adopted to improve the TCP performance.

The wireless contention and interference issues can be improved by splitting traffic [57]. Several multi-path video streaming techniques [57, 127, 54, 100] have been proposed. Flow splitting technique is proved to be able to improve the video quality and reduce the delay of video stream. In order to reduce the delay of the retransmission mechanism in TCP transmissions and to further improve the video quality, a multi-path solution is proposed, which is integrated with a disjoint algorithm in this research. The I-frames and the inter-frames are transmitted in separated paths, which are selected by the disjoint algorithm. The disjoint algorithm achieves node disjoint and link disjoint between the paths of TCP flow and UDP flow. The path of TCP flow is also selected by a TCP-ETX routing metric, which is able to estimate the number of transmissions to successfully deliver a TCP segment. This routing metric helps findind a suitable path for TCP transmissions.

## 4.2 Multi-path Video Streaming

In this section, we present the design of a multi-path video streaming technique. Multi-path transmissions increase the resource of the transmission by splitting the flow in different paths. An important issue in video streaming is the high data rate required. By splitting the flow into different paths, it is able to achieve load balancing during the transmission, especially during the video streaming. Most of the recent studies on multi-path video streaming mainly focus on path selection algorithm. They rarely consider the video data, which is transmitted over the path. Besides the path selection, the type of

the video data distributed in different paths and the protocol that each type of video data should be transmitted on should be carefully considered. In our work, we have designed a two-path video streaming protocol by splitting the video stream in two different flows: the intra-frame flow and the inter-frame flow.

In MPEG compression, video frames are mainly defined as I-frames, P-frames and B-frames. An example of the MPEG video stream is presented in Figure 4.1. I-frames are encoded with the information of an entire video frame, without any reference frames. It can be decoded independently to retrieve the visible video frame. P-frames are encoded based on the previous I-frame or P-frame, which have to be decoded with the reference video frame. B-frame depends upon both directly previous and directly following I-frame or P-frame. Therefore, both P-frames and B-frames are predicted frames based on the reference frames. A Group of Pictures (GOP) is composed of an I-frame and the following P-frames and B-frames, as indicated in Figure 4.1. The I-frame is the direct reference frame or indirect reference frame of the P-frames and B-frames in the GOP. This is because P-frame can be a reference frame but it is also a predicted frame based on other reference frame. If we chase the source of the prediction, we will reach an I-frame, which is not independent upon any reference frames. Therefore, if an I-frame is damaged or lost, the whole GOP might be damaged. If the transmission of a I-frame is ensured, the quality of the whole GOP can be improved.

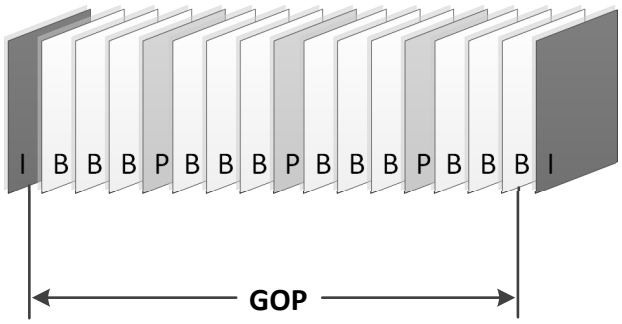



Figure 4.1: Example of MPEG video frames structure

In order to preserve the video quality during the transmission, the type of video data transmitted on different paths should be classify based on their priority. Because of the importance of the I-frame in a GOP, it should have higher priority on accessing the network resource comparing to P-frames and B-frames. In our design, it separates the transmission of I-frames and the inter frames into two different paths: the primary path and the secondary path. I-frames are transmitted on the primary path, and P-frames and

B-frames are transmitted on the secondary path. The primary path has higher priority to choose the suitable path than the secondary path, because I-frames have the higher priority than the other frames.

The transmission protocols to distribute different types of video data should also be various. Because of the importance of the I-frame in a GOP, I-frames should be transmitted on a reliable protocol to preserve its completeness. Many reliable transmission protocols are proposed in terms of error recovery during video transmission. Forward Error Correction (FEC) is one of the most popular techniques to ensure the video quality during the transmission. FEC attempts to recover the important video data by duplicating the data during the transmission, which increases its receiving probability. However, the duplicated video data leads to the increment of the data amount, which is needed to transmit over the network. Because of the limitation of the network resource of wireless network and the large data amount of the video streaming, the increment of the data amount might exceed the network capability. The packet burst caused by the FEC may also lead to extra interference and collision of the network. In this case the transmission of the P-frames and B-frames may also be affected.

In our design, it transmits the I-frames in Transmission Control Protocol (TCP), and the P-frames and B-frames are transmitted in User Datagram Protocol (UDP). The reason to use TCP in I-frame transmissions is that TCP only duplicates the video data if the data is identified as lost. Therefore, the duplicate data is limited at a low level. The drawback of TCP transmission is also obvious. TCP is known as low performance in wireless networks, because of its delay on wireless transmissions. In our design, the delay of TCP is reduced by the splitting flows. The P-frames and B-frames is less important comparing to I-frames, so they are transmitted in regular RTP/UDP protocol to avoid extra resource consumption. The two paths are also chosen based on the link disjoint and node disjoint manner in order to reduce the interference and collision.

Figure 4.2 represents the system structure of the multi-path video streaming solution. I-frames are transmitted in the primary path and the other frames are transmitted in the secondary path. The path decision phase is designed at the network layer. The routing protocol is designed based on Ad Hoc On Demand Distance Vector (AODV) [81], which is one of the most popular routing protocol in Vehicular Ad-hoc Network (VANET). Routing packets are handled by the routing request handler and the routing reply handler. Paths will be generated based on the routing packets inside the Disjoint Agent. In our design, two kinds of paths will be generated: the primary paths and the secondary paths, which will be stored in different routing tables. Any new discovered

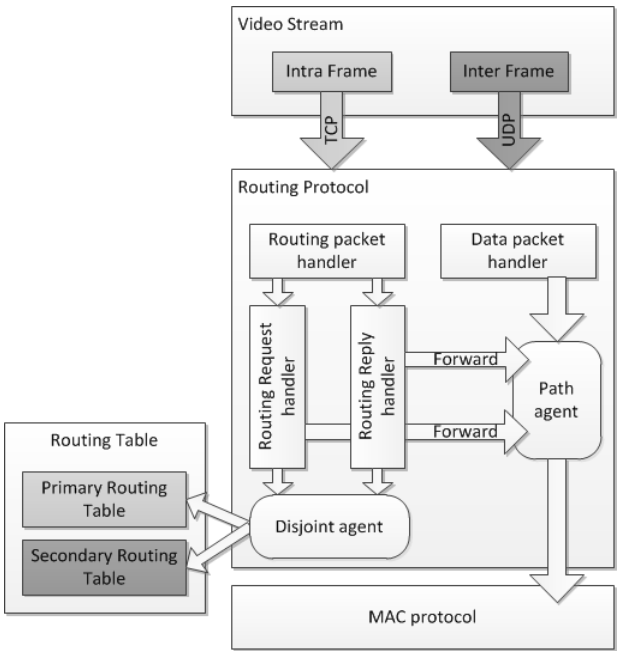



Figure 4.2: Structure of the multi-path video streaming protocol

path is compared to the existing path in the Disjoint Agent in order to update the routing table. The Path Agent is responsible to select a suitable path for the forwarding routing packets and the data packets. After the route is decided, Path Agent forward the packets to the Media Access Control (MAC) Layer. The detail description of the routing protocol will be presented in the following section.

One of the main concerns of the flow splitting of I-frames and the inter-frames in a video stream is the computation complexity. There is a patent [103] publishes in 2013, which contains a video frame parser that can separate the I-frames and the inter-frames from the video stream. After the separation, it stores the I-frames and the inter-frames in different folders for fast trick play. If this technique is used in our solution, the computation complexity to split the video frame flows will be minimized. A unique automatic increment ID will be assigned to each packet, when the packet is generated. At the receiver side, two flows are sorted by the ID before the combination in order to minimize the computation complexity. After the sorting, two flows will be scanned and combined to generate the final video stream. The total computation complexity of the regeneration of the video stream is $O(n \log n) + O(n)$, where $n$ is the size of the video stream.

## 4.3 Route Discovery Protocol

In this section, the route discovery protocol of the multi-path video streaming solution will be discussed. The components of route discovery protocol in Figure 4.2 will be also discussed in detail.

### 4.3.1 Routing Discovery Protocol based on AODV

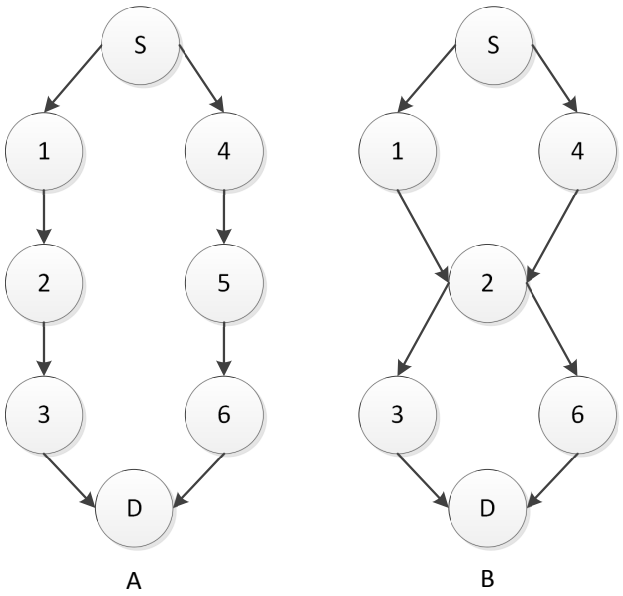



Figure 4.3: Example of node disjoint and link disjoint

The designed route discovery protocol is based on AODV, which is one of the most used ad-hoc routing protocol in VANETs. AODV contains mainly three types of routing packets: routing request (RREQ), routing reply (RREP) and routing error (RERR). If a source node can not find a route in the routing table, it will initiate a RREQ to its one-hop neighbours. If a node receives an RREQ, it firstly stores the reverse path to the routing source node in order to forward back the RREP when a path is found. If an intermediate node that received the RREQ has the route to the destination, it generates a RREP immediately and forwards it to the routing source node directly. If an intermediate node that received the RREQ does not have the route to the destination, it forwards the RREQ to its one-hop neighbours. When the destination node receives the RREQ, it generates a RREP and sends it back to the source node. RERR will only be broadcasted, when a route fails during the transmission.

| Seq | Dest | Path | TCPETX |
|---|---|---|---|
| 2 | D | 1, 2, 3, D | 4.3213 |
| 1 | 3 | 1, 2, 3 | 3.7642 |
| 1 | 4 | 4 | 1.200 |

Table 4.1: Primary routing table of node S in Figure 4.3.b

| Seq | Dest | Path | Hops |
|---|---|---|---|
| 2 | D | 4, 2, 6, D | 4 |
| 1 | 3 | 4, 2, 3 | 3 |

Table 4.2: Secondary routing table of node S in Figure 4.3.b

The routing table of AODV contains the sequence number, destination node, next hop and number of hops of each path. If a node receives a new path to the destination with a higher sequence number, it will notice that it is a new path from a new routing request. Thus, it replaces the current path with the new path immediately. If the new path has the same sequence number as the current path, AODV will choose the one with less hops. Instead of storing merely the next hop, the designed route discovery protocol in this work stores the whole path from origin to the destination, so we have information to define a disjoint path. Because TCP is used to transmit I-frames, the routing metric should be reconsidered, rather than the minimum hop algorithm. The minimum hop metric is not sufficient to decide its quality on transmitting TCP segments. A TCP-ETX metric will be integrated to our solution to decide the path quality in the primary routing table. TCP-ETX is an improvement of ETX routing metric, which can calculate the number of the expected transmissions to successfully delivery a TCP segment along a path. The TCP-ETX algorithm will be explained in Section 5.2. Tables 4.1 and 4.2 present an example of the routing table of node $S$ in Figure 4.3.B.

The main purpose of our routing discovery protocol is to discover a primary path and a secondary path for video transmission, which are link disjoint or node disjoint. The example of the node disjoint path and link disjoint path is shown in Figure 4.3. Figure 4.3.A represents a node disjoint situation, in which path $S, 1, 2, 3, D$ and path $S, 4, 5, 6, D$ have no common nodes. Figure 4.3.B is a link disjoint example, in which there is a common node 2 in path $S, 1, 2, 3, D$ and path $S, 4, 2, 6, D$ but there is no common links in these two paths. The benefits of disjoint path are load balancing and reliability.

Many researches on multi-path data transmissions provide disjoint path over the networks . However, most of the solutions are centralized solution based on the analysis of the graph theory. Frequent control messages are required to maintain the information of the global topology. In VANET, because of the highly dynamic topology, it is expensive to maintain the global topology information. Therefore, a distributed solution is more efficient than a centralized solution in VANET. In next section, a distributed route discovery protocol with node disjoint and link disjoint algorithm will be proposed.

### 4.3.2 Routing Request Handler

In AODV, RREQ is also responsible for the reverse path discovery. When a node receives a RREQ, it will record the reverse next hop to the routing source of the RREQ. In our route discovery protocol, the whole path is needed to be stored in the routing table so we can apply the disjoint algorithm. Therefore, each node has to append itself to the RREQ during the route discovery process. The RREQ handling algorithm is presented in Algorithm 2. After appending itself to the RREQ, the protocol attempts to add the reverse path to the primary routing table. If the path does not exist in the primary table, it will add the new path into the primary routing table. If a path to the RREQ routing source exists, the protocol compares if the new path has a greater destination *seq*, or the RREQ has the same *seq* but smaller *TCPETX*. If the new path is newer or it has a better quality than the current path, it will replace the current path in the primary routing table. If the new path is not newer and is not better than the current path, it will be abandoned. The replaced path or the abandoned path will not be dropped immediately, and the protocol will attempt to add it to the secondary routing table by function $addRouteToSRT(TempRoute1, TempRoute2)$, which is shown in Algorithm3. $TempRoute1$ is the route going to be add to the secondary routing table, and the $TempRoute2$ is the route to do the disjoint comparison. First of all, it will check if the dropped path and the path in the primary routing table are disjoint. If they are disjoint, the dropped path is qualified for the secondary routing table. To add the path to the secondary routing table, the procedure is almost the same as that of the primary routing table. One difference is that the secondary routing table measures the number of *hops* instead of the *TCPETX*.

After adding the reverse path, another task of the protocol is to discover the path to the routing request destination. If a path to the routing request destination exists in the primary routing table of the current node or the current node is the routing destination, it

will send a RREP to the routing source directly. If the path does not exist in the primary routing table of the current node, it will forward the RREQ to one hop neighbours. The reason that we only look at the primary routing table in forwarding the RREQ is the high priority of the primary routing table. If a path does not exist in the primary routing table, it will not exist in the secondary routing table as well. If both tables contain the path to the destination, we only need to return the primary path. The reason is that the broadcasting of RREQ can discover all the alternative paths.

The computation complexity of Algorithm 2 is mainly caused by the *look_up* function of the routing table and the *disjoint* function of disjoint agent. The computation complexity of the *look_up* function is $O(n)$. The computation complexity of the disjoint agent will be discussed in the following section. The best case scenario of the Algorithm 2 is when there is no existing path in the primary routing table to the destination. Therefore, no *disjoint* function is conducted and only the *look_up* function is conducted, whose computation complexity is $O(n)$. The worst case scenario is when there are paths in both tables for the destination. The cost of the worst case scenario is the same as the average case, in which the computation complexity depends on the *disjoint* function.

### 4.3.3 Routing Reply Handler

Because the requirements of the node disjoint is stricter than that of the link disjoint and the collision probability of the node disjoint paths is much lower than that of the link disjoint path, the route discovery protocol primarily chooses the node disjoint paths as the secondary path. If the node disjoint path does not exist, the link disjoint path will be chosen instead. When there are two paths: the first one with length $n$ and the second one with length $m$, the complexity to decide if two paths are node disjoint by comparing all the nodes in two paths is $O(nm)$. To decide if two paths are link disjoint by comparing all links, the complexity should be $O((n-1)(m-1))$. Based on the implementation of the ad hoc network infrastructure, this complexity can be further reduced. If the addresses of the nodes in the ad hoc networks are sortable, it is better to sort the paths before the comparison. The average complexity of Quicksort is $O(n \log n)$. Based on the two sorted paths, we compare the nodes in the node comparison algorithm in Algorithm 5.

### 4.3.4 Disjoint Agent

Because the requirements of the node disjoint is stricter than that of the link disjoint and the collision probability of the node disjoint paths is much lower than that of the link

**Algorithm 2** Algorithm for RREQ

**Require:** $primary\_rtable.length \geq 0$

**Require:** $secondary\_rtable.length \geq 0$

**Require:** $RREQ \neq 0$ RREQ should not be empty

1: $calTReverseTCPETX(RREQ)$
2: $RREQ.append(self.addr)$
3: $primary\_rtable.look\_up(RREQ.rt\_src) \rightarrow temp$
4: **if** !$temp$ **then**
5: $primary\_rtable.add\_rt(RREQ)$
6: **else**
7: **if** $RREQ.seq > temp.seq$
**or** $(RREQ.seq == temp.seq$
**and** $RREQ.TCPETX < temp.TCPETX)$ **then**
8: $addRouteToSRT(temp, RREQ)$
9: $primary\_rtable.upd(temp, RREQ)$
10: **else**
11: $addRouteToSRT(RREQ, temp)$
12: **end if**
13: **end if**
14: **if** $primary\_rtable.look\_up(RREQ.rq\_dst)$
**or** $RREQ.rq\_dst == self.addr$ **then**
15: $sendRREP(RREQ)$
16: **else**
17: $forward(RREQ)$
18: **end if**

**Algorithm 3** Add route to the secondary routing table $addRouteToSRT(TempRoute1, TempRoute2)$

**Require:** $TempRoute1 \neq NULL$
**Require:** $TempRoute2 \neq NULL$
1: **if** $disjoint(TempRoute1.route, TempRoute2.route)$ **then**
2: $secondary\_rtable.look\_up(TempRoute1.rt\_src)$
$\rightarrow temp2$
3: **if** $!temp2$ **then**
4: $secondary\_rtable.add\_rt(TempRoute1)$
5: **else**
6: **if** $TempRoute1.seq > temp2.seq$
**or** $(TempRoute1.seq == temp2.seq$
**and** $TempRoute1.hops < temp2.hops)$ **then**
7: $secondary\_rtable.upd(temp2, TempRoute1)$
8: **end if**
9: **end if**
10: **end if**

disjoint path, the route discovery protocol primarily chooses the node disjoint paths as the secondary path. If the node disjoint path does not exist, the link disjoint path will be chosen instead. When there are two paths: the first one with length $n$ and the second one with length $m$, the complexity to decide if two paths are node disjoint by comparing all the nodes in two paths is $O(nm)$. To decide if two paths are link disjoint by comparing all links, the complexity should be $O((n-1)(m-1))$. Based on the implementation of the ad hoc network infrastructure, this complexity can be further reduced. If the addresses of the nodes in the ad hoc networks are sortable, it is better to sort the paths before the comparison. The average complexity of Quicksort is $O(n \log n)$. Based on the two sorted paths, we compare the nodes in the node comparison algorithm in Algorithm 5.

The average complexity of the above algorithm is $O(n \log n) + O(m + n)$, where $m < n$. Besides the node disjoint, link disjoint can also be improved by a similar algorithm. Because the link in a path is directed, it is possible to sort the link at the source node. If the source nodes of two paths are the same in any step of the comparison, the destination nodes of the two links will be compared. If both the source nodes and the destination nodes are the same, it can be determined that the link is the same, and these two paths are not link disjoint. The complexity of the link disjoint should be the same as

**Algorithm 4** Algorithm for RREP

**Require:** $primary\_rtable.length \geq 0$

**Require:** $secondary\_rtable.length \geq 0$

**Require:** $RREP \neq 0$ RREP should not be empty

1: $calTCPETX(RREP)$
2: $suppress\_reply \leftarrow false$
3: $primary\_rtable.look\_up(RREP.rt\_dst) \rightarrow temp$
4: **if** !$temp$ **then**
5: $primary\_rtable.add\_rt(RREP)$
6: **else**
7: **if** $RREP.seq > temp.seq$
**or** $(RREP.seq == temp.seq$
**and** $RREP.TCPETX < temp.TCPETX)$ **then**
8: $addRouteToSRT(temp, RREP)$
9: $primary\_rtable.upd(temp, RREP)$
10: **else**
11: $suppress\_reply \leftarrow true$
12: $addRouteToSRT(RREP, temp)$
13: **end if**
14: **end if**
15: **if** !$suppress\_reply$ **then**
16: $forward(RREP)$
17: **else**
18: $drop(RREP)$
19: **end if**

**Algorithm 5** Algorithm for node disjoint

**Require:** $path1.length > 0$
**Require:** $path2.length > 0$
1: $sort(path1)$
2: $sort(path2)$
3: $i, j = 0$
4: **while** $i < path1.length$ or $j < path2.length$ **do**
5: **if** $path1[i] = path2[j]$ **then**
6: $return(false)$
7: **else if** $path1[i] < path2[j]$ **then**
8: $i++$
9: **else if** $path1[i] > path2[j]$ **then**
10: $j++$
11: **end if**
12: **end while**
13: $return(true)$

that of the node disjoint. Therefore, the computation complexity of Algorithms 2 and 4 is also $O(n \log n) + O(m+n)$. The best case scenario of Algorithm 5 is when the number of nodes in one path is smaller than in the other path. The computation complexity is $O(n)$, because only one path is scanned. The computation complexity of the best case of Quicksort is $O(n \log n)$. Therefore, the computation complexity of the best case of the disjoint algorithm is $O(n \log n)$. The worst case scenario of Algorithm 5 is that both paths are fully scanned. In this case, the computation complexity is $O(m + n)$. The computation complexity of the worst case of Quicksort, which can be avoided, is $O(n^2)$, where $m < n$. Therefore, the worst case computation complexity is $O(n^2)$.

### 4.3.5 Path Agent

Path agent is responsible for selecting the path for any forwarding packet, including routing packets and data packets. RREQ is the routing packet to initiate a routing discovery process. The path selection algorithm of the designed routing protocol is similar to AODV, when transmitting the RREQ. It simply broadcasts the RREQ to one hop neighbours. The proposed protocol will send a packet based on the reverse path inside the RREP hop by hop. The maintenance phase of the routing process is also the

same of the AODV.

There are two types of data packets in our system: TCP data packets and UDP data packets. Based on the design, TCP data packets are only transmitted over the paths in the primary routing table, and the UDP data packets are mainly transmitted over the paths in the secondary routing table. There is still possibility that for one destination there is a path in the primary routing table but there is no path in the secondary routing table. In this case, the UDP packets have to be transmitted by the path inside the primary routing table, because no available disjoint path is detected from the source to the destination in this topology. When both the primary path and the secondary path are the same path, the situation is complicated, because UDP flow can disturb the TCP flow when they are in the waiting queue and in the transmissions. This case will be discussed in the future study.

## 4.4 TCP Efficiency Routing Metric


Figure 4.4: Example of TCP transmission

TCP-ETX is an improvement of ETX routing metric to support TCP transmissions. ETX attempts to calculate the expected number of transmissions to successfully deliver a packets, in order to measure the quality of the link. In TCP, if any transmission failure happens, it will start the packet transmission again from the source node. ETX does not calculate these retransmissions of a TCP transmission after the transmission failure. Figure 4.4 is an example of a TCP transmission with a transmission failure. The transmission from node 2 to node 3 fails and node 1 detects the failure by timeout or duplicated acknowledgement, so node 1 sends the packet again. Therefore, a packet transmission failure in a link will cause the retransmissions of all the previous successful transmissions. Equation (4.1) is used to calculate the TCP-ETX of a TCP transmission from node $i$ to node $j$.

$$\begin{cases} \text{TCPETX}_{j,j} = 1. \\ \text{TCPETX}_{i,j} = \frac{1}{P_{i,i+1}}(1 + \text{N}(i)) + \text{TCPETX}_{i+1,j}. \end{cases} \tag{4.1}$$

$P_{i,i+1}$ denotes the TCP segment transmission probability of link $(i, i+1)$. In ETX, the probability of a link is calculated as bi-direction. Therefore, $P_{i,i+1}$ should be calculated as follow:

$$\mathrm{P}_{i,i+1} = P_{i \to i+1} P_{i \leftarrow i+1}. \tag{4.2}$$

$N(i)$ denotes the average number of transmitted packets caused by the failure of the following links. With the initial packet, $1+N(i)$ is the total number of packets that need to be transmitted at node $i$. Each packet may needs several number of transmissions to successfully deliver to next hop in the MAC layer, because of the failure. Therefore, $\frac{1}{P_{i,i+1}}(1+\mathrm{N}(i))$ is the total number of transmissions at node $i$ to node $i+1$. The $TCPETX$ of the whole path should be the accumulation of the total number of transmissions of all nodes along the path. Therefore, the $\mathrm{TCPETX}_{i+1,j}$ should be added in Equation (4.1). The calculation of $N(i)$ is also an accumulation from the destination node. The way to calculate $N(i)$ is presented as follow:

$$\begin{cases} N_{Destination} = 0 \\ N_{Destination-1} = 0 \\ \mathrm{N}(i) = (1 - P_{i+1 \to i+2}) + \mathrm{N}(i+1). \end{cases} \tag{4.3}$$

TCP-ETX is calculated from the destination node to the source node in a path. Therefore, in the routing request handler, it calculates the TCP-ETX for the reverse path. In the routing reply handler, it calculates the TCP-ETX for the path from the routing request source node to the routing request destination node. For example, if node 5 sends a RREP for the RREQ initiated by node 1 in Figure 4.4, the RREP will follow the reverse path $5, 4, 3, 2, 1$ and eventually reaches node 1. The TCP-ETX calculation of the path $1, 2, 3, 4, 5$ will be initiated at node 5. When RREP reaches node 4, it stores the path $4, 5$ in the routing table for destination node 5 and calculate the $TCPETX_{4,5}$ based on the $TCPETX_{5,5}$ and $N(5)$ forwarded from node 5. After that node 4 sets the corresponding $TCPETX_{4,5}$ and $N(4)$ in the RREP, and forwards the RREP to node 3. When node 3 receives the RREP, it first stores the path $3, 4, 5$ in the routing table and it also calculates $TCPETX_{3,5}$ and $N(3)$ based on $TCPETX_{4,5}$ and $N(4)$ from node 4. After the calculation, node 3 sets the RREP and forwards it to the next hop. The process will stop when node 1 receives the RREP from node 5 and the corresponding $TCPETX_{1,5}$ is calculated. The process to calculate the TCP-ETX value of a reverse

path is in a similar approaches, but the calculation of the $TCPETX$ is from node 1 to node 5 and the $TCPETX$ and $N(i)$ is forwarded inside the RREQ.

## 4.5 Performance Evaluation

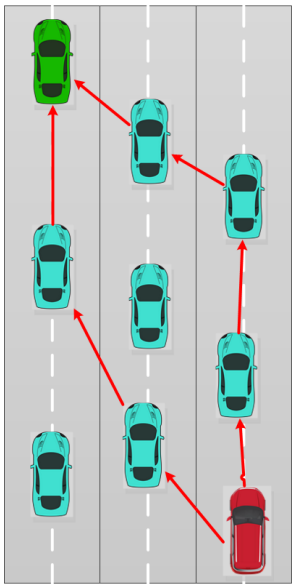


Figure 4.5: Example of simple lanes scenario

To evaluate our protocol, we simulate two scenarios using NS-2 [75] and SUMO [6]. NS-2 a is mature network simulator, which can simulate network communication and network traffic. SUMO is a vehicle traffic generator, which can generate the real traffic on the road and provide mobility model for NS-2. The first scenario of the performance evaluation is a simple lane scenario, in which all vehicles are on one direction of the road and multiple lanes are supported like Figure 4.5. The number of lanes is 3 and around 50 vehicles are running on the lanes in the simple lane scenario. During the simulation, a video is transmitted from the source vehicle to the destination vehicle. The length of the simple lane scenario is 2500 m. The speed of each vehicle is randomly chosen from 45 km/h to 55 km/h.

The second scenario is a urban scenario, which is based on the downtown region of Ottawa, Ontario, Canada. The topology layout is shown in Figure 4.6. In the urban scenario, a source vehicle will transmit a video to all vehicles in a distance. The simulation results are based on the average of all vehicles in the networks. All traffic is generated

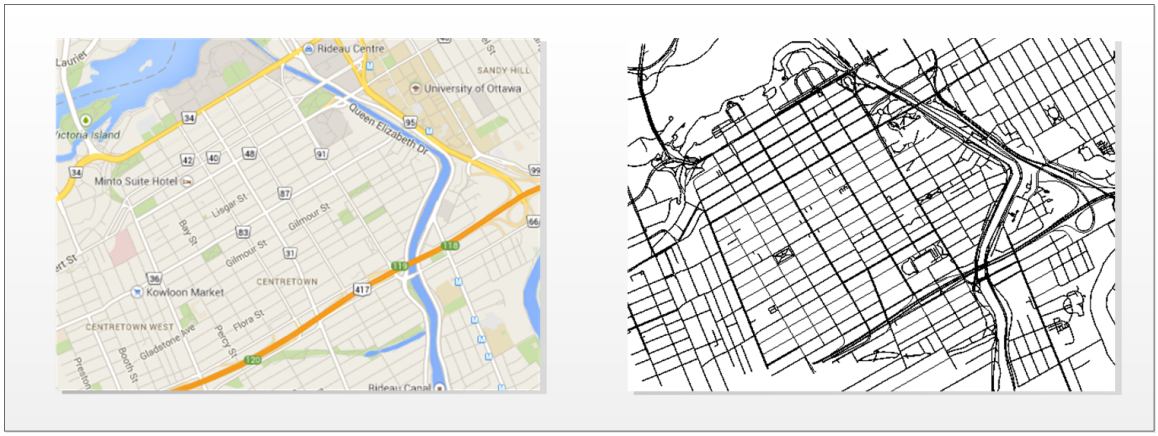



Figure 4.6: Map of urban scenario

by the random trip model of SUMO, which is the mobility model for NS-2. In the urban scenario, the numbers of vehicles in the simulations are increased from 100 to 500 in order to measure the performance in different network densities. The area of the urban scenario is 2300 m × 2000 m.

The chosen transmitted video is bridge-far_cif(139 seconds) rather than akiyo_cif(9 seconds), in order to test the long term effect of different protocols. The measured metrics are Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity(SSIM) index, jitter and receiving data rate. PSNR, SSIM and jitter are calculated to measure the quality of the transmitted video. The receiving data rate is also measured, since the video quality is defined by the transmission rate of the receiver. The transmitting data rate is fixed in our simulation because the real-time video streaming is tested. All simulations are ran around 20–25 times in order to retrieve the average simulation results. The confidence intervals are also plotted with a confidence level of 95%. The criteria of the simulations and measured metrics are presented in Table 4.3. There are three protocol that are compared in the simulations: the multi-path solution with disjoint algorithm, the Forward Error Correction (FEC) and the RTP/UDP. The Two-Ray Ground Radio Propagation Model, with a communication of 150 m, is chosen.

In the simple lane scenario, the PSNR, SSIM are calculated based on the receiver side. Jitter is the average jitter of each frame of the transmitted video. Receiving data rate is calculated through the total received video size divided by the total transmission time. In the urban scenario, the simulation results are the average results of all nodes that actually received the video streaming. The calculation of each metric is same as that of the simple scenario.

| Parameters | Value |
|---|---|
| MAC | IEEE802.11p |
| Simple lanes length | 2500m |
| Urban area | 2300m × 2000m |
| Video play time | 139s |
| Video Resolution | 352×288 |
| Measured metrics | Peak Signal-to-Noise Ratio |
| | Structural SIMilarity (SSIM) index |
| | Jitter |
| | Receiving data rate |
| Scenarios | simple lanes |
| | urban |
| Comparative algorithms | Multi-path solution |
| | FEC |
| | UDP |

Table 4.3: Simulation parameters of VANET scenarios

### 4.5.1 Results

In this section, the results of the simulation will be presented and discussed. The results of the first scenario that is going to be presented is the simple lanes scenario. The video transmission starts about 50s after the simulation starts. The transmission lasts 139s, which is same as the play time of the video. There are 50 vehicles in total running in the simple lanes scenario simulation. The simulation results are shown in Figure 4.7 to Figure 4.9.

PSNR is one of the most widely used metrics to measure the quality of the video after the transmission. The PSNR results comparing different protocol are presented in Figure 4.7. Because SSIM is known to be more sensitive to image degradation and more consistent with human eye perception comparing to PSNR, the SSIM results are also plotted in Figure 4.8. From both the PSNR result and the SSIM result, it can be clearly seen that the multi-path solution with disjoint algorithm has better video quality than that of the FEC and UDP. The reason is that the multi-path solution use TCP transmissions to ensure the transmission of the packets of the I-frames, which is the most importance frames in one GOP. The successful delivery of the I-frame maintains a certain quality of the other visible frames, which are generated by the predicted frames: P-frames and B-frames. FEC also has the mechanism to ensure the transmission of the important frames, by duplicating the important frames. Therefore, FEC can achieve higher video quality of both PSNR and SSIM than that of UDP. However, the burst of transmitted packets caused by the duplicated packets of FEC leads to extra packet loss in the video transmission, which limits its improvements of quality. This drawback even worst in VANET, because of the limited network resource and high dynamic topology.

The data receiving rate is another metric to indicate the capability of the successfully received data. Figure 4.9 shows that multi-path solution has a higher data receiving rate comparing to FEC and UDP. The reason is that the multi-path solution uses TCP instead of FEC to ensure the transmissions, which reduces the number of actual transmitted packets over the network. Meanwhile, the disjoint algorithm decreases the contention and collision probability, which leads to a higher success rate of the delivered data. In the extra analysis of the results, the successfully delivered I-frames packets of the multi-path solution is also higher than that of the FEC and UDP.

The delay of the video transmission is another critical metric in the real-time vide streaming. The delay has to be in an acceptable range of the human perception. The average delay for the different protocols is presented in Figure 4.10. The multi-path solu-

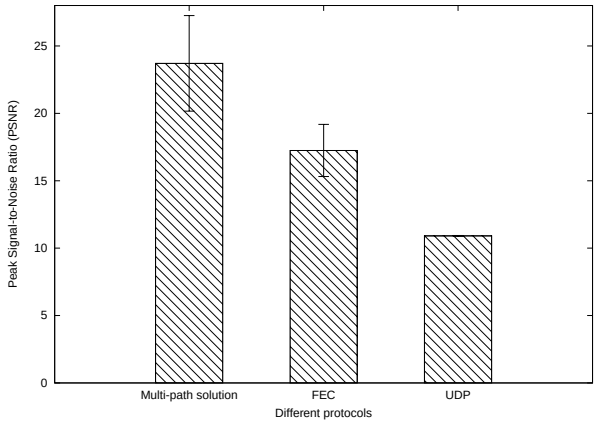



Figure 4.7: PSNR of the simple lanes scenario

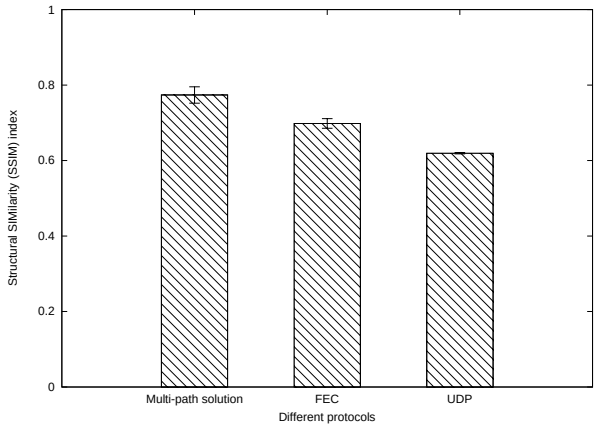



Figure 4.8: SSIM of the simple lanes scenario

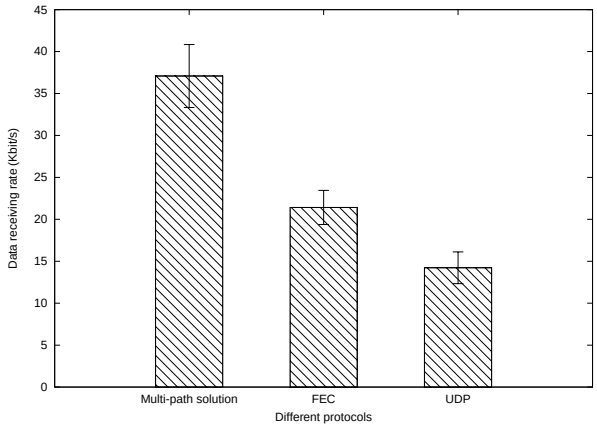



Figure 4.9: Receiving data rate of the simple lanes scenario

tion has the highest average delay, which is caused by the TCP transmissions. Actually, most of the delayed packets are TCP packets, and delay is the main problem of TCP as discussed in the previous section. FEC has a higher average delay than that of UDP because of the extra transmitted packets. Even though the multi-path solution has the highest average delay, it is still lower than 0.5 second, which is reasonable for the human perception in most of the cases. However, the average delay should still be improved in the future research.

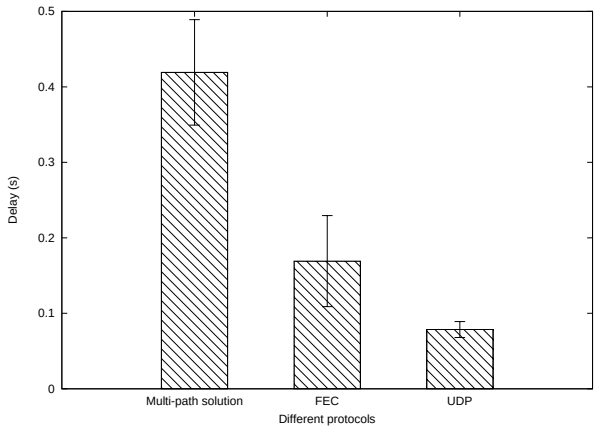



Figure 4.10: Delay of the simple lanes scenario

In the urban scenario, the topology is based on the map of downtown Ottawa. To help create the connections among the vehicles, a set of Road Side Units (RSU) are deployed uniformly in the simulation area. There are 99 RSUs in the simulations of the urban scenario. Because this work mainly considers the V2V connection, only ad-hoc routing is enabled. The simulations are run for different number of vehicles, in order to test the performance of the protocols for different vehicle densities. The simulation results presented in this section are the average results of all nodes that actually received the video streaming. The received data in the relay nodes is not counted in the result. The video quality observed in Figures 4.11 and 4.12 shows that a multi-path solution has the highest average PSNR and average SSIM comparing to that of FEC and UDP. In the simulation results, the video quality increases, when the number of vehicles increases from 100 to 300. The reasons is that the increased density of vehicles helps creating extra connections for the video streaming. The video quality decreases, when the number of vehicles increases from 300 to 500. One possible reason is that the increased density of the vehicles near the source vehicle increases the number of video streaming requests during the simulation. Another possible reason is that the density of vehicles cause

the link to be saturated because of the broadcasting routing packets of AODV. In the simulation results presented in [49], the increment of the routing overhead from 50 nodes to 100 nodes in a 100 m × 100 m area is very low. Even though the maximum receive range in [49] is only 25 m, the density of its proposed scenario is much higher than our scenario. Therefore, the possibility that the link be saturated because of the routing packets is rare. The video quality of the UDP decreases faster, because it does not have any mechanism on video quality guarantee. Even though the number of video streaming request increases, the multi-path solution still can achieve a higher average video quality comparing to FEC and UDP in the simulation.

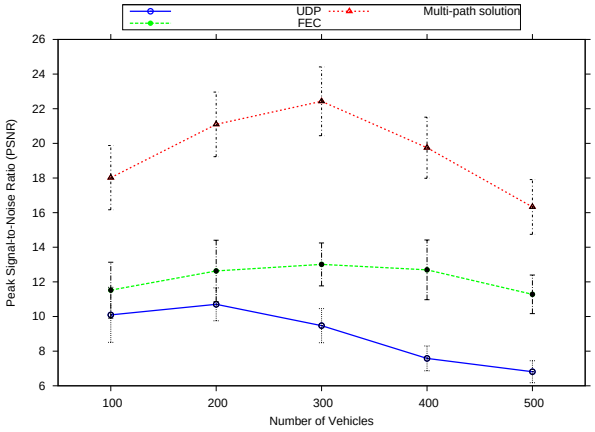



Figure 4.11: PSNR results of the urban scenario

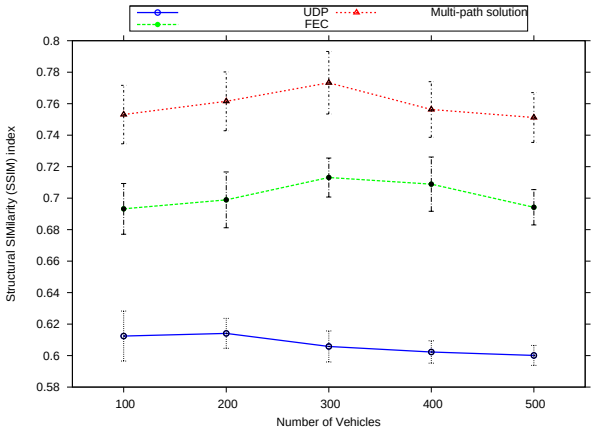



Figure 4.12: SSIM results of the urban scenario

In order to analyze the data rate of different protocols, the receiving data rate is also plotted in Figure 4.13. Receiving data rate can measure the number of the successfully

transmitted packets in the receiver side, which helps measuring the performance of the protocols. The simulation results show that multi-path solution has a highest average receiving data rate among the tested protocols. A higher receiving data rate is one of the factors that leads to the higher video quality.

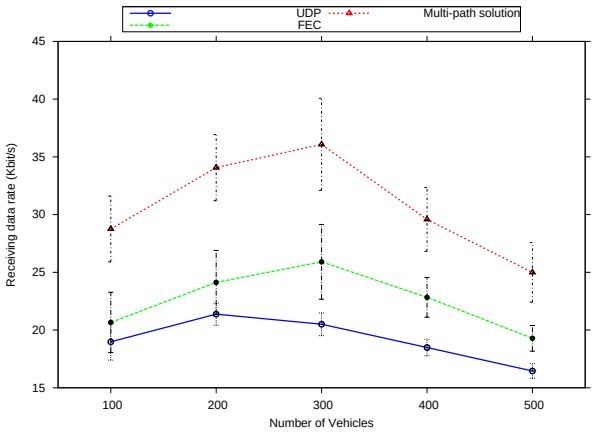



Figure 4.13: Receiving data rate of the urban scenario

The delay of the urban scenario simulations is similar to that of the simple lane scenario. The delay of the multi-path solution is still the highest of all protocols. However, the average delay is still limited to 0.5 s. Most of the delay is also caused by the TCP transmissions. If the delay of the TCP transmissions can be improved, the delay of the multi-path solution can also be improved.

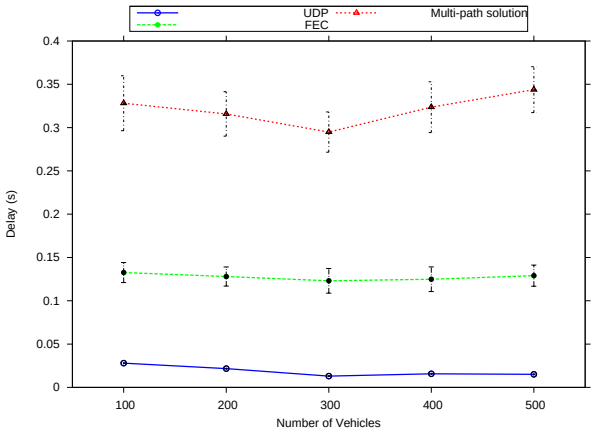



Figure 4.14: Delay of the urban scenario

## 4.6 Summary

The purpose of this research is to achieve high quality video streaming on VANET. Most of the current researches uses FEC to ensure the quality of the video during the transmissions. However, because of the limited resource of wireless networks, the dynamic changed topology of VANET and the large amount of video data, the duplicated packets created by FEC causes extra interference and contention during the video streaming. To avoid generating extra packets, ARQ is used to ensure the video transmissions in this research instead. The main problem of ARQ is the delay caused by each packet loss. To solve this problem, only selected data will be transmitted with the ARQ technique. Because of the importance of the I-frames, only the packets of the I-frames will be used ARQ to ensure the transmissions. In our solution, the TCP protocol is used to transmit the I-frames and UDP protocol is used to transmit the inter-frames. To further minimize the delay, the I-frames and inter-frames are transmitted in separate paths with the node disjoint and link disjoint algorithm, in order to reduce the probability of the packet loss caused by interference and contention. Because of the different mechanism of the TCP and UDP, different routing metrics are chosen to select the paths. A TCP-ETX routing metric is integrated to calculate the expected number of transmissions to successfully deliver a TCP segment, which can evaluate the quality of a path for TCP transmissions. UDP transmission still uses the regular shortest path routing metric. To evaluate the designed scheme, several simulations are conducted by comparing to the FEC and UDP/RTP protocol. The simulation results indicate that the multi-path solution can achieve the highest video quality and receiving data rate in the simulation, but the multi-path solution also suffer higher delay comparing to the other two protocols. Even though the average delay is reasonable for the human perception in most of the cases, the delay should be improved in the further research.

# Chapter 5

# Cross Layer Routing Scheme

Besides MERVS, lower layer improvement is also an optional technique to improve the performance of the video streaming. If the resource of the transmission path are guarantee, the video stream will be more stable and efficient. In this chapter, two cross layer routing scheme are proposed. One is the cross layer delay sensitive wireless routing metric for UDP channel. Another one is the TCP-ETX, which is a cross layer metric for TCP improvement. Both technique has shown its improvement on the network throughput.

## 5.1 Cross Layer Delay Sensitive Wireless Routing Metric

The choice of a suitable path for packet transmission is a fundamental issue of any routing protocol. The principle of choosing the shortest path is no longer a good option for route selection. The link quality of the path is more important than the length of the path in a wireless network, because of the channel's unstable condition. Expected Transmission Count (ETX) is a routing metric, which selects the path with fewer transmissions for a successful packet delivery. However, other factors should be also considered for route selection, like the retransmission delay. In this section, a comprehensive analysis on the effect of the link layer delay to the transmission throughput is discussed. A delay sensitive routing metric for IEEE 802.11 to measure and exam the link layer delay of each hop along the path to evaluate the quality of the path.

### 5.1.1 Cross Layer Routing Metric

Most of the routing metrics only consider the number of hops, such as DSDV, AODV and DSR. However, in wireless networks the number of hops is no longer as critical as other factors. In wired networks, the number of hops increases the probability of packet collision and bit error, and the quality of the link can be ignored in most of the scenarios since it exhibits a high quality. On the other side, in wireless networks, the quality of the link is one of the major factors that should be considered in route selection. The number of hops is still a factor for route selection, because the accumulation of the packet loss probability along the path increases when the route becomes longer. However, the effect of the link quality causes a stronger effect on the throughput of a wireless transmission, because of the interference, fading and collision. In a layered architecture, the link quality is not immediately available to the network layer protocol for the purpose of establishing a routing metric. Therefore, a cross-layer routing metric should be proposed for routing in wireless networks, such as the IEEE 802.11 standard considered in this work.

Expected transmission count (ETX) is a cross-layer routing metric to estimate the quality of a communication link. It accumulates the packet loss probability of each hop along the path to estimate the quality of the whole path. Its purpose is to reduce the number of transmissions for one packet delivery by choosing the route with the lowest packet loss probability. ETX is able to choose a high quality link for transmission with the less number of transmissions. However, it does not guarantee the highest throughput of a path. Figure 5.1 shows an example about two one-hop route, in which the source nodes are set to different sizes of the contention window (CW). The route with a large contention window size performs less transmissions to successfully deliver a packet, but if a packet is lost it needs to wait a longer time to retransmit and also needs a longer time to transmit if the channel is busy. The route with a smaller contention window size needs extra transmissions to successfully transmit a packet. However, it waits a shorter period for a retransmission and to access the communication channel. ETX will prefer the first path with a large contention window rather than the second one because of its smaller number of transmissions.

Notice that the link layer delay of the first route is higher than the link layer delay of the second one. The link layer delay can be calculated by Equation 5.1. The $E[delay_{LL}]$ denotes the link layer delay, $E[ReTr]$ the average number of retransmission for a successful packet delivery, $E[BF]$ the average backoff, $E[Def]$ the average deferring time and $E[Tr]$ the average transmission time for the last success transmission. Before each

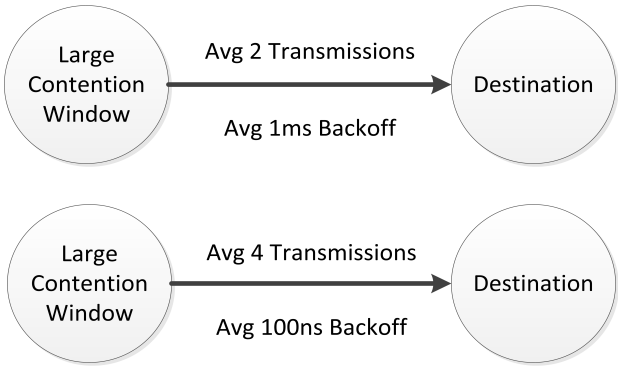



Figure 5.1: Impact of the contention window to ETX

retransmission, IEEE 802.11 will back off for a certain period to avoid another collision. If the channel is busy while the node requests a transmission it will defer for a certain time to start the request. The extra deferring time in the equation is for the last success transmission. Both $E[BF]$ and $E[Def]$ are determined by the contention window mechanism. Equation 5.1 shows that the link layer delay is determined by the average retransmission time and the contention window range. Therefore, in Figure 5.1, the link layer delay of the first route is around 5 ms, whereas of the second route is around 900 ns. The second route spends less time waiting for the retransmission, which makes it transmit faster than the first route. In case the sources of both routes have always a packet to send, the transmission interval of the first route is about 5 ms whereas of the second route is around 900 ns, which has the higher throughput. It is important to notice that this scenario cannot be identified by ETX, which only considers the packet loss probability.

$$\begin{aligned} E[delay_{LL}] &= E[ReTr] \times (E[BF] + E[Def]) + \\ &\quad E[Def] + E[Tr]. \end{aligned} \tag{5.1}$$

### 5.1.2 Link Layer Delay Analysis

As discussed in the previous section, the main factor of the transmission throughput is the link layer delay, which limits the transmission speed. In the one hop scenario, as depicted in Figure 5.1, each packet of the transmission flow has to follow a certain procedure to be successfully transmitted to next hop. The next packet should wait in the queue until the successful delivery of the current packet, which is the link layer delay. If this delay is

too large, it will affect the transmission flow by increasing the time interval between two continue packets, as shown in Figure 5.2. The transmission interval between two packets in sequence of the original flow is smaller than the link layer delay. Therefore, packet #2 needs to wait in the queue until packet #1 is successfully transmitted, and the same applies to packets #3 and #4. The transmission flow will be modified as it is affected by the link layer delay. It is better to choose a one hop route with a smaller link layer delay for the one hop example in Figure 5.1 to guarantee the transmission throughput.

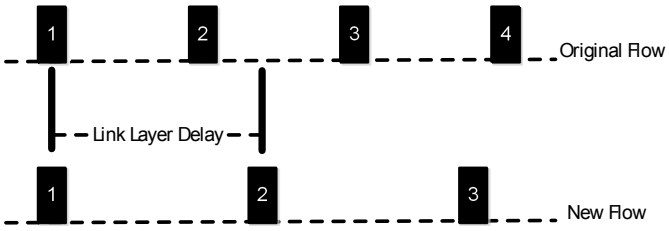



Figure 5.2: Transmission flow is affected by the link layer delay

In a multi-hop scenario, the analysis of link layer delay is similar. IEEE 802.11 only handles the packet forwarding along the path, and at a given time only one packet can be forwarded. Therefore, the packet is forwarded by a station along the path. If there is a slow transmission hop, all the following packets will be blocked, as shown in Figure 5.3. A good node means a node in a link layer with a small delay connection, whereas a bad node means a node in a link layer with a large delay connection. A good node will not affect the traffic speed, since the link layer delay is low. A bad node will change the transmission speed of a flow, since it delays the transmissions. The transmission flow after the bad node will be totally changed, and the transmission interval of the flow is equal to the link layer delay of the bad node. The transmission throughput cannot be recovered after the bad node, since the bad node blocks the traffic.

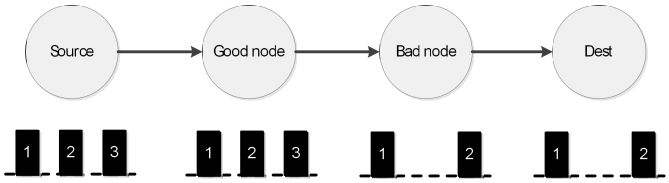



Figure 5.3: Effect of link layer delay on traffic flow along the path

Therefore, the quality of each hop should be carefully considered when choosing a transmission path. Merely accumulating the link quality of each hop is not enough. The

algorithm should ensure that the link layer delay of each hop is small enough. Roughly the transmission interval of a flow is equal to the largest link layer delay of the path.

### 5.1.3 System Architecture

The proposed routing metric is a cross layer solution that measures the link layer delay of a path and chooses the path with the smallest link layer delay. There are two layers involved in this algorithm: the network layer and the Media Access Control (MAC) layer. The network layer handles the routing process to send routing requests and routing replies. The MAC layer retrieves and stores the link quality of all one hop neighbors for the route decision in the network layer.

#### Network Layer

A basic function of the network layer is to send a "request" and a "reply". There are two pieces of information stored in the routing request and routing reply: (i) the path that the routing request is sent, and (ii) the link layer delay of the corresponding path. The link layer delay of the path is equal to the greatest link layer delay of the link in the path and is calculated as described in Algorithm 6. Whenever sending a packet, the network protocol first searches the routing cache for an existing route. If the destination is new to the current node, a routing *Request* will be broadcasted. Any other node upon receiving the routing *Request* stores its information in the *Request* and forwards the *Request* until it reaches the destination. When the destination receives the routing *Request*, it sends back a routing *Reply* by reversing the route used to transmit the packet. Each node upon receiving a routing *Reply* compares the link layer delay ($LLD$) recorded in the *Reply* and its current $LLD$ to the next hop of the path. If the $LLD$ to the next hop of the path is greater than that in the *Reply*, the current $LLD$ is replaced by the received *Reply*. If the origin node receives a routing *Reply* for the sent routing *Request*, it compares the $LLD$ of the routing *Reply* to the existing route in the routing cache, and stores the smallest value. If there is no existing route in the cache, it stores the *Reply* in the cache.

#### Media Access Control (MAC) Layer

The MAC protocol keeps measuring the link layer delay to all one hop neighbors, in order to obtain the link quality information for the routing metric. The IEEE 802.11 standard employs the messages "Request To Send" (RTS) "and Clear To Send" (CTS) to secure the channel from collisions on the data packets. There is also an acknowledgement

**Algorithm 6** Routing Algorithm

**Require:** *Request* Routing request
**Require:** *Reply* Routing reply
**Require:** *CurPath* Current path in the cache
**Require:** $LLD \geq 0$ Link layer delay
**Require:** $CurNode \neq NULL$ Link layer delay of the current hop

```
while running do
    if receive Request then
        if Request.dst = CurNode.addr then
            Send Reply
        else
            Request.add(CurNode)
            Forward Request
        end if
    else if receive Reply then
        if Reply.dst ≠ CurNode.addr then
            Reply.LLD ← max(Reply.LLD, CurNode.LLD)
            Forward Reply
        else
            if CurPath.dst = Reply.Route.dst exist then
                if Reply.LLD ≤ CurPath.LLD then
                    CurPath ← Reply
                end if
            else
                Store Reply in the cache
            end if
        end if
    end if
end while
```

message for the data packet that is successfully received. The link layer delay is defined as the time interval between a packet is available to be sent and its acknowledgement is received. The concept of Link Layer Delay is similar to the concept of Round Trip Time (RTT) of TCP. The concept of smooth RTT is adopted and is called Smooth Link Layer Delay (SLLD) and the following Equation 5.2 is obtained.

$$SLLD \quad = \quad \alpha \times LLD + (1 - \alpha) \times SLLD, \tag{5.2}$$

where $LLD$ denotes the current measured link layer delay, $\alpha$ the smooth factor in the interval $[0, 1]$ (this parameter is set to $1/8$ in this work, which is obtained from the experiments to avoid sudden change of the networks).

The MAC protocol records the time the data packet is received from the corresponding acknowledgement to calculate the $LLD$. By accumulating $LLD$, based on Equation 5.2, $SLLD$ is stored as the link quality for the route selection in the network layer.

### 5.1.4 Simulation Results

In this section, the proposal is evaluated through simulations using the Network Simulator 2 (NS-2) [75]. The protocol stack is based on the IEEE 802.11 protocol and the User Datagram Protocol (UDP). All sources in the simulation transmit a 160-byte packet in a transmission interval of 0.02 ms. The proposal is evaluated through two network topologies: chain scenario and grid scenario. In the grid scenario, a number of low quality nodes are generated to create data links with different qualities. These low quality nodes have a larger link layer delay to deliver a packet.

The evaluation is conducted by simulating the chain scenario, as depicted in Figure 5.3. There is one low quality node in the path, in which the contention window range is increased to add extra link layer delay. Figure 5.4 shows the simulation results considering different network sizes (nodes): 7, 10, 15 and 20. The goal of this simulation is to exam how large link layer delays affect the throughput of the transmission flow. As can be seen, the throughput of the transmission flow is the same or similar for all network sizes, considering the same low quality node. As discussed in Section 5.1.2, the low quality node reorganizes the transmission flow by blocking the next packets, and, thus, decreasing the throughput. In this case, the path size is not as critical as the link quality of the node.

In the grid scenario, the link layer delay based routing metric is compared to ETX and AODV. There are two transmission flows in this case, which have an intermediate node.

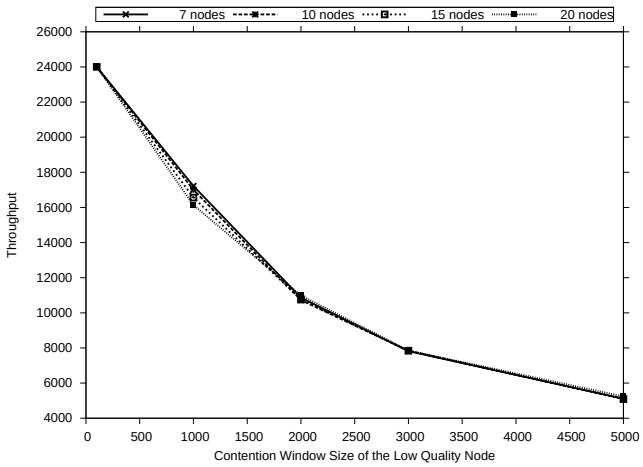



Figure 5.4: Effect of the low quality link to different network sizes

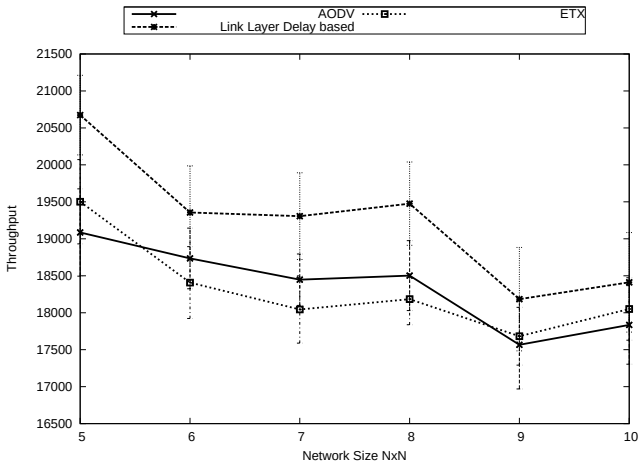



Figure 5.5: The overall throughput of the network by increasing the network size

The overall network throughput is measured in this simulation. Firstly, both the network size and the number of low quality nodes are increased (at the same rate of the network size). This simulation aims to exam the performance of the routing metric considering the link layer delay for different network sizes. Figure 5.5 shows the simulation results. The proposed routing metric considering the link layer delay exhibits a higher performance than the other two routing metrics, since it considers the link layer delay, which is one of the main limitations of data throughput. ETX presents a performance similar to AODV, because ETX only considers the number of transmissions and ignores the average time to wait for a retransmission. When the network size changes from $6 \times 6$ to $8 \times 8$, the throughput has a slight increase. This happens because there are more available good nodes in the network, which absorb the effect of the bad nodes. The throughput is lower at a network size around $9 \times 9$ and $10 \times 10$. That is because the path length increases the chance to reach a bad node.

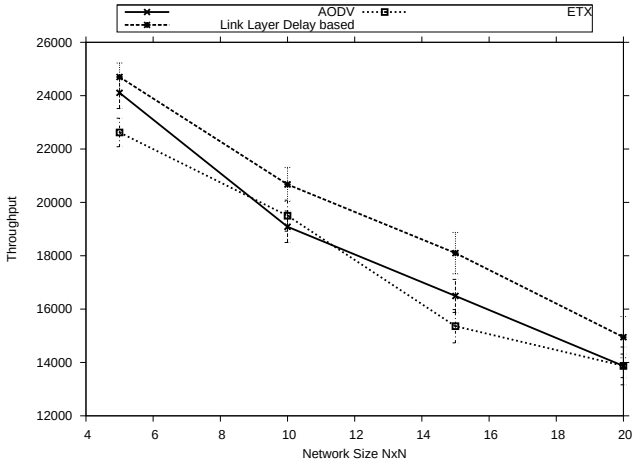



Figure 5.6: The overall throughput by increasing the number of low quality nodes

In the second simulation of a grid scenario, the network size is fixed to $5 \times 5$, and the number of low quality nodes is increased. The performance of the three algorithms is very close to each other, but it can be seen that the routing metric considering the link layer delay can achieve a higher performance than the other two. As explained above, ETX fails to avoid the low quality node along the path.

## 5.2 TCP-ETX: A Cross Layer Metric for TCP Improvement

Transmission Control Protocol (TCP) suffers degradation in wireless networks because, among other factors, low quality links tend to interrupt a TCP connection. Expected Transmission Count (ETX) calculates the estimated number of transmissions to successfully transmit a packet, but it fails to calculate the correct number of transmissions for a TCP segment. The reason is that it does not consider the Automatic Repeat reQuest (ARQ) mechanism of TCP. A novel cross layer path metric, TCP-ETX, is proposed which calculates the accurate number of transmissions for TCP segments and minimizes the protocol overhead.

### 5.2.1 TCP-ETX

As discussed above, ETX does not consider the upper layers of a network and their influence in the obtained value of ETX. To overcome that limitation, this work proposes a new TCP-ETX routing protocol to improve the TCP performance over wireless networks. To calculate the overall number of transmissions to successfully transmit a TCP segment, the behavior of TCP needs to be analyzed. In a correct transmission, there is at least one successful transmission for each packet of the segment. The number of transmissions is estimated by the link layer performs before stopping transmitting a frame. For instance, usually the maximum retransmission limit of the IEEE 802.11 protocol is 7, i.e., a station tries to transmit a frame up to seven times before giving up. Basically, the estimation can start either at the source node going into the direction of the destination node (forward approach) or the other way around (backward approach). In the forward estimation, a source node has to know the exact path to the destination and has fully knowledge of the network information, which is unrealistic in many wireless networks. Therefore the routing estimation is performed from the destination.

The principle behind TCP-ETX is based on the observation that given the current node, if any node on the path to destination fails to transmit a link layer message, the current node needs to retransmit the message anyway. Based on ETX, the probability to successfully transmit a packet in one hop is equal to the multiplication of the transmission probability for both directions, as presented in Equation 5.3. In Equation 5.4, Pro($i$) is the probability of a node to transmit a message caused by a failure in the transmission of any hop on the path. Each node has to send at least one message successfully in order

to reach its destination. If any node on the path that is closer to the current node fails its transmission, the current node transmits the packet again. Equation 5.5 explains the TCP-ETX metric, where $MAX$ denotes the maximum retransmission limit, $N$ denotes the current node and $D$ denotes the destination node. Fraction $\frac{1}{P_i}$ is the estimation of the number of transmissions to transmit a packet to next hop. By multiplying $\frac{1}{P_i}$ and the probability to transmit a packet $\mathrm{Pro}(i)$, the estimated number of transmissions that the current node needs for a successful delivery of a TCP segment can be retrieved. TCP-ETX sums the number of transmissions of all nodes along the path to get the total number of transmissions.

$$P_i = P_{i \to i+1} P_{i+1 \to i}. \tag{5.3}$$

$$\mathrm{Pro}(i) = 1 + \sum_{j=i}^{D} (1 - P_j)^{MAX}. \tag{5.4}$$

$$\mathrm{TCPETX}_{N,D} = \sum_{i=N}^{D} \frac{1}{P_i} \mathrm{Pro}(i). \tag{5.5}$$

### 5.2.2 One Hop Notification System

#### TCP-ETX Calculation

The main drawback of ETX is the overhead of its probing system. To know the delivery ratio of all destinations, each node has to send the probing information for all destinations, which is resource consuming mainly in wireless networks and may cause a heavy traffic in the network. Thus a more efficient notification system should be introduced into an ETX routing metric. By analyzing the TCP-ETX metric above, a one hop notification system is proposed, which estimates the corresponding routing metric based on one hop information. This means that each node only needs routing information from neighbours one hop distant, as well as probe the delivery ratio from one hop neighbours.

$$\begin{aligned} \mathrm{TCPETX}_{N,D} \;=\; & \frac{1}{P_N} \mathrm{Pro}(N) + \\ & \frac{1}{P_{N+1}} \mathrm{Pro}(N+1) + \ldots + \\ & \frac{1}{P_{D-1}} \mathrm{Pro}(D-1). \end{aligned} \tag{5.6}$$

$$\text{TCPETX}_{N,D} = \frac{1}{P_N}\text{Pro}(N) + \text{TCPETX}_{N+1,D}. \tag{5.7}$$

Equations 5.6 and 5.7 incorporate the recursive idea to calculate the corresponding TCP-ETX metric. However, Pro($N$) still contains the delivery ratio information from all nodes along the path. There are two ways to get information from the path: (i) all nodes from the path send their delivery ratio information to the source node, which causes a lot of traffic; and (ii) all nodes put their delivery ratio into the routing packets (piggybacking process), which enlarges the size of routing packets. Both ways consume network resources. In the following, those equations are optimized.

Equation 5.4 contains the delivery ratio for all nodes along the path. If a recursive equation for Pro($i$) can be found, the routing information can be eliminated from other nodes. Equation 5.8 recursively calculates the value of Pro.

$$\text{Pro}(i) = 1 + (1 - P_i)^{MAX} + (\text{Pro}(i+1) - 1). \tag{5.8}$$

Based on those equations, each node can estimate the corresponding TCP-ETX for all destinations, only based on one hop information. Instead of probing the whole network, each node merely probes their neighbours for routing information.

### TCP-ETX algorithm

Based on the TCP-ETX metric, our algorithm can maintain the accurate routing table with a low cost for TCP transmission. In this algorithm three agents are created to handle different aspects in our system: Probing Agent, Routing Agent and TCP-ETX Agent. These agents handle different aspects of our algorithm:

1. Probing Agent: obtains the delivery ratio from the neighbour nodes. The delivery ratio is maintained within a period of time, which is based on the quality of network. As mentioned above, it only probes one hop neighbours. Broadcast probing packets are sent at a probing period. The Time To Live (TTL) of the probing packet is set to 1, since it is interested in only one hop neighbours that can hear the probing.

2. Routing Agent: handles the routing table entries, which are updated for one hop neighbours. This agent sends and receives routing packets from one hop neighbours. It works similar to DSDV. However, when updating the entries of the routing table, it does not compare the number of hops.

3. TCP-ETX Agent: handles the TCP-ETX metric based on the information from the Probing Agent and Routing Agent. This agent measures and calculates the corresponding TCP metric for a specific destination.

Algorithm 7 and Algorithm 8 defines the behavior of the probing agent and the routing agent.

**Algorithm 7** Probing Agent

```
1: Define probeTable                          ▷ A table to count the Probing
2: Define probablityTable          ▷ A table to record the delivery ratio to neighbour
3: Define totalProbe               ▷ Total number of probing messages sent each time
4:
5: procedure CountProb(probePacket)                    ▷ calculates the Probing
6:     probe ← probeTable[packet.dst]
7:     probe++
8: end procedure
9:
10: procedure AddProb(dst)                             ▷ calculate the probability
11:     probability ← probeTable[dst]/totalProbe
12:     probabilityTable[dst] ← probability
13: end procedure
14:
15: procedure SendProbe                  ▷ send out probing to one hop neighbour
16:     probePacket.TTL ← 1
17:     probePacket.dst ← BROADCAST
18: end procedure
19:
```

### 5.2.3 Simulation

This section presents the simulation of the TCP-ETX metric in the ns-2 simulator [75] by rewriting the DSDV routing protocol. As source data traffic, the VOIP G7.11 CODEC is used with a segment size of 160 bytes and a 20 ms sending interval. The protocol used in this simulation is TCP, IEEE 802.11, TCP-ETX routing protocol, ETX protocol and DSDV protocol. The simulations are run for two different network topologies: grid and two paths. The main criterion used in the simulation is the total TCP throughput. Low

**Algorithm 8** Routing Agent

```
1: Define probeTable                                   ▷ A table to count the Probing
2: Define probablityTable          ▷ A table to record the delivery ratio to neighbour
3: Define p                                                          ▷ routing packet
4: Define TCPETX                          ▷ TCPETX metric table of all destinations
5: Define oldTCPETX                  ▷ old TCPETX metric table of previous node
6: Define Pro ▷ table about probability of the need to transmit a packet to a specific
   destination
7: Define seqnum                                  ▷ current stored sequence number
8:
9: procedure UpdateRoute          ▷ send out routing packet to one hop neighbour
10:     p.TTL ← 1
11:     p.dst ← BROADCAST
12:     p.TCPETX ← TCPETX[dst]
13:     p.Pro ← Pro[dst]
14: end procedure
15:
16: procedure HandlePacket(p)                          ▷ handle the routing packet
17:     if p.seqnum > seqnum or
          p.TCPETX < oldTCPETX[packet.dst] then
18:       Pro[p.dst] ← calculatePro(p.Pro)
19:       TCPETX[p.dst] ← calculateTCPETX
20:                       (p.TCPETX,Pro[d.dst])
21:     end if
22: end procedure
```

quality nodes are created randomly in the test network to have different quality routes. By comparing the total TCP throughput of the algorithms, it is concluded that our algorithm significantly improves the TCP throughput by choosing a high quality path.

First our algorithm is tested with the grid topology, which contains some low quality nodes to generate different quality paths. The simulation is run under different sizes of the grid topology with multiple available paths to the destination. Our purpose is to evaluate the performance of our protocol by comparing it with other routing metrics. Figure 5.7 shows that TCPETX presents extra TCP throughput when compared to ETX and DSDV. DSDV has lower throughput, because it only considers the number of hops of the path without considering the quality of the path. DSDV is able to detect the link loss to maintain the traffic, but it is too slow to react to changes because it lacks link quality measurement and estimation. While the grid size and number of hops are small, ETX maintains a similar throughput as TCPETX, because the position of the low quality link on a path is not significant. Accompanying the increment of path length, the throughput of ETX is closed to DSDV, because the low quality link on the tail of the path leads to a large number of retransmissions. Comparing to DSDV, ETX has a chance to find out the suitable link to get better performance than DSDV, while the low quality nodes exist at the beginning part of the path. On the other hand, if the low quality nodes locate at the end part of the path, ETX will have a lower performance than DSDV, because it believe it is the optimized path even though it leads to the packet loss. TCP-ETX is a cross layer routing metric that considers both lower layer link quality and upper layer TCP congestion window mechanism. In this simulation, TCP-ETX chooses the path that is suitable for TCP transmission over the multiple available paths, which can avoid the hot spot and increases the probability of the transmission.

Figure 5.8 presents the average TCP throughput on a $6 \times 6$ grid. It also explains the behavior of TCPETX over the time. TCPETX maintains a higher throughput than that of ETX and DSDV most of the time, because it correctly predicts the path quality for TCP. ETX gets similar throughput as DSDV, because it fails to evaluate the path for TCP by ignoring the position of the low quality link.

The second topology is the two-path topology. There are two available paths in the network from source to destination with the same number of hops but with different qualities. In this simulation, the purpose is to discover the efficiency of our protocol to perform route decision. There are only two routes in the network, and the quality of the route may switch because of the transmission traffic. Therefore, a fast and accurate decision is needed to optimize the transmission rate. Figure 5.9 shows that our protocol

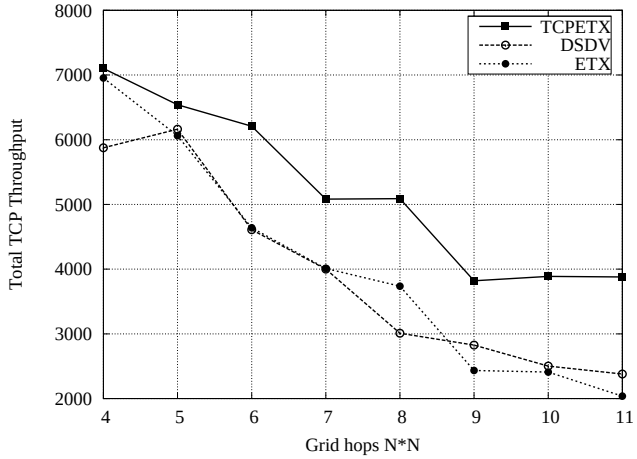



Figure 5.7: Comparing total throughput of TCP in a grid $N \times N$

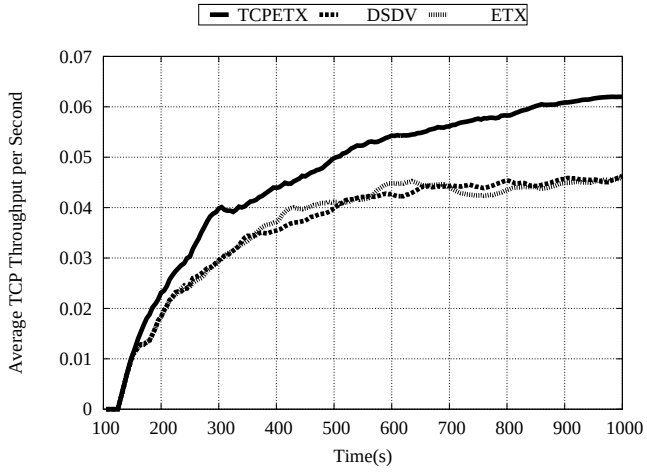



Figure 5.8: Comparing average TCP throughput on a $6 \times 6$ grid

provides a higher TCP throughput than the other protocols. By randomly choosing the path from the two paths with the same number of hops, DSDV only maintains a basic transmission traffic. The oscillation of TCP throughput with ETX is mainly caused by the mistake on estimating the number of expected transmissions for TCP. Therefore, the path that ETX chooses may not be suitable for TCP transmissions and causes the degradation of TCP throughput. In the simulation, TCP-ETX shows much higher throughput than the others, when the number of hops increase. TCP-ETX provides a high correctness on path estimation with long transmission path.

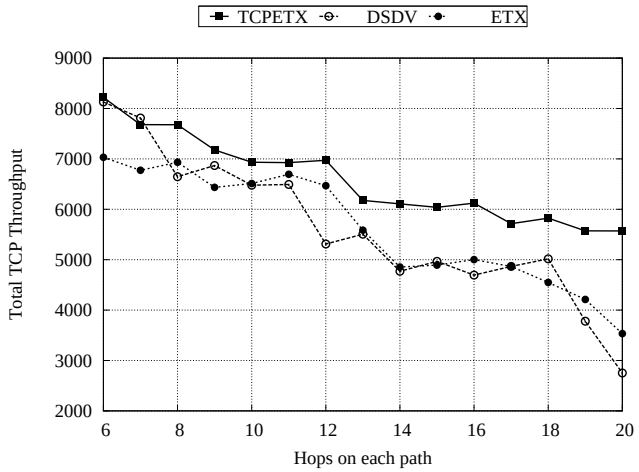



Figure 5.9: Comparing total TCP throughput in two paths topology with $N$ hops

Based on those results, the ETX protocol gets almost the same TCP throughput as DSDV, which means that ETX fails to improve the TCP performance. Our protocol fixes the problem that ETX encounters, and shows significant improvement on TCP throughput especially on large scale TCP transmission.

# Chapter 6

# MAC Layer Improvement

Besides the improvement on the Network layer, MAC layer can achieve improvement on video streaming as well. MAC layer is responsible for the link connection among network nodes. If the link quality can be improved, the quality of the whole path of a transmission can be improved as well. In this chapter, a cross layer QoS insurance technique is proposed to ensure the UDP transmissions. A TCP-CC is proposed for TCP throughput improvement, which is a cross layer TCP pacing protocol by contention control.

## 6.1 Cross Layer QoS Insurance by Contention Window Adjustment

It is crucial to achieve Quality of Service (QoS) on IEEE802.11 in order to provide stable and reliable communication for real time and multimedia applications. Most recent QoS techniques for ad hoc networks rely on basic QoS classifications such as Enhanced Distributed Channel Access (EDCA) and Hybrid Coordination Function Channel Access (HCCA) with pre-fixed contention window range for specific services; furthermore, these techniques do so without considering overall resource consumption and the external correlation among transmitting source nodes. In this research, a transmission analysis on link layer delay and transmission interval was conducted in three different scenarios: the unsaturated scenario, the saturated scenario and the mixed scenario. Based on this analysis, a novel contention window adjustment QoS scheme is proposed; this applies dynamic and automatic QoS assignments with low complexity by considering the restrictions among various requirements simultaneously. We measure the performance of

different QoS algorithms theoretically and verify improvements on QoS-fairness resulting from our algorithm on QoS-fairness with detailed simulations. Other simulations are also conducted in order to verify our theory on the relation of link layer delay and transmission interval.

### 6.1.1 Introduction

Contention window mechanisms are designed to eliminate the possibility of collisions on networks. If transmitting nodes attempt to access one resource simultaneously, they will cause a collision. When a collision occurs, the contention window mechanism forces the transmitting nodes, which have started the transmission, to wait for a random period of time for retransmission. Thus, the retransmissions have a smaller opportunity of colliding. A suitable contention window range can simultaneously absorb the increased system load and guarantee the maximum throughput of the transmitting nodes. However, if the contention window range is too large, it may cause an unnecessary delay in the transmission, the reason for this is that for each collision a longer backoff timer may be set.

There are two purposes for contention window adjustment: network performance optimization and Quality of Service (QoS) insurance. Several different research approaches on contention window adjustment are proposed for network performance optimization, example of which are [53, 110]. The aim of both is to optimize the overall network throughput by separating the different traffic flow in different time-slots for collision avoidance. The deferring and backoff mechanism of the contention window is one of the most popular approaches to achieve collision avoidance.

The research on contention window adjustment aims to conduct QoS insurance on wireless networks; the following are examples of such: [72, 124, 7, 118, 31, 111]. This research classifies the throughput requests in different categories in order to simplify the classification of the contention window size. A contention window size is assigned to each category in order to guarantee the requested throughput of each transmitting node. However, the contention window size of each category is pre-fixed and never changes during the runtime; on the other hand, each category may have various composition during the runtime. It is difficult for the pre-fixed contention window size to guarantee QoS in the case of dynamically changed requests. Even when throughput requests are in the same category, there are still slight differences between these two requests. One pre-fixed contention window size will leads to QoS unfairness, because one of the requests

may receive additional network resources.

In this research, an analysis of the transmission throughput is presented, which is classified in three scenarios: 1) all nodes are unsaturated; 2) all nodes are saturated and 3) there is a mix of unsaturated nodes and saturated nodes. The reason why these three scenarios are considered is that contention window size has different effect in these scenarios. An approach, to determine if a node is saturated or not, is also discussed in this research, the purpose of this is to explain the reason for the different effect that contention window size leads to. The link layer delay is the key factor affecting the transmission throughput. The contention window size is one of the method used to control the link layer delay. Therefore, eventually, contention window size can affect the transmission throughput. In the scenario where all nodes are saturated, a Markov Chain model is presented to analyze the relation between contention window size and transmission throughput. A proportional relation between contention window and transmission throughput is derived. This is proven to achieve higher QoS satisfaction and QoS fairness. The QoS satisfaction measures the percentage of the satisfied resource requests. The QoS fairness measures the percentage in the difference between the assigned resource and the requested resource.

### 6.1.2 Basic Theory

The purpose of our research is to achieve quality of service (QoS) assurance over wireless networks on IEEE802.11. Each node can request a specific throughput for its transmission; and, the system assigns and limits the traffic flow from this node according to the requested throughput. There are several ways to achieve traffic control over wireless networks. We would like to use the contention window adjustment technique to achieve the quality of service assurance; this is because the contention window mechanism is built in IEEE802.11, which does not require modification on other layers. To simplify our research, we assume that all sources transmit at a fixed transmission rate. In other words, all sources flows have a fixed interval between two continuously transmitting packets.

In this section, we are going to present the basic theory about the wireless networks related to our research. Through this explanation, we can further understand the problem that we are facing. The infrastructure of our research is IEEE802.11, which contains a RTS/CTS (Request to Send and Clear to Send) mechanism and a contention window mechanism. RTS/CTS mechanism is designed to reduce the packet transmission collision over wireless networks, which is caused by a hidden node problem. The contention

window mechanism is designed to control and reduce the contention level over wireless networks. These are the two main features of IEEE802.11, which relate to our research.

To analyze network condition, there are two concepts, which we want to clarify, before beginning the discussion: the first on pertains to saturated nodes and the second to unsaturated nodes. A saturated node is a node that always has something to transmit. By contrast, an unsaturated node always experience some delay between two continuous transmissions. In this section, we are going to explain the cause and to identify both saturated and unsaturated nodes.

### RTS/CTS

Before each data packet transmission, IEEE802.11 requires the MAC(Media Access Control) Layer to send a RTS message to the next hop node; this is done in order to confirm whether the channel is idle and whether the next hop node is ready to receive the packet. If both the next hop node and the channel are ready for transmission, the next hop node will send back a CTS message to inform the transmitting node. After successfully receiving the CTS, the transmitting node will start the transmission. The purpose of the RTS/CTS mechanism is to determine the network condition before starting the actual data transmission.

### Contention Window Mechanism

The contention window mechanism is designed to achieve contention control in IEEE802.11. Contention window mechanism is used to determined the time needed for deferring and backoff. If the media is unavailable to transmit packets, the transmitting node defers the access until the medium is available. If a collision happens among the packet transmissions of two nodes, each transmitting node will set a backoff timer and retry the transmission when the timer becomes zero. Both the deferring time and the backoff time are determined by the contention window mechanism. If a collision happens when two nodes intend to transmit a packet simultaneously, the contention window size will increase in order to increase the range of various choices of deferring and backoff times. In summary, the contention window mechanism tries to separate the time of transmission launched by each node, in order to avoid transmission collision.

Since the contention window mechanism introduces time intervals among packet transmissions, it affects the transmission rate in two main aspects. In one case, the additional time interval delays the packet transmission or the packet retransmission,

when a packet transmission has started or a collision has happened. Therefore, the transmission rate is lowered by the contention window mechanism. Another case is when the contention window mechanism lowers the collision probability of the networks, which decrease the time of the backoff. In this case, it actually decreases the total cost of waiting for the retransmission caused by collision. This tradeoff of the contention window mechanism will be discussed and considered in our research. A detailed design will be proposed in the following section.

**Saturated Networks**

There are two kinds of transmission rates are considered in our research: source transmission rate and actual transmission rate. Source transmission rate is the transmission rate of the source applications or devices. Actual transmission rate is the transmission rate that a traffic flow actually transmits through the channel. These two transmission rates are different; this is because the link layer delay may interrupt the transmission.

Theoretically, saturation means the system has reached its maximum communication capacity. In other words, the saturated node always has something to send. In practical terms, saturation is related to both the link layer delay and the source transmission rate. If the link layer delay is too high, the network will easily become saturated. If the transmission rate is too fast, the traffic flow may be saturated as well. In Figure 6.1, $F1$ is the original traffic flow in the source transmission rate, in which two simultaneously transmitted packets are separated by the fixed interval. If the average link layer delay is larger than the interval between two simultaneously transmitted packets, the transmission of the second packet will be delayed. The actual traffic flow will become like that illustrated in $F2$ in Figure 6.1. The transmission delay will be forwarded to all the following packets, which are planned to transmit by the source. The link layer delay will be applied to the second packet, when the first packet is successfully delivered. All this delay will be accumulated on the following transmission. Therefore, there are always packets in the queue, which wait for delivery. And thus, the reason for saturation can be considered as being caused by two aspects: link layer delay and source transmission rate. The definition of saturation is as found in Definition 1.

**Definition 1** *If the average link layer delay is higher than the fixed transmission interval of a source transmission flow, the source transmission flow is saturated.*

Based on the above analysis, we can infer that the actual transmission rate of a saturated node is different from the source transmission rate. The actual transmission

rate is determined by the link layer delay. If the link layer delay can be measured or estimated, eventually the actual transmission rate of a traffic flow can be calculated. It is also true that if the link layer delay can be changed, the actual transmission rate can be changed. This concept is very important in our research, which will be discussed in the following section.

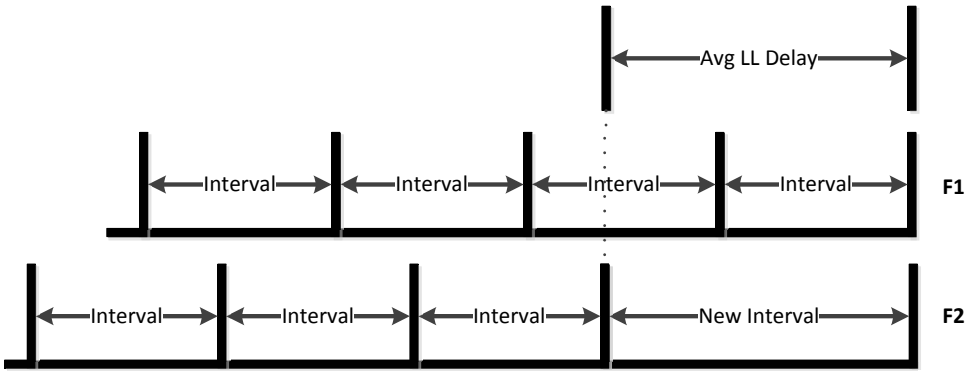



Figure 6.1: Example for Saturated Network Transmissions

### Unsaturated Networks

Based on the explanation of saturation from the section above, we can understand the definition of unsaturation in the same way. In Figure 6.2, there is a source which transmits packets at a fixed interval. The transmission interval is larger than the average link layer delay, which means the time to transmit a packet is smaller than the fixed interval between two simultaneously transmitted packets. Therefore, there is a gap after the first packet finish the transmission and before the transmission of the second packet start. During this gap, the channel is idle and the queue is empty, because no packet intends to transmit in the channel. There are not always packets in the queue, which are waiting to be transmitted. Therefore, the node is unsaturated as described in Definition 2. The cause for an unsaturated network can be classified as: a large network capacity or a slow transmission rate. The solution of the unsaturated networks is much simpler; this is because the link layer delay does not have any impact on the traffic flow. All traffic flow in unsaturated networks actually transmit from its own source transmission rate.

**Definition 2** *If the average link layer delay is lower than the fixed transmission interval of a source transmission flow, the source transmission flow is unsaturated.*

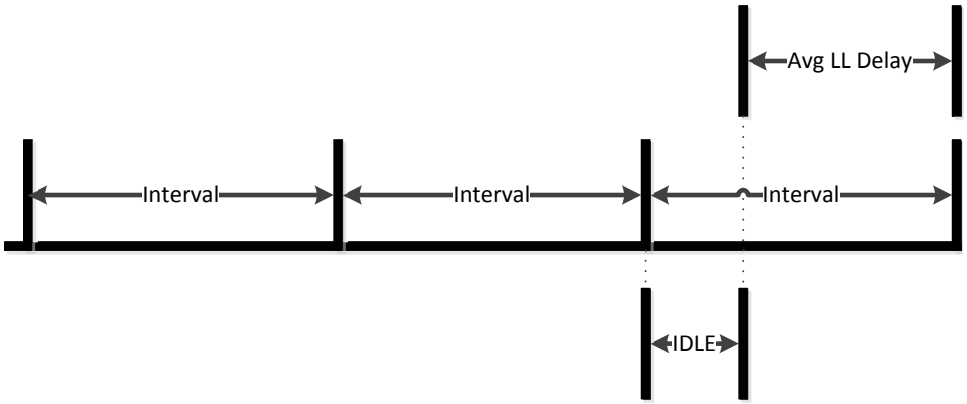



Figure 6.2: Example of Unsaturated Network Transmissions

**Summary**

In order to achieve quality of service assurance in IEEE802.11, the scenarios have to be classified in three categories: 1) all nodes are unsaturated, 2) all nodes are saturated and 3) there is a mix of saturated and unsaturated nodes. The reason for this classification is that the behavior of saturated nodes and unsaturated nodes is different. The actual transmission rate of saturated nodes is more difficult to estimate and control, when compared to unsaturated nodes. They all need to be considered separately. The condition of these three situations is discussed and the solutions are proposed in this section.

The purpose of the solution is to satisfy the requested transmission throughput of all transmitting nodes. Based on the discussion above, the contention window adjustment algorithm eventually affects the transmission throughput, this is because the contention window adjustment algorithm changes the link layer delay. Our solution is based on this theory. In this section, all scenarios only consider one hop transmission. The scenario contains one destination node and multiple source nodes.

### 6.1.3 All Unsaturated Nodes

This is the simplest case, because the average link layer delay is lower than the transmission interval of all transmitting nodes. Therefore, the average link layer delay has no impact on the transmission rate. The actual transmission rate is the same as the source transmission rate. All requests from the transmitting nodes are certainly satisfied, because each node is transmitting at its source rate. The overall network capacity will not be exceeded in this situation; this is because there is a gap in the form of an idle

channel after each successful packet transmission, which means there is room to increase the transmission rate.

However, there is still one thing that can be considered in this situation. Even though the transmission rate is not affected, the packet loss probability can be reduced. The purpose of reducing the packet loss probability is to create extra available network resources for the upcoming transmission. Meanwhile, this can save the energy of a communication node consumed by packet retransmission. If the transmission of a RTS fails in IEEE802.11, the MAC layer will retransmit it until a CTS is successfully received or until the maximum retransmission times is reached. To reduce the amount of retransmission, it reduces the total number of transmissions in the networks; this eventually creates extra idle channels for upcoming transmissions.

The contention window mechanism is one of the main approaches to reduce the packet loss probability. In our collision probability based adaptive contention windows adjustment, we proposed a collision probability model to estimate the collision probability in a wireless networks, which is shown in Equation (6.1). $F_c$ denotes the probability of a collision, $E[BF]$ denotes the average backoff and $N_{node}$ denotes the number of nodes attempting to transmit data to the same destination node simultaneously. This model shows that if the average backoff gets higher, the collision probability will get lower. Therefore, a large contention window size can be chosen in order to reduce the packet loss probability. However, if the contention window size is too large, it may cause high link layer delay, which causes the node to become saturated. A saturated node may eventually reduce the throughput of the network. A suitable size of contention window can be estimated by a link layer delay model, which will be proposed in the following section.

$$F_c = 1 - (1 - \frac{1}{E[BF]})^{N_{node}-1} \tag{6.1}$$

### 6.1.4 All Saturated Nodes

In the first step of the saturation analysis, we would like to simplify the analysis by merely considering that all nodes in the network are saturated. In this case, the transmission rate of all nodes is determined by the link layer delay; this is because the link layer delay is larger than the time interval between two simultaneously transmitted packets. Since all nodes are in the same situation, we can consider the network condition as stable; this is because there is always the same number of packets attempting to transmit over

the network simultaneously. Therefore, we also can assume that the average packet loss probability of the network is always the same, since the network condition is stable.

### Throughput Model

The pre-fixed contention window range of a set of source nodes cannot satisfy the requirements of real time applications in ad hoc networks, because the pre-fixed parameters cannot adjust the throughput according to the various QoS requirements. Therefore, we are going to propose a dynamic contention window adjustment mechanism through a self-determined algorithm. In our analysis, we assume that the network is in a saturated condition, which means all source nodes always have a packet to send and all queues are not empty. Furthermore, we assume that the total requested throughput is lower than the network capacity, and that the nodes transmit in an aggressive way, which makes them saturated. Therefore, the source transmission rate of each node is higher than the transmission rate that they requested. There are two reasons to make this assumption. One reason is that if the total requested throughput is higher than the overall network capacity, it is impossible to satisfy all the requests since the network is overloaded. To simplify the analysis, we would like to model the normal loaded networks at the beginning. The overloaded cases will be discussed in the following section. The reason for another assumption is as follows: if all sources are transmitting at the rate they request, the only situation leading to saturation in all sources is that the network is overloaded. This is because no idle channel is available for the packets waiting in all queues.

Our analysis attempts to model the transmission failure on a saturated channel. Though the causes of transmission failure vary, they trigger the contention window mechanism every time in IEEE802.11. In order to study the behavior of the saturated channel, we model the system as a discrete time Markov Chain. Figure 6.3 demonstrates the Gilbert Burst model[43], while a source node is sending a packet on the channel; $a$ and $b$ are the probability corresponding to the transmission of the source node, which is greater than 0 and smaller than 1. We have two states in our transition model: the *Start* state and the *Transmission* state. If a transmission fails, the model will go to or stay in *Start* state. The transition will go to *Transmission* state, only when a transmission is successful. At each state, there is a probability for it to transit to another state. For example, the transition probability from the *Start* state to *Transmission* state is the probability of a successful packet transmission on a saturated channel. In the following sections, we will discuss how to calculate the transmission failure probability with our model, and how to estimate the average throughput of one source node in a saturated

channel.

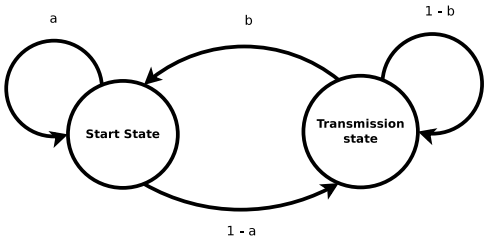



Figure 6.3: Gilbert Burst Model of Transmission on a Saturated Channel

### Transmission Failure Probability

We make the assumption that the network model is in a saturated condition. We can therefore assume that the network condition is stable at any point during the runtime. The probability of sending a packet successfully is the same during the runtime. The probability of each transmission is shown in the transition matrix $\mathbf{P}$ in Equation (6.2). Transition matrix $\mathbf{P}^n$ represents the transmission matrix after n transmission times.

$$P = \begin{bmatrix} a & 1-a \\ b & 1-b \end{bmatrix} \tag{6.2}$$

In a saturated channel, the probability of losing a packet during each transmission is greater than 0 because of the busy traffic. According to the theory of Markov Chain, every transition matrix $\mathbf{P}^n$ of the regular Markov Chain can converge to a matrix $\mathbf{W}$, which has constant columns. Therefore, the Markov Chain that we create is a regular Markov Chain because all elements in the matrix are positive.

$$W = \begin{bmatrix} w_1 & w_2 \\ w_1 & w_2 \end{bmatrix} \tag{6.3}$$

Based on the theory of Markov Chain, we can calculate the $w_1$ and $w_2$ from the equations derived from the Equation (6.4).

$$\begin{cases} w_1 + w_2 = 1 \\ w_1 * a + w_2 * b = w_1 \\ w_1 * (1-a) + w_2 * (1-b) = w_2 \end{cases} \tag{6.4}$$

We then determine the convergence matrix $\mathbf{W}$ of $\mathbf{P}^n$ as shown in Equation (6.5), where all rows in matrix $\mathbf{W}$ have the same vector. When $n \to \infty$, $\mathbf{P}^n$ is equal to $\mathbf{W}$. We can predict the probability of the transmission based on matrix $\mathbf{W}$ after sufficient steps are taken. In the next section, we will discuss the prediction for the average throughput of each source node according to the matrix $\mathbf{W}$.

$$W = \begin{bmatrix} \frac{b}{1-a+b} & \frac{1-a}{1-a+b} \\ \frac{b}{1-a+b} & \frac{1-a}{1-a+b} \end{bmatrix} \tag{6.5}$$

**Average Throughput Analysis**

The contention window mechanism aims to achieve contention avoidance, in order to improve the average bandwidth $E[B]$ of each source node. Therefore, in our analysis, we seek to know how the contention backoff parameter affects the average bandwidth. Firstly, we create an equation for predicting the average bandwidth in the Markov Chain, as generated in the previous section. $E[B]$ can be formalized as Equation (6.6).

$$E[B] = \frac{TP_{total}}{T_{total}} \tag{6.6}$$

$T_{total}$ denotes the total measured time, and $TP_{total}$ denotes the total throughput within $T_{total}$. We assume that a source node transmits $\mathbf{N}$ packets during $T_{total}$, while $\mathbf{N}$ is large enough. In this case, transmission matrix $\mathbf{P}^N$ is closed to matrix $\mathbf{W}$. The Markov chain begins at *Start* state, so we only focus on the probability that begins at the *Start* state. The transition matrix can multiply to a vector $\begin{bmatrix} 1 & 0 \end{bmatrix}$ in order to eliminate the probability corresponding to the transmission state at the first state. The amount of $TP_{total}$ and $T_{total}$ can be calculated as Equation (6.7).

$$\begin{cases} TP_{total} = \sum_{n=1}^{N} \begin{bmatrix} 1 & 0 \end{bmatrix} * P^n * \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ T_{total} = \\ \sum_{n=1}^{N} \begin{bmatrix} 1 & 0 \end{bmatrix} * P^n * \begin{bmatrix} E[BF] \\ E[T_{tx}] \end{bmatrix} \end{cases} \tag{6.7}$$

$E[BF]$ denotes the average Backoff parameter and $E[T_{tx}]$ denotes the average time of packet transmission. The total throughput $T_{total}$ of a source node is the number of successful transmissions during the runtime. Therefore, it multiplies a vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ to choose the number of successful transmissions from the matrix. The total time slots that

the source node uses should be separated into two situations: 1) When the transmission succeeds, the cost time slot is the time spent transmitting a packet; 2) When the transmission fails, the cost timeslot should be based on the backoff time chosen by the contention window mechanism. Here we will use the average transmission time and the average backoff parameter to replace the exact execution time.

$$E[B] = \frac{\begin{bmatrix} 1 & 0 \end{bmatrix} * ((N+1)W - I) * \begin{bmatrix} 0 \\ 1 \end{bmatrix}}{\begin{bmatrix} 1 & 0 \end{bmatrix} * ((N+1)W - I) * \begin{bmatrix} E[BF] \\ E[T_{tx}] \end{bmatrix}} \tag{6.8}$$

Based on the theory of the regular Markov Chain, $\mathbf{P}^n$ is predictable, because it converges with $\mathbf{W}$. By replacing $\mathbf{P}^n$, we derive Equation (6.8) from the Equation (6.7), when the number $\mathbf{N}$ is large enough.

$$\lim_{n\to\infty} E[B] \approx \frac{1}{\frac{b}{1-a} * E[BF] + E[T_{tx}]} \tag{6.9}$$

After simplifying Equation (6.8) by making n sufficiently close to infinite, we obtain Equation (6.9). In order to avoid further contention after collision occurs, the contention window must not be too small. Thus, the average transmission time is usually very small compared to the average backoff time. In a saturated channel, the probability $a$ and probability $b$ are very similar; this is because of their stable environment. Based on the research in the study [109], the collision probability of a reasonable network size with 30 source nodes is within the range of [45%, 55%], while the size of the initial contention window range from 16 to 32. In a reasonable network size with about 30 source nodes transmitting data simultaneously, $a$ and $b$ are similar and within the range of [0.45, 0.55]. The factor $\frac{b}{1-a}$ is greater than 80% in a network with a size of 30 source nodes. Therefore, $\frac{b}{1-a} * E[BF]$ is still much greater than $E[T_{tx}]$. We can predict that the average throughput $E[B]$ is approximately equal to Equation (6.10) in a reasonable network size with 30 source nodes.

$$E[B] \approx \frac{1}{\frac{b}{1-a} * E[BF]} \tag{6.10}$$

We acquired the estimated average throughput of a source node from our analysis of transmission probability in a saturated channel. The result shows that the average throughput of a source node principally depends on its average contention backoff parameter as well as on the channel condition. In other words, it is affected by the size

of the contention window, because the contention window range determines the average backoff parameter. Therefore, when the contention window range gets smaller, the average throughput should get larger until the system resource is exhausted.

$$E[BF] \approx \frac{1}{\frac{b}{1-a} * ReqTP} \tag{6.11}$$

**Basic Theory**

Our goal is to achieve QoS fairness over ad hoc networks, which means each source node will get the corresponding resources that it requests. If we set the average throughput to correspond with the throughput requested by the QoS requested throughput, we can easily predict the average backoff parameter based on the corresponding QoS requested throughput as seen Equation (6.11). In Equation (6.11), *ReqTP* denotes the QoS requested throughput. The suitable average backoff parameter is dependant on the channel condition, $a$ and $b$, and the requested throughput of the application. From the previous equations, we can see that the average throughput is basically decided by the backoff parameter, but it is also affected by the channel condition. We cannot determine the average throughput of one source node without considering the channel condition. Therefore, we need to analyze the external relation among source nodes. Since we assume that all source nodes work in a saturated channel, we can assume that the channel condition of all source nodes is the same. According to this assumption, we acquired the ratio for the contention backoff parameter of all source nodes within the system in Equation (6.12).

$$E[BF]_1 : E[BF]_2 : ... : E[BF]_n \approx \frac{1}{ReqTP_1} : \frac{1}{ReqTP_2} : ... : \frac{1}{ReqTP_n} \tag{6.12}$$

This ratio reveals the external relation among the source nodes within the system; it also reveals how the QoS requested throughput affects the average contention backoff parameter. Based on the definition of a contention window mechanism, backoff time is the random number of time slots in the range from 0 to the size of the contention window. The average backoff parameter should be half of the size of the contention

window; therefore, we can obtain another ratio between the range of the contention window and the requested throughput in Equation (6.13).

$$CW_1 : CW_2 : ... : CW_n \approx \frac{1}{ReqTP_1} : \frac{1}{ReqTP_2} : ... : \frac{1}{ReqTP_n} \tag{6.13}$$

Equation (6.13) is the primary theory behind our algorithm. Like HCCA, there is a QoS coordinator in our algorithm, which collects channel information and sets source nodes to an appropriate contention window size corresponds to the ratio of requested throughput.

**Overload Control**

The definition of overload in our algorithm is when the requested throughput exceeds system capable throughput. there is one drawback to the main theory, proposed in the previous section; the actual throughput of each source node will not reach the requested throughput when the overall requested throughput is greater than the channel capacity. The reason for this is that channel capacity has been divided by the ratio of requested throughput, and not by the real number of the requested throughput. The reason for this is that all source nodes compete to access the network medium, which causes; high collision probability during the runtime. This high collision probability can not be absorbed by the contention window mechanism, because of the limited network capacity. In this section, we will discuss this problem; and, we will proceed to present an overload control for the system. In overloaded networks, only part of the requests can be satisfied.

While the requested resource is more than the actual resource, the coordinator has to decide whose QoS request should be guaranteed. Therefore, few source nodes, called guaranteed source nodes, will get the QoS requested throughput, while others, called Best Effort source nodes, will get the shared bandwidth in an overloaded condition. As shown in Equation (6.14), $ReqTP_{grt}$ denotes the requested throughput of guaranteed source nodes, and $TP_{BE}$ denotes the assigned throughput of Best Effort source nodes. $TP_{actual}$ denotes the channel capacity. The overall throughput of the guaranteed source nodes $ReqTP_{overall}$ has to be smaller than the channel capacity, so that it has extra throughput for Best Effort requests. The overall throughput of Best Effort source nodes should be equal to the rest of the channel capacity; therefore, they will not seize resources from the guaranteed source nodes.

$$\begin{cases} 1. & ReqTP_{overall} = \\ & ReqTP_{grt_1} + ReqTP_{grt_2} + ... + ReqTP_{grt_n} \\ 2. & TP_{BE_1} + TP_{BE_2} + ... + TP_{BE_n} = \\ & TP_{actual} - ReqTP_{reserved} \\ 3. & ReqTP_{overall} < TP_{actual} \end{cases} \tag{6.14}$$

In this phase, we have two methods by which to separate the rest of the actual throughput for the Best Effort source nodes. 1) the rest of the throughput will be distributed equally among all Best Effort source nodes; 2) or, the rest of the throughput will be distributed based on the ratio of requested throughput of the Best Effort source nodes.

$$\begin{aligned} CW_{grt_1} : ... : CW_{grt_n} : CW_{BE_1} : ... : CW_{BE_n} = \\ \frac{1}{ReqTP_{grt_1}} : ... : \frac{1}{ReqTP_{grt_n}} : \\ \frac{K}{ReqTP_{BE_1}} : ... : \frac{K}{ReqTP_{BE_n}} \end{aligned} \tag{6.15}$$

$$K = \frac{ReqTP_{BE_{overall}}}{TP_{actual} - ReqTP_{overall}} \tag{6.16}$$

Equation (6.15) and (6.16) illustrates how to conduct overload control with the ratio division on Best Effort source nodes. All Best Effort source nodes have to multiply an extra factor, $K$, which converts the normal ratio value to the Best Effort ratio value. Therefore, the throughput will be redivided based on the new ratio. To implement this mechanism, the coordinator, which retrieves data from the source nodes, should measure the channel capacity, in order to control the overall throughput.

### 6.1.5 Mix of Saturated Nodes and Unsaturated Nodes

The instances of all unsaturated nodes and all saturated nodes are easy to analysis, because the scenario is simple. In the case in which there is a mix of saturated nodes and unsaturated nodes, the status of the source nodes may be changed, if the network condition changes. The unsaturated and saturated nodes should be considered separately because of their different behavior. Since the analysis in this section only considers one hop scenarios and all source nodes are transmitting at a fixed interval, all source nodes

transmitting to the same destination node access the same channel; and, the channel condition that they attempt to access is similar. Based on this assumption, we can infer that the transmission rate of the unsaturated nodes is lower than the transmission rate of the saturated nodes in the mixed scenario. The reason for this is that if the channel condition is similar, we can assume that the link layer delay is similar as well. If a node is unsaturated, it means its transmission interval is higher than that of the link layer delay; this means its source transmits at a slow rate. In contrast, the source node transmits at a fast rate, if its transmission interval is lower than the link layer delay, which makes it in a state of saturation. We assume the saturated nodes are transmitting at a rate faster than is required. The reason for this assumption is that if a node is saturated and it transmits at its required transmission rate, it is impossible to satisfy this request; this is because the link layer delay is too high for the requirements.

**Handling Unsaturated Nodes**

Unsaturated nodes can be handled similarly to those in the unsaturated scenario, since the link layer delay is lower than the transmission interval. The transmission rate is guaranteed. Based on the theory in Section 6.1.3, a suitable contention window size should be assigned to avoid extra network collisions over the networks, which saves the network resource for upcoming connections. Based on Equation (6.1), higher contention window size leads to a lower packet collision probability; thus, a large contention window size should be assigned. In contrast, a large contention window size results in longer periods of deferral and backoff time for each transmission in the MAC layer; this may in turn increase the link layer delay and may cause the node to become saturated. Therefore a suitable contention window size should be large enough, but it will not make the link layer delay higher than the transmission interval. The best contention window size should make the link layer delay equal to the transmission interval.

In order to estimate the link layer delay, we propose a set of equations to model the link layer delay of IEEE802.11. To simplify this modelling, we assume that if a CTS is received, the data packet will be transmitted successfully. We also assume that the channel from source nodes to destination node is always busy, since there are saturated nodes. Therefore, the packet transmission will always be deferred. The channel from the destination node to source node is always idle, because destination node is not sending packet. Based on this analysis, we can infer that the transmission of CTS will not be deferred and it will not collide as well.

$$\begin{aligned} E[T_{LLD}] &= E[T_{def}] + E[T_{Tr}] \\ &\quad +(E[N_{BF}]+1) \times E[T_{def}] \\ &\quad +E[N_{BF}] \times E[T_{BF}] \end{aligned} \tag{6.17}$$

Average link layer delay, $E[T_{LLD}]$, can be calculated by Equation (6.17), in which $E[T_{def}]$ denotes the average time assigned for deferring; $E[T_{Tr}]$ denotes the average time for a data packet transmission; $E[N_{BF}]$ denotes the average number of times RTS transmission fails to successfully retrieve a CTS; and, $E[T_{BF}]$ denotes the average period of time of backoff. $E[T_{def}]$ and $E[T_{BF}]$ are the same, because they are both determined by the contention window mechanism. The first $E[T_{def}]$ and $E[T_{Tr}]$ listed are the average deferral and transmission time for the data packet. The transmission time for RTS and CTS is ignored in this equation; this is because they are very small compared to the transmission time of data packets and the backoff time. Therefore, we only consider the deferral time and backoff time of RTS in our modelling. There are $(E[N_{BF}]+1)$ RTS needed to successfully receive a CTS packet; thus, the RTS transmission defers for $(E[N_{BF}]+1)$ times and there are $E[N_{BF}]$ times backoff because of the RTS failure.

$$\begin{aligned} E[N_{BF}] &= (1-p)p + 2(1-p)^2p + 3(1-p)^3p \\ &\quad +... + (m-1)(1-p)^{m-1}p \\ &\quad +m(1-p)^m \end{aligned} \tag{6.18}$$

$$E[N_{BF}] = \sum_{i=1}^{m-1} i(1-p)^i p + m(1-p)^m \tag{6.19}$$

To measure the average number of RTS failure undergone in order to successfully receive a CTS packet, Equation (6.18) has been derived. In Equation (6.18), $p$ denotes the probability of successfully transmitting a packet, and $m$ denotes the maximum amount of time RTS transmissions occur for each data packet. The equation can be written as Equation (6.19). $(1-p)^i p$ represents the probability of the situation in which there are $i$ times of RTS failure; and, in the case of $i+1$ times of RTS transmission, the source node successfully receives a CTS packet. The sum of the $i$ multiplied by $(1-p)^i p$ is an accumulation of the average number of RTS failure. The last situation is on in which the total number of RTS transmissions reach the maximum times $m$ the data packet

transmission has failed and the RTS/CTS process stops. The next data packet will be selected and the transmission will be started again. The probability for this situation is $(1-p)^m$.

$$E[T_{Tr}] = \frac{E[L_{data}]}{r} \tag{6.20}$$

$$E[T_{def}] = E[T_{BF}] = \frac{CW}{2} \tag{6.21}$$

$E[T_{Tr}]$, $E[T_{def}]$ and $E[T_{BF}]$ can be calculated in a more simple way, as shown in Equation (6.20) and Equation (6.21). $E[L_{data}]$ denotes the average length of data packets, $r$ denotes the network transmission rate and $CW$ denotes the contention window size.

The probability of successful packet delivery, $p$, can be simply measured by the probing system. Based on the set of equations discussed above, we are able to estimate the link layer delay of a source node. For unsaturated nodes in the mixed scenario, the corresponding contention window size $CW$ can be calculated by simply assigning the transmission interval to the average link layer delay $E[T_{LLD}]$.

### Handling Saturated Nodes

Unlike unsaturated nodes, which will not be affected by saturated nodes, the throughput and transmission rate of the saturated nodes in the mixed scenario is affected by the transmission rate of the unsaturated nodes in the mixed scenario. From the analysis in section 6.1.4, we can see that the throughput of the saturated nodes is limited by the network capacity. The total network capacity is separated; and, this is based on a ratio relation of the contention window size. Therefore, in the scenario in which there is a mix of saturated and unsaturated nodes, the total network capacity for saturated nodes should be calculated.

As it is mentioned above, the transmission rate of unsaturated nodes is guaranteed, because the link layer delay is lower than the transmission interval. Since the transmission rate of saturated nodes is very fast, this makes the link layer delay higher than the transmission interval. Because the throughput of unsaturated nodes is guaranteed, the network capacity left for saturated nodes is equal to the overall network capacity deducted by the network capacity the unsaturated nodes employ, as shown in Equation (6.22). $Q_{saturated}$ denotes the network capacity left for saturated nodes, $Q_{overall}$ denotes the overall network capacity and $Q_{unsaturated}$ denotes the network capacity utilised by unsaturated nodes.

$$Q_{saturated} = Q_{overall} - Q_{unsaturated} \tag{6.22}$$

After the left over network capacity is calculated, the throughput we can offer for the guaranteed services can be measured. The throughput can be shared by the saturated nodes based on Equation (6.13). Since the contention window size is assigned a ratio, a large enough base number can be used to assign the contention window size in order to avoid additional link layer delay; this may cause the unsaturated nodes to become saturated. If the requested throughput is higher than the left over network capacity $Q_{saturated}$, the overloaded control mechanism in Equation (6.15) can be used to guarantee part of the requests which have a higher priority.

From Equation 6.17, it shows that the link layer delay is scaled by an average backoff time scince $E[T_{def}]$ is equal to $E[T_{BF}]$, when the $E[T_{Tr}]$ is much smaller than $E[T_{BF}]$. Usually, the time used for a 160 bytes data packet transmission in a 54 Mbps is measured in microseconds. As we mentioned in the paragraphs above, we would like to assign a large enough base number for the contention window ratio equation to avoid additional link layer delay. A contention window size as large as 100 will have an average backoff time measured in milliseconds. Therefore, usually, $E[T_{BF}]$ is much larger than $E[T_{Tr}]$. Since the throughput of saturated nodes are determined by the link layer delay, the throughput of saturated nodes can be determined by $E[T_{BF}]$ as well.

The base number for the proportional relation actually can be calculated by Equation (6.17), because the link layer delay is the actual transmission interval of the saturated network. The highest throughput requested can be chosen to calculate the contention window size, and its contention window size can be selected as the base number of the system. The throughput of the other nodes will be scaled by the contention window size.

### 6.1.6 Simulation Result

In this section, we will discuss the simulation results that verify our theory and algorithm. The simulations were performed on the Network Simulator (NS-2) [75] with a built in IEEE802.11 protocol. We chose IEEE802.11, because it is one of the most widely used MAC protocols over wireless networks. The environment we built was made up of a coordinator receiving message from multiple nodes. The coordinator accepted connections from source nodes, and each source node sent packets by way of the UDP protocol with a size of 160 bytes at various transmission rates. The requested QoS of each node was selected based on the continuous uniform distribution.

The algorithm was tested in three aspects: 1) one simulation result concerns the unsaturated network, which reveals the interaction among the contention level, packet loss probability and transmission throughput; 2) one simulation result includes content concerning the relation of contention level packet loss probability and transmission throughput in the saturated network, and several simulation results are presented to verify the accuracy of the QoS assurance algorithm in the saturated network; 3) and, some simulations were conducted to verify our theory about a network with a mix of saturated and unsaturated nodes.

### Unsaturated Networks

As we mentioned in Section 6.1.3, the algorithm for unsaturated networks is simple; this is because all requests are satisfied already. Because the link layer delay is lower than the transmission interval, the transmissions are not interrupted. The only thing that should be considered in unsaturated networks is the need to lower packet loss probability, which will create extra available network resources for upcoming transmissions. The approach is to enlarge the contention window size, which may generate additional link layer delay for the transmission as well. Figure 6.4 presents a simulation about this situation. We created a scenario with 14 source nodes surrounding the coordination node. The average packet loss probability and the total network throughput, calculated by the number of packets, was measured, to examen if their behavior match our theory when the contention window size increases. In this simulation, we increased the contention window size of all source nodes simultaneously.

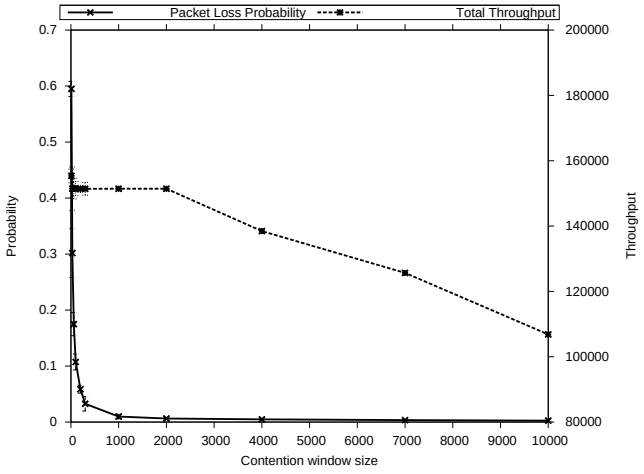



Figure 6.4: Packet Loss Probability and Total Throughput in Unsaturated Networks

Figure 6.4 shows that the packet loss probability drops from 60% to around 1% when contention window size is in the range of $[10, 1000]$. During this huge change in packet loss probability, the total network throughput is still not the same. The reason for this is that the decrement in packet loss probability reduces the link layer delay, which makes it even lower than the transmission interval. However, when the contention window size is larger than 2000, the total throughput starts to decrease. In this case, because the packet loss probability is too low, the change in packet loss probability during the increment in the contention window size can be ignored. The decrement in the total throughput is caused by the increment in the contention window and the stable packet loss probability. Every packet transmission failure will trigger a large backoff time, which creates a large link layer delay among the packets. Since the packet loss probability is the same, a larger contention window size leads to a larger link layer delay. When the link layer delay is larger than the transmission interval, the source node becomes saturated and the transmission rate is lowered. Therefore, in order to lower the packet loss probability and to preserve the transmission rate of the unsaturated network, a suitably large contention window size should be selected.

**Saturated Networks**

A similar simulation is conducted in saturated networks to verify our theory concerning saturated networks. The transmissions of the saturated nodes are always limited by link layer delay, since the link layer delay is larger than the transmission interval. By changing the contention window size, it is possible to control the link layer delay, because the contention window can affect both the packet loss probability and the backoff time when a transmission fails. In the simulation, 14 source nodes were created around a coordination node, in which all source nodes transmit at a full transmission rate. All nodes are saturated. We measure the packet loss probability and the total throughput, in terms of the number of packets in the network, when the contention window size increases.

The simulation result is presented in Figure 6.5. The packet loss probability is similar to the simulation result of unsaturated networks when the contention window is in the range of $[10, 1000]$. However, the total throughput is different from that of an unsaturated network. It did not have a stable period, in which the total throughput does not change. The total throughput decreases in most of the cases, because the large contention window size generates larger a backoff time when a transmission fails; and, this creates extra link layer delay for the transmission. However, there are two exceptions in Figure 6.5: when

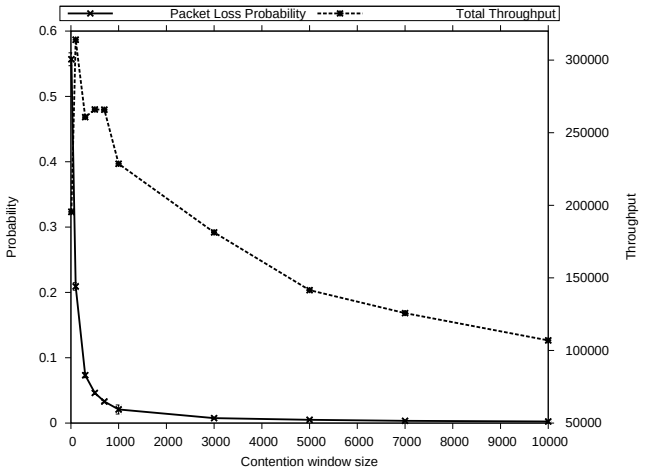



Figure 6.5: Packet Loss Probability and Total Throughput in Saturated Networks

the contention window size is 100, its total throughput is much higher than that of the contention window size 10; this is because the packet loss probability decreases 35% in this case. When contention window size range between 300 to 700, there is a small stable range in the total throughput; this is because the effect of packet loss probability decreases and the effect resulting from backoff increment is neutralized.

To verify the proportional algorithm that we presented in section 6.1.4, several simulations are conducted in a one hop scenario. The criteria measured in our simulations were the success rate, which measures whether or not the assigned throughput of each source node meets its requested throughput. Success rate measures the percent the requested requirement successfully satisfied. In the simulation, each node requests a specific throughput to transmit data. There are two types of source nodes: guaranteed source nodes and Best Effort source nodes. Our algorithm attempts to satisfy the requests of guaranteed nodes, while Best Effort source nodes do not receive any guaranteed performance. We only measured the success rate of the guaranteed source nodes in our simulation. If there is available bandwidth, a coordinator will accept the connection and reserve the specified throughput of the source nodes. The requirement of these source nodes become guaranteed. Therefore, in a situation where the network is under light workload conditions, every source node becomes guaranteed. Figure 6.6 presents the simulation results when the network performs under a light workload network. We compare our algorithm to the DCF in the simulations. DCF is unable to satisfy all the requests from the source nodes, because it does not consider the different QoS requests from each source node. In contrast, under a network with a light workload, our Con-

tention Window algorithm and Overload Control algorithm are able to provide almost complete satisfaction for the source node requests.

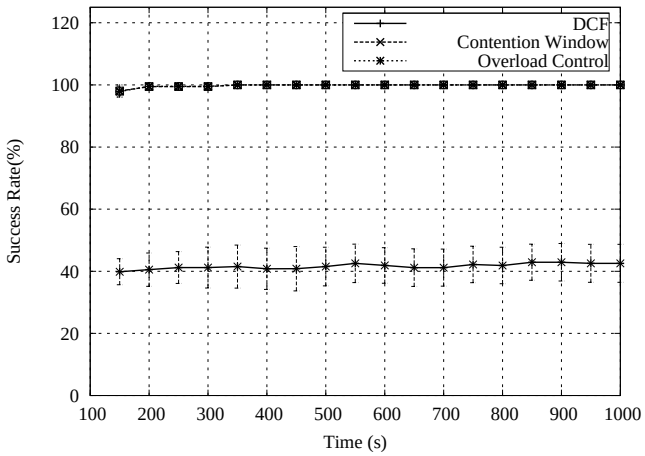



Figure 6.6: Simulation Results under a Lightly Loaded Network

If there is a lack of network resources while a node requests for transmission, the coordinator will still accept the connection but the source node will not guarantee network usage. Only a portion of the source nodes will receive the reserved resources. Our simulation used the First Come First Served approach to decide which source nodes would become guaranteed source nodes for all tests in the algorithms. Figure 6.7 illustrates the simulation results, after it has performed in a heavily loaded network. In this simulation, we compare our algorithm with the DCF and the MILD based algorithm. The overall requested resources are 30%-50% exceeding the channel capacity. The MILD based algorithm has three priority classes: high priority, low priority and Best Effort. The total requested throughput is 50% higher than the channel capacity. Our contention window algorithm still performs better than the DCF algorithm, but it is unable to satisfy most of the source nodes. The reason for this is that our basic contention window algorithm makes QoS decisions based solely on the requests. In order to consider the actual network utilization, we introduce an Overload Control Algorithm; this optimizes the system performance over an overloaded network. Our Overload Control algorithm is superior to the MILD based algorithm, even though the MILD based algorithm has a QoS classification mechanism. The MILD based algorithm sets the contention window range only according to the priority. It does not consider the specific requested QoS and the relationship between these requested QoS values.

Figure 6.8 shows the difference between requested QoS and assigned QoS during the

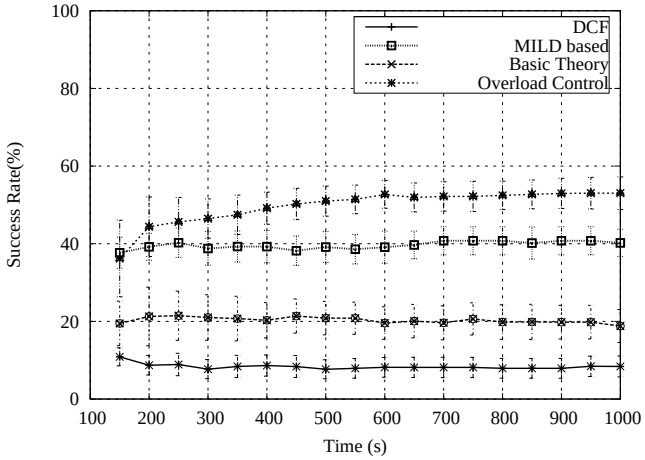



Figure 6.7: Simulation Result under a Heavily Loaded Network

heavily loaded network transmission simulation, used to measure the QoS fairness of each algorithm. The QoS fairness that we consider is the assigned resource which should be the same as the requested resource. If more resources are assigned, they will be wasted. If less resources are assigned, the QoS will not be guaranteed. We can see that our Overload Control algorithm manages the difference between these two parameters at around 5%, while the difference between the requested and assigned QoS of DCF and MILD based algorithms is over 20%. Even though the MILD based algorithm can be satisfactory, it wastes several resources on the unrequested portions. Though our algorithm fails some QoS requirements, it is still able to assign resource to the source nodes that is approximate to the requirement. Therefore, despite the fact that some source nodes not fully satisfied, they have been partially satisfied.

Based on the simulations, we can see that our algorithm can achieve one hop QoS fairness in our scenarios. It successfully split network capacity based on a fair resource distribution algorithm; and, it assigns the specific contention window size to each source node dynamically, without selecting a fixed contention window size for a group.

### Mix of Saturated and Unsaturated Nodes

In the scenario in which there is a mixed of saturated node and unsaturated nodes, it is difficult to determine the effect of the link layer delay on saturated and unsaturated nodes. Based on our analysis, the link layer delay will not interrupt the transmission of unsaturated nodes, and the transmission of saturated nodes will be limited by the link layer delay. The coexistence of saturated nodes and unsaturated nodes should be observe

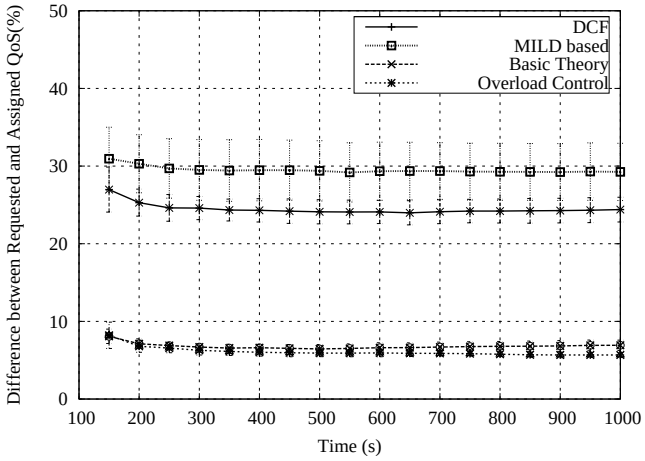



Figure 6.8: Difference between Requested and Assigned QoS on Heavy Load Network

to verify our theory. A simulation of the one hop scenario was created to explain the situation just mentioned. All nodes are unsaturated at the beginning, and the contention window size of all nodes is increased simultaneously to add an extra period of link layer delay. Like Figure 6.4, when the contention window size is large enough, it will add additional period of link layer delay to each transmission. The transmission rate of the nodes increases based on the numbers assigned to the nodes. We measure the total throughput of each node in terms of the number of successfully delivered packets.

The simulation result is shown in Figure 6.9. When contention window size is at 100 and 3000, the link layer delay is still low and the transmissions are not interrupted. However, when contention window size becomes 5000, node 7 to node 9 become saturated, and their throughput is limited by the contention window; thus, the throughput of these nodes become the same. When the contention window size gets larger, more nodes become saturated and the throughput becomes lower; this is because link layer delay becomes larger. The throughput of node 1 to node 3 is still the same in all contention window sizes; this is because the link layer delay is not large enough to interrupt this throughput. In the simulation result, it shows that only the nodes with a faster transmission rate will be affected when the contention window size is increased. It verifies our assumption in Section 6.1.5 that the transmission rate of the unsaturated nodes is lower than the transmission rate of the saturated nodes in the mixed scenario.

To further verify our theory about the contention among the saturated and unsaturated nodes and how it will affect the throughput performance of the saturated nodes, we created another simulation, in which all nodes have a contention window size of 10000 at

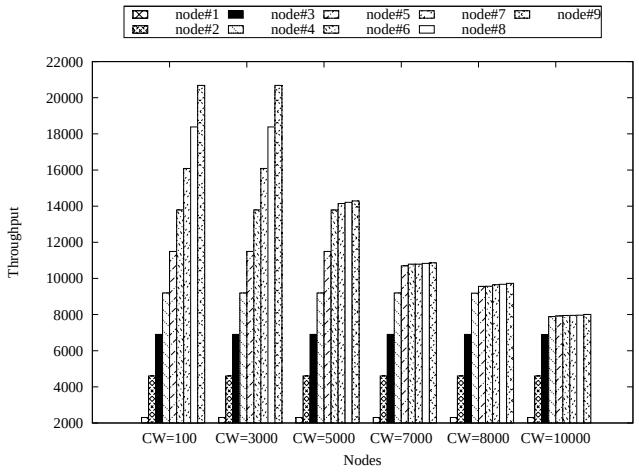



Figure 6.9: The Mixed Scenario Obtained by Increasing the Contention Window Sizes

the beginning. The contention window size recovers to normal one window at a time; and, in order to show the increased contention of the network, we will lower the throughput of the saturated node. In Figure 6.10, it can be observed that when more and more nodes are recovered from a state of high contention window size, the throughput of saturated nodes become lower. The reason for this is that the decreased contention window size of the unsaturated node eventually increases the packet loss probability; this adds extra link layer delay to the saturated nodes and lower its throughput. When all contention window sizes become normal, all nodes become unsaturated. Therefore, the throughput decrement of the saturated nodes is not affected by the limit of the network capacity in this simulation.

The algorithm in section 6.1.5 indicates that the throughput of saturated nodes will be in a proportional relation and will be based on the contention window size that it is assigned. The base number of the proportional relation is the critical issue in the algorithm; it can affect the throughput of saturated nodes. If the base number is too small, it will add extra packet loss probability to the network. If the base number is too large, it will increase the link layer delay of the source nodes. The effect of the base on the throughput is presented in Figure 6.11. In this simulation, we create a scenario with 9 nodes, in which 4 nodes are unsaturated and 5 nodes are saturated. To ensure that the saturated nodes are fully saturated, they are made to transmit at a full rate. The contention window size of the saturated nodes are of a proportional relation. The base number of the proportional relation was chosen from the range of $[100, 8000]$. The unsaturated nodes are all of a normal contention window size. Figure

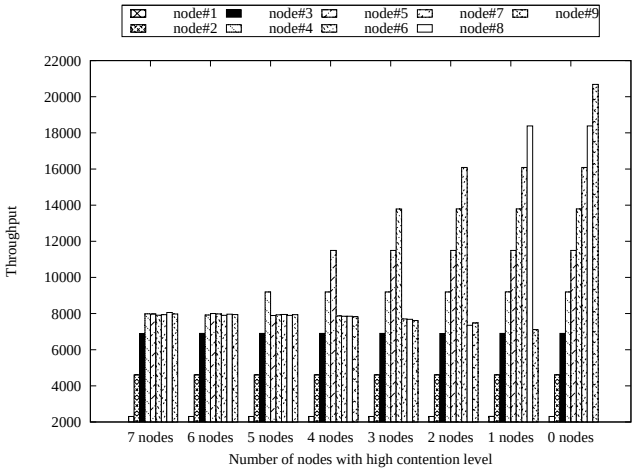



Figure 6.10: The Mixed Scenario by Increasing some of the Contention Window Sizes

6.11 shows that with a different base number, the throughput of saturated nodes will become different. The larger base number leads to lower throughput, because additional link layer delay is inserted in the transmission. However, from Figure 6.12, we can see that the proportional relation is not broken even though the base numbers are different. Therefore, the proportional relation is not related to the base number.

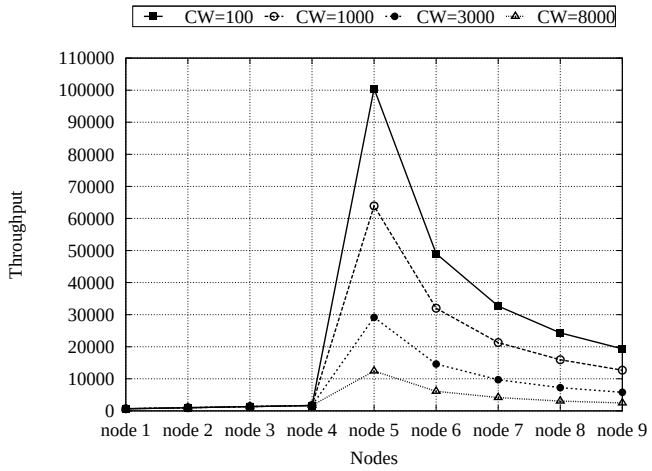
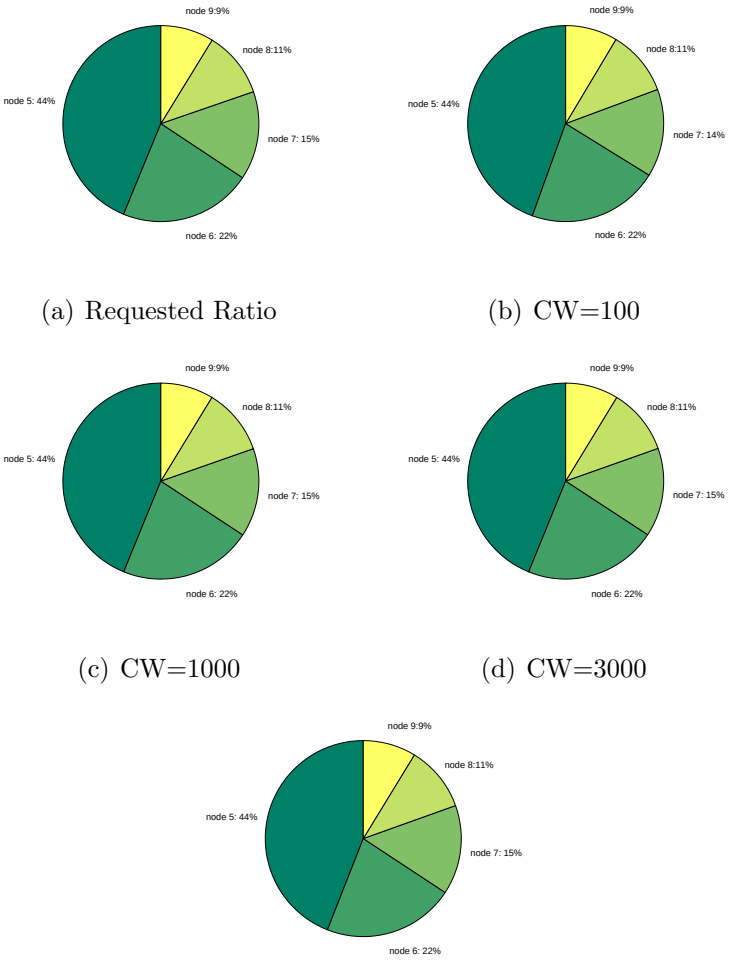



Figure 6.11: The Mixed Scenario with Different Base Numbers of CW Sizes

### 6.1.7 Summary

Recently, most of the studies on contention window adjustment were based on QoS classification, which separates the requests into different categories. A pre-fixed contention




(a) Requested Ratio (b) CW=100

(c) CW=1000 (d) CW=3000

(e) CW=8000

Figure 6.12: The Ratio Relation of Saturated Throughput in Different Base Numbers

window size is assigned to each category in order to limit or guarantee the transmission throughput. However, the pre-fixed contention window size is unable to satisfy all the requests during runtime; this is because it does not consider the composition of the requests and the slight difference among the requests in the same category. We propose a novel contention window adjustment mechanism, which considers the network in three one hop scenarios: 1) all nodes are unsaturated, 2) all nodes are saturated and 3) there is a mix of unsaturated and saturated nodes. In these three scenarios, the contention window adjustment mechanism serves different functions, which will cause different effects. Several simulations are conducted and the simulation results are presented to verify our

algorithm. The results show that our theory on link layer delay and contention window size is correct in different scenarios. The results for the saturated network show our algorithm with a proportional relation can achieve higher QoS satisfaction and higher QoS fairness. Even in the mixed scenario, our theory about both the unsaturated and saturated nodes is also proven correct.

## 6.2 TCP-CC: Cross Layer TCP Pacing Protocol by Contention Control

Transmission Control Protocol (TCP) has poor performance over wireless networks. Some researches indicate that TCP congestion control mechanism may cause burstiness in traffic flow. Numerous TCP segments are delivered simultaneously, while an acknowledgement of a retransmission is successfully received. The burstiness leads to a highly contentious network, which markedly increases the probability of packet loss on wireless networks. TCP pacing is one of the solutions for TCP burstiness on multi-hop networks. The idea is to spread the TCP segment transmissions over the whole Round Trip Time (RTT). Most of the pacing protocols attempt to insert a delay interval into the TCP transmissions. However, there is a similar pacing algorithm on IEEE802.11, which is the contention window mechanism. In this research, we first measure and analyze the way that the contention window size affects TCP throughput in different scenarios. We propose a cross layer TCP pacing protocol by contention control on MAC layer, called TCP Contention Control (TCP-CC). It adjusts the lower bound of the contention window in order to optimize the overall TCP throughput on both one hop and multi-hop topology. Finally, comparative simulations are conducted in order to verify the improvements of our protocol on both TCP Reno and TCP Vegas.

### 6.2.1 Introduction

With the rapid spread of wireless Ad Hoc networks, most existing protocols no longer satisfy the requirements of various applications over multi-hop wireless networks. Transmission Control Protocol (TCP) [85] is a well-known example of a protocol that performs poorly over wireless networks. Though new techniques have been proposed for replacing TCP in several areas, such as the Real-time Transport Protocol (RTP) for multimedia, TCP is still one of the most widely used protocols, because it is can achieve flow control and congestion control, and provides reliability over the network. Another important

reason for TCP's current popularity is its ease of use, as well as the fact that most current software still runs on TCP [69].

The purpose of the TCP congestion window is to avoid congestion over TCP transmissions. Based on the TCP protocol, each packet loss, regardless of the cause (whether time out or duplicate acknowledgements), which leads to the decrement of the congestion window size. The reason for this is that TCP always treats the cause of any TCP segment loss as congestion. This assumption is correct only in a stable network environment, where packet loss on the lower layer is rare. However, packet loss that occurs over wireless networks is mainly due to contention, not congestion [41], on the Media Access Control (MAC) layer, and this packet loss is very common in wireless networks because of fading and interference. Even though the channel is idle, a packet will possibly be lost in wireless networks due to fading or interference, which leads to the decrement of the congestion window size with unnecessary bandwidth loss.

Most researchers focus on the optimization of the congestion control mechanism [69][41][88], by modifying queueing management, delay acknowledgement or congestion window adjustment. However, these developments are limited, because the performance of TCP is easily interrupted by the unstable conditions in wireless networks. TCP is unable to confirm the cause of segment loss, because the lower layer is hidden from TCP. Most of the protocols proposed on the TCP layer are fragile when faced with an unstable wireless network.

Other researchers attempt to optimize the TCP performance through the cross-layer technique [97][126] [79] [62][30]. Since observation supports the notion that contention is the main reason for TCP segment loss in wireless network [41], contention avoidance eventually improves TCP performance.

Burstiness remains one of the main causes of poor performance of TCP in wireless networks. Observation has indicated that bursty traffic caused by congestion control mechanisms degrades TCP performance [2]. The causes of TCP burstiness are slow start, segment losses, Acknowledgement (ACK) compression and Multiplexing. It can be summarized in two scenarios: 1) buffer overflow or 2) a large number of segments sent simultaneously. Wireless network buffer overflow is rare with a reasonably large buffer [41]. Therefore, we will focus on the second scenario in our research, by spreading a large number of TCP segments over the whole RTT with TCP pacing. TCP pacing eventually circumvents contention by eliminating burstiness. Currently, research regarding TCP pacing is mainly focused on the transport layer [36] [8] [63] [76] [42] [71].

To the best of our knowledge, we are the first to present the cross-layer measurement

and analysis on the lower bound of the contention window and TCP performance. In this paper, we propose a novel TCP pacing protocol by contention control in both one-hop and multi-hop scenario. The existing TCP pacing protocols ignores the potential pacing techniques in IEEE802.11, known as the deferring and backoff mechanisms. Each packet on the Media Access Control (MAC) waits for a deferring time if the channel is busy. Over a saturated network, nearly each MAC packet is eventually deferred through a time calculated by the contention window size, and hence the packet flow is paced. By measuring and analyzing the effects of the contention window size on TCP throughput in different scenarios, we propose a cross-layer TCP pacing protocol called TCP-CC. It functions using contention control on the MAC layer, which adjusts the lower bound of the contention window with the aim of increasing the overall TCP throughput. By achieving contention control, the performance of the TCP protocol is improved without any modifications. In a one-hop scenario, a centralized protocol is conducted in order to optimize the TCP throughput. It is difficult to create a centralized coordinator in multi-hop scenarios due to high resource consumption. We propose a dynamic distribution protocol to select the corresponding lower bound of the contention window for multi-hop TCP transmissions.

### 6.2.2 Measurement and Analysis in Different Scenarios

In this section, the relationship between the contention window size and TCP throughput are tested and measured utilizing various scenarios in wireless networks, such as the one-hop topology, the chain topology, the cross topology, and the grid topology. Simulations are conducted in order to demonstrate how contention window size affects the TCP throughput. As we know, the length of the transmission path will cause the degradation of TCP throughput, because RTT increases with higher packet loss probability and longer paths. The contention window size is also one of the factors affecting the packet loss rate. A greater contention window size leads to lower packet loss probability but longer link layer delay, which will reflect on the RTT. In contrast, a smaller contention window size causes an increase in packet loss probability, but packets have a shorter waiting time for the retransmissions. To measure this tradeoff of contention window size in relation to the TCP throughput, we set up several simulations on NS-2. These simulations are conducted over different topologies to indicate that this tradeoff is rarely affected by the topology and interference.

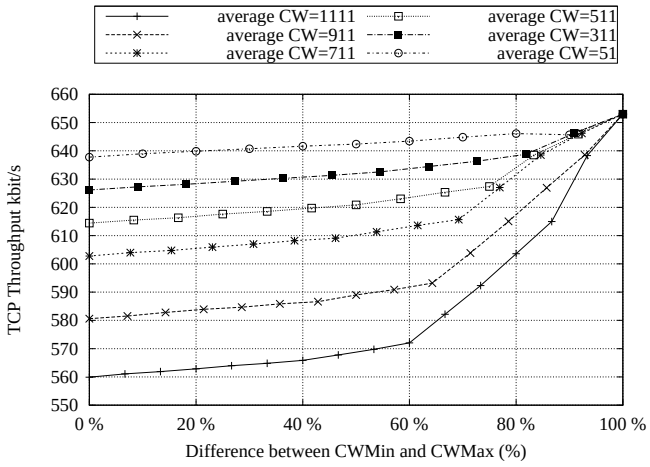



Figure 6.13: One hop TCP transmission with different average contention window size

### One Hop Topology

Contention in one-hop TCP transmissions mainly originates from several terminals trying to access the same media. We ran a simulation with 5 terminals to start TCP transmissions to the same one-hop end with different average contention window sizes, as seen in Figure 6.13. We measured the overall TCP throughput of the network's successfully received TCP segments, which is an accumulation of the TCP throughput of all the terminals, by enlarging the contention window range for each average contention window size. While the average contention window size increases, the overall TCP throughput also increases. When the difference between $CWMin$ and $CWMax$ is 100%, denoting that $CWMin$ is very small, the overall throughput of various average contention window sizes is the same. The reason for this occurrence is that if $CWMax$ is unreasonably large, it will be ignored. Based on this observation, the contention window size is able to affect the TCP throughput in the one-hop scenario.

### Chain Topology

In addition to the one-hop scenario, multi-hop scenarios will also be discussed. The chain topology is the basic multi-hop topology. In our simulation, we make sure that the distance between each node is almost equal to the maximum transmission range, and packets will be passed from one hop to another hop, following the pass. A TCP traffic flow is also passed through the entire path, from the first terminal to the last. We measured the overall TCP throughput for the TCP segments that are successfully received.

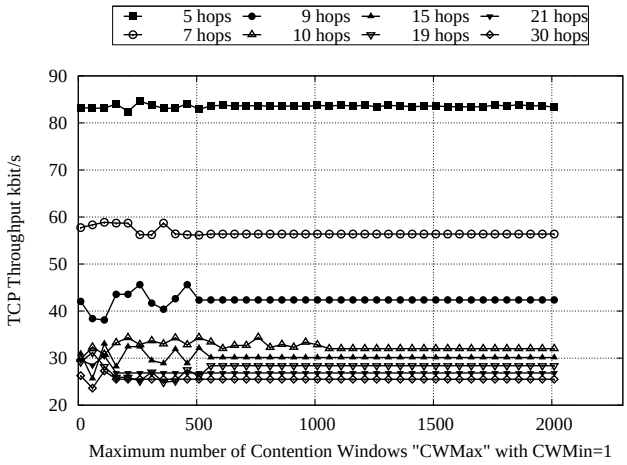



Figure 6.14: Fixed lower bound of contention window range of Chain topology

We first tested the TCP throughput by increasing the maximum contention window size, $CWMax$, along the fixed minimum contention window size, $CWMin$. We see from the simulation results in Figure 6.14 that the TCP throughput hardly changes. Therefore, the range of the contention window size is not a factor of TCP throughput. It also illustrates that the upper bound of the contention window size does not affect the TCP throughput or have less effect on the TCP throughput. While the upper bound of the contention window becomes too large, it seems to be impossible to reach the upper bound, and the range is useless. That is the reason that no oscillation exists on the TCP, while the upper bound is becoming unreasonably large, more than 500 in Figure 6.14.

Another simulation on chain topology increases the average contention window size with fixed small contention window ranges, as seen in Figure 6.15. As we discussed in the previous simulation, the range of the contention window size is irrelevant to the TCP throughput. We ran the simulation on various lengths of chain paths. We can see from the results that with a 4 degree polynomial trend line there is a point of contention window size that leads to the maximum TCP throughput from 5 hops to 9 hops. While the length of the path increases, the point of contention window size increases. The reason for this is that the packet loss rate of chain topology with a large number of hops is actually very high because of the interference and collision between data flow and acknowledgement flow. A larger contention window range is needed to absorb such network degradation.

By excluding the upper bound of the contention window size in Figure 6.14, we test the lower bound of the contention window size, $CWMin$. The simulation is designed

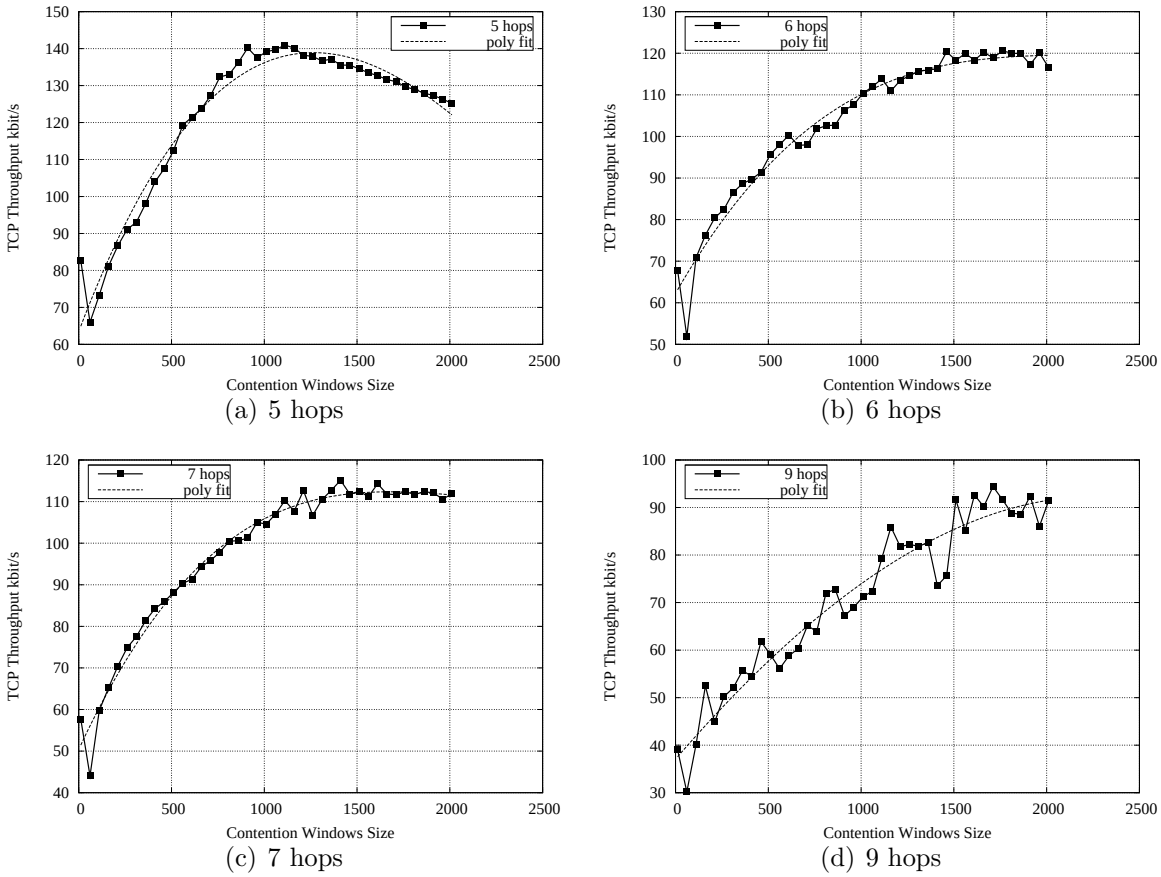



Figure 6.15: Different average CW size with different length of paths of Chain topology

to increase the lower bound of contention window size with a fixed average contention window size in Figure 6.16. Results show that as $CWMin$ increases, TCP throughput with fixed average contention window size also increases. Therefore, we conclude that $CWMin$ is one of the major factors that should be considered on the TCP throughput in the chain topology. We will test this theory on more complex topologies, cross topology and grid topology, in the following sections.

### Cross Topology

The cross topology is more complicated than chain topology because the two traffic flows try to access the same media simultaneously. The level of contention in this scenario is demonstrably higher compared to the chain topology. In order to measure the tradeoff of the contention window size, we set up several simulations with different criteria. We also measure the overall TCP throughput of the whole network based on the successfully

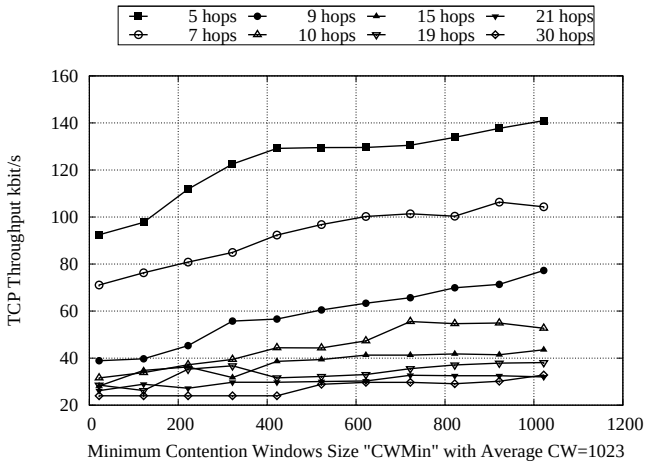



Figure 6.16: Fixed average contention window size of Chain topology

received TCP segments in the simulation, which is an accumulation of the throughput of all TCP flows in the network. During the simulation, the contention window range of both TCP flows will be changed.

To verify that our theory on the cross topology is similar to that of chain topology, we first conducted a simulation on fixed average contention window size with increasing $CWMin$. Figure 6.17 shows similar results to chain topology. As $CWMin$ increases, so does the TCP throughput. In cross topology, $CWMin$ is also one of the main factors to reduce contention in TCP transmissions.

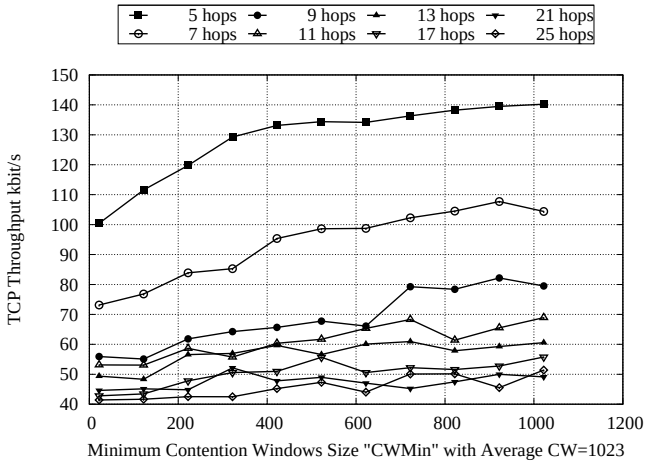



Figure 6.17: Fixed average contention window size of cross topology

Figure 6.18 is the result of a simulation that increased average contention window size in different sizes of networks with a fixed contention window range. Enlarging the

contention window size eventually increases the deferring time and the backoff time, which inserts a delay among outgoing packets. The inserted delay reduces the chance of the occurrence of burstiness and leads to an increase in TCP throughput. When the average contention window size is unreasonably large, overall TCP throughput stabilizes or even decreases. The reason for this is that if the inserted delay is too large, the channel will be idle too frequently to wait for a defer or backoff timer, and the network resource are not sufficiently occupied. Therefore, we can infer that there is a point in the contention window size that optimizes the TCP throughput in cross topology. To further verify our theory, a grid topology test has been conducted in the next section.

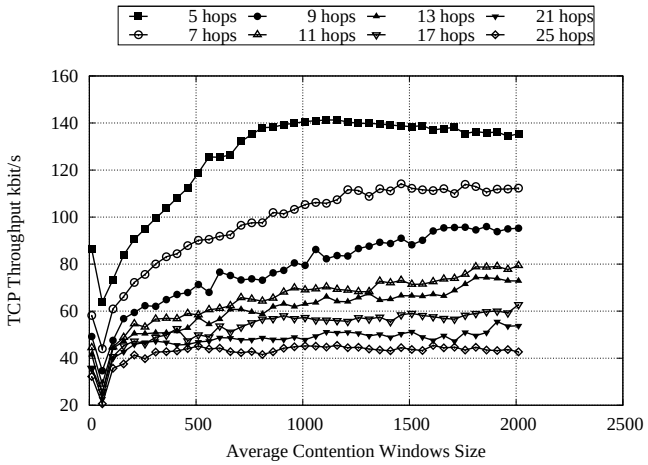



Figure 6.18: Increasing average CW size with different size cross topology

### Grid Topology

The grid topology scenario is an extension of cross topology. By inputting more traffic into the scenario, we measure the overall throughput for successfully received TCP segments as well. The same simulation is achieved in the grid topology scenario with increasing $CWMin$ and a fixed average contention window size. As we can see in Figure 6.19, the results are identical to chain topology and cross topology.An increase in $CWMin$ is accompanied by an increase in overall TCP throughput. Moreover, when $CWMin$ is over 300 at $3 \times 3$ grid, the throughput stabilizes or decreases slightly. It matches our theory that unreasonably large $CWMin$ is useless or even lowers the overall TCP throughput. This situation does not occur in chain topology or cross topology, because there are more available paths and there are more potential resources to be occupied in grid topology. Packets can be transmitted faster in order to utilize the resource

sufficiently with a smaller delay interval.

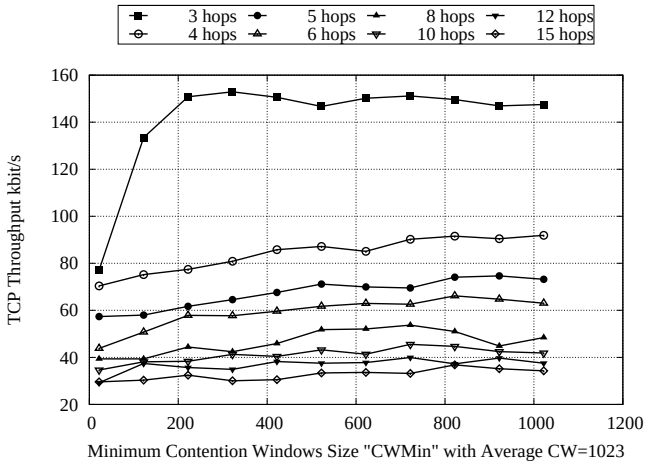



Figure 6.19: Fixed average contention window size of grid topology

Figure 6.20 shows the simulation results with different average contention window sizes and a fixed contention window range in different grid sizes. When the network size is small, we can clearly see that there is an upper bound of the traffic flow with a proper average contention window size. When the contention window size increases enough, the overall TCP throughput stabilizes or even decreases, similarly to cross topology. Compared to cross topology, the point to maximize the overall TCP throughput is smaller. The same reasons seen in Figure 6.19 can be used to explain this situation.

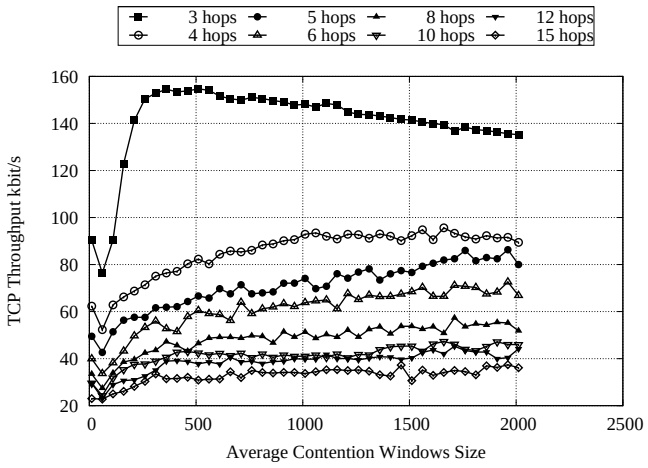



Figure 6.20: Different average CW size with different sizes grid topology

**Summary**

The contention window mechanism in IEEE802.11 introduces several features for wireless networks. Packet deferring is a delay interval applied to the MAC traffic flow before it accesses the media, which is decided by the contention window size. Therefore, TCP segments at various rates will be split and paced into smaller MAC packets with deferred intervals in a saturated network. In the contention window mechanism, a minimum contention window size $CWMin$ and a maximum contention window size $CWMax$ will be defined in order to restrict the range of the contention window size. When a packet is dropped or lost during the transmission, the contention window size will increase. When a packet is successfully received, the contention window size will become $CWMin$.

In one-hop TCP transmission, the TCP throughput prefers smaller $CWMin$, based on the observations from our simulation. This is so that it can access media faster when it fails. Contention is not a concern in one-hop transmissions when compared to fast retransmission times.

As we discussed in the previous section, in multi-hop TCP transmissions, $CWMax$ is not a main factor for TCP throughput because it is ineffectual when it becomes incredibly large, and thus the contention window size will never reach the upper bound. The reason why $CWMin$ is a factor is that the contention window size will return to $CWMin$ while a packet is successfully transmitted. If $CWMin$ is too small, the delay interval inserted between packets can be disregarded. In order to achieve TCP pacing in a multi-hop topology with the contention window mechanism, a suitable size for the lower bound of the contention window should be assigned to pace the TCP traffic flow.

Based on our observations, one-hop and multi-hop scenarios are different. We should consider them separately, and different protocols should be proposed for each of them. In the next section, by modeling the TCP throughput and MAC delay, we measure TCP performance with the derived equations on wireless networks.

### 6.2.3 TCP Throughput, RTT and Segment Loss Probability Estimation

In our analysis, we make some reasonable assumptions in order to simplify the problem. We assume that the network is in a saturated condition, which means all TCP sources always have a segment to send and all queues are non-empty. We also assume that RTT is independent of congestion window size [78], which is only affected by network conditions. Two TCP throughput models will be included in this section: TCP Reno

and TCP Vegas. Both models only involve TCP segment loss probability and RTT. The purpose of the analysis in this section is to model the effect of the contention window size to TCP throughput. In the following sections, we develop an algorithm for TCP throughput optimization, based on the derived models.

The model in this paragraph aims to derive an equation for one-hop TCP Reno transmission. [78] proposes a TCP throughput model of TCP Reno, as shown in Equation 6.23 and Equation 6.24. $E[B]$ denotes the mean number of TCP throughput. $E[RTT]$ denotes the mean number of RTT. $b$ denotes the number of segments each ACK has acknowledged. $T_0$ denotes the initial timeout. $p$ denotes the segment loss probability. $W_{max}$ denotes the maximum congestion window size. Our cross-layer TCP optimization protocol discusses an RTT estimation and segment loss probability estimation for deriving the corresponding TCP throughput.

$$E[B] \approx \frac{1}{E[RTT]\sqrt{\frac{2bp}{3}} + T_0 \min\left(1, 3\sqrt{3bp/8}\right) p(1+32p^2)} \tag{6.23}$$

$$B(RTT, p) = \min\left(\frac{W_{max}}{E[RTT]}, E[B]\right) \tag{6.24}$$

[93] introduces a TCP throughput model for TCP Vegas. Equation 6.25 is the throughput estimation of TCP Vegas, which only involves TCP segment loss probability and average RTT. $n$ denotes the expected number of consecutive Loss Free Periods (LFP) that occur between one slow start (SS) and another slow start period, which can be derived from TCP segment loss probability [93]. $N_{SS2TO}$ is defined in Equation 6.26, where $W_0$ denotes the average congestion window size during a stable backlog state and $T_{TO_0}$ denotes the average duration of the rst TO in a TO series. $D_{TO}$ is defined in Equation 6.27.

$$E[B]_{Vegas} \approx \frac{(n+1)\left(\frac{1-p}{p} + W_0\right) + \frac{2p(2-p)}{(1-p)^2}}{N_{SS2TO}RTT + D_{TO}} \tag{6.25}$$

$$N_{SS2TO} = 2logW_0 + (n+1)\frac{1-p}{pW_0} + (1 + \frac{n}{4}\frac{W_0}{8}) + \frac{9n}{8} - \frac{11}{4} + \frac{4 - 2^{logW_0}}{W_0} \tag{6.26}$$

$$D_{TO} = \left( \frac{64}{(1-p)^2} - 321 \right.$$
$$\left. +(1-p)^2 \sum_{k=1}^{6} ((2^k - 64k + 320)(2p - p^2))^{k-1} \right) T_{TO_0} \quad (6.27)$$

**RTT Estimation**

In these two TCP throughput models, we only consider the RTT and segment loss probability in the equation because the parameters of the network condition on the lower layers are hidden by the OSI model. With the cross layer technique, RTT and segment loss probability can eventually be estimated. RTT is composed by the time to transmit the data and the time to transmit the ACK. Equation 6.28 indicates the composition of RTT. $E[T_{data}]$ denotes the mean number of time to transmit a data segment. $E[T_{ACK}]$ denotes the mean number of time to transmit the ACK. The transmission time of data segment and ACK can be split in the following ways: 1) time to wait in the queue $E[t_q]$; 2) time to transmit the segment $E[t_{data}]$ and $E[t_{ACK}]$; 3) delay on link layer $E[t_{LLD}]$ and $E[t_L LA]$. Equation 6.29 and 6.30 present the composition of $E[T_{data}]$ and $E[T_{ACK}]$.

$$E[RTT] = E[T_{data}] + E[T_{ACK}] \quad (6.28)$$

$$E[T_{data}] = E[t_q] + E[t_{data}] + E[t_{LLD}] \quad (6.29)$$

$$E[T_{ACK}] = E[t_q] + E[t_{ACK}] + E[t_{LLA}] \quad (6.30)$$

Because all queues are full in a saturated network, the time spent waiting in the queue of each segment is approximately equivalent. Therefore, for the remainder of this section we will only explain how we estimate the remaining composition of the RTT. $E[t_{data}]$ is equal to the length of data segment $L_{data}$ over current bandwidth $r$, while $E[t_{ACK}]$ is equal to the length of acknowledgement $L_{ACK}$ over current bandwidth $r$. $N$ denotes the number of frames in each TCP segment after fragmentation.

$$E[t_{data}] = \frac{L_{data}}{r} N \quad (6.31)$$

$$E[t_{ACK}] = \frac{L_{ACK}}{r} \quad (6.32)$$

Link layer delay $E[t_{LLD}]$ presents the delay caused by the contention ARQ mechanism. The sender will send a Request To Send (RTS) to the receiver and wait for a CTS message in order to ensure that the channel is idle. If the message is timed out within $E[TO_{LL}]$ whilst waiting for the CTS, the sender assumes that the RTS is lost and enters a backoff phase. $E[BF]$ denotes the mean number of backoff time. $E[N_{rts}]$ denotes the mean number of times transmitting a RTS when a data frame needs to be sent. $E[N_{data}]$ denotes the mean number of times to transmit a data frame. $E[TO_{LL}]$ is equal to the maximum estimation of network propagation delay $PDelay_{max}$. The link layer delay of acknowledgement $E[t_{LLA}]$ is similar to $E[t_{LLD}]$, without multiplying $N$.

$$E[t_{LLD}] = \left(\frac{L_{rts}}{r} + \frac{L_{cts}}{r} + E[delay_{LL}](E[N_{rts}] - 1)) + E[delay_{LL}](E[N_{data}] - 1)) + \frac{L_{ACK_{LL}}}{r}\right)N \tag{6.33}$$

$$E[delay_{LL}] = E[BF]slotTime + E[TO_{LL}] \tag{6.34}$$

$$E[TO_{LL}] = \frac{L_{rts}}{r} + \frac{L_{cts}}{r} + 2 * PDelay_{Max} \tag{6.35}$$

We consider the average number of transmission time of an RTS and data frame in order to estimate the average time of delay caused by the link layer. Therefore, we consider the conditions when eventually the transmission either succeeds or fails. Equation 6.36 is the formula for $E[N_{rts}]$, while $F_{rts}$ denotes the failure probability of RTS transmission and $N_{rts_{max}}$ denotes the maximum number of retransmission of the RTS. After simplification, we get Equation 6.37.

$$E[N_{rts}] = (1 - F_{rts}) + 2F_{rts}(1 - F_{rts}) + 2F_{rts}^2(1 - F_{rts}) + (N_{rts_{max}} + 1)F_{rts}^{N_{rts_{max}}}(1 - F_{rts}) + (N_{rts_{max}} + 1)F_{rts}^{(N_{rts_{max}}+1)} \tag{6.36}$$

$$E[N_{rts}] = \frac{1 - F_{rts}^{(N_{rts_{max}}+1)}}{1 - F_{rts}} \tag{6.37}$$

The data frame transmission is similar to the RTS transmission, as shown in Equation 6.38 and 6.39, where $F_e$ denotes the failure probability of the data frame transmission, and $N_{data_{max}}$ denotes the maximum retransmission times of the data frame.

$$\begin{aligned} E[N_{data}] = (1-F_e) + 2F_e(1-F_e) + 2F_e^2(1-F_e) + \\ +(N_{data_{max}}+1)F_e^{N_{data_{max}}}(1-F_e) \\ +(N_{data_{max}}+1)F_e^{(N_{data_{max}}+1)} \end{aligned} \tag{6.38}$$

$$E[N_{data}] = \frac{1-F_e^{(N_{data_{max}}+1)}}{1-F_e} \tag{6.39}$$

As we see from the aforementioned equations, RTT is determined by the link layer packets failure probability, which refers to the network conditions. In the next section, we will explore the probability estimation over wireless networks.

### TCP Segment and Link Layer Packet Loss Probability

In this section, we provide estimation about RTS transmission failure probability $F_{rts}$, data frame transmission failure probability $F_e$, and TCP segment loss probability $p$. There are two reasons of RTS packet loss: Bit Error Rate (BER) and collision. Failure probability caused by BER is denoted as $F_{BER}$, while $L_{rts}$ is the length of s RTS packet. The failure probability of collision $F_c$ is determined by average backoff $E[BF]$ and the number of stations $N_{station}$ connecting to the same terminal. To obtain the $N_{station}$ parameter, cooperation of the receiver side is required. The probability of a terminal sending a packet at one time slot can be estimated as $\frac{1}{E[BF]}$ because the terminal usually needs to wait for a backoff time in order to send a packet in a saturated network, when there is a high chance of contention.

$$F_{rts} = F_{BER} + F_c \tag{6.40}$$

$$\begin{cases} F_{BER} = 1-(1-BER)^{L_{rts}} \\ F_c = 1-(1-\frac{1}{E[BF]})^{N_{station}-1} \end{cases} \tag{6.41}$$

As we assume at the start of the section 6.2.3, we assume that after receiving a CTS the channel model is clear and the data frame is not disrupted by collision. Therefore, collision probability is not considered in the data frame failure model. The causes of

data frame loss are BER and exceeding RTS retransmission time in Equation 6.42. $L_{data}$ denotes the size of a data frame.

$$F_e = 1 - (1 - BER)^{L_{data}} + F_{rts}^{N_{rts_{max}}+1} \tag{6.42}$$

To successfully transmit a TCP segment, all the data frames of this segment have to be received and acknowledged successfully. Therefore, the probability of a TCP segment loss is formulated as Equation 6.43.

$$p = 1 - (1 - F_e^{N_{data_{max}}})^N \tag{6.43}$$

With the aforementioned models, we can conclude that RTT is mainly determined by the parameters of the link layer. One of the most important factors is average backoff, which is decided by the contention window size. The contention and delay on the link layer affects the range of RTT.

### 6.2.4 TCP Throughput Maximization For The One Hop Scenario

Based on the aforementioned equations we derived, we are able to determine the corresponding TCP optimization protocol for a one hop scenario.We choose a robust algorithm in our optimization dilemma, and select the corresponding contention window size, which exhibits the highest TCP throughput return. As we mention in Chapter III, the main issue in TCP optimization in one hop scenarios is fast retransmission. TCP connections from different terminals try to fill all the idle channels and time slots in order to maximize the throughput rather than considering the collision probability.

The receiver acts as a coordinator to the network, which probes nearby terminals and assigns them the corresponding contention window size based on the probed information. Because it is a one hop connection, the overhead of the control message is very low. After assigning the corresponding contention window size, no further control messages need to be exchanged.

### 6.2.5 TCP Pacing for Multi-hop Scenarios

TCP optimization in multi-hop scenarios is different from that of the one-hop scenarios, because the packet loss probability is much higher due to the increased length of the path. However, a centralized solution is not suitable in multi-hop scenarios because

of the overhead of the control messages. Therefore, a distributed solution for multi-hop scenarios is proposed in this section. Section 6.2.2 indicates that a corresponding contention window size should be assigned to optimize the overall TCP throughput. The main purpose of the algorithm is to locate the suitable contention window range for the corresponding network topology.

### Distributed Algorithm

Section 6.2.2 indicates that a corresponding contention window size should be assigned to optimize the overall TCP throughput. Our distributed algorithm increases or decreases the lower bound of the contention window size by modifying the $CWMin$ and $CWMax$. It is difficult to calculate a corresponding $CWMin$ for a terminal, because of the unknown network topology, network capacity, contention level and even routing information. Therefore, our algorithm attempts to increase and decrease $CWMin$ to find the maximum point of the system, as shown in Algorithm 9. There are two operations, increment or decrement. It measures several criteria relating to TCP throughput, such as maximum average throughput $maxAvgTP$, current average throughput $avgTP$ and previous average throughput $preAvgTP$. By comparing the $avgTP$ and $maxAvgTP$, it knows if the throughput is capable of greater range. If $avgTP$ is larger than $maxAvgTP$, it keeps the current operation of $CWMin$. If not, it compares $avgTP$ and $preAvgTP$, to check whether the throughput is greater than the previous record. If yes, it keeps the current operation of $CWMin$. If not, it switches the current operation of $CWMin$. By probing the TCP throughput, our protocol eventually reaches the optimization point. The $Decrease$ and $Increase$ functions of $CWMin$ can be linear or exponential. We use linear functions in our simulations.

### Initial Lower Bound of Contention Window Size

To enhance speed and to find the optimization point of TCP throughput by our cross-layer TCP pacing algorithm, the initial $CWMin$ is the key. If we set an initial $CWMin$ close to the optimization point, the algorithm can find the optimization point more efficiently. Based on the one-hop RTT estimation in Equation 6.28, we can obtain the m-hop RTT estimation Equation 6.44. Instead of probing nearby terminals to find the density information of the transmission, it reduces the nearby neighbour to 2 terminals in order to estimate contention. One of these terminals is the upstream terminal and another is the downstream terminal. The rest of the calculation of $CWMin$ is similar

**Algorithm 9** Calculate corresponding $CWMin$ and $CWMax$

**Require:** $totalTP \geq 0$ total TCP throughput is greater than 0
**Require:** $preAvgTP \geq 0$ previous average TCP throughput is greater than 0
**Require:** $maxAvgTP \geq 0$ maximum average TCP throughput is greater than 0
**Require:** $now \leftarrow Time_{Current} > 0$ current time throughput is greater than 0
**Require:** $incrCW$ is *boolean* indicate to increase $CWMin$

```
1: while running do
2:     avgTP ← totalTP/now
3:     if avgTP < maxAvgTP then
4:         if avgTP < preAvgTP then
5:             if incrCW then
6:                 incrCW ← false
7:             else
8:                 incrCW ← true
9:             end if
10:        end if
11:    else
12:        maxAvgTP ← avgTP
13:    end if
14:    if incrCW then
15:        Increase CWMin
16:        CWMax ← 2 × CWMin
17:    else
18:        Decrease CWMin
19:        CWMax ← 2 × CWMin
20:    end if
21:    preAvgTP ← avgTP
22: end while
```

to the one-hop scenario. This estimation is actually the lower bound of $CWMin$ that optimized the TCP throughput.

$$E[RTT] = m(E[T_{data}] + E[T_{ACK}]) \tag{6.44}$$

### 6.2.6 simulations

In order to verify our algorithm, simulations are conducted and the simulation results are discussed in this section. We simulate our algorithm on the NS-2 simulator [75] by modifying the built-in TCP protocols. We test our protocol at VOIP G.711 CODEC with a segment size of 160 bytes and a 20ms sending interval. In our simulation, the link layer is IEEE802.11. The simulation is achieved in the following scenarios: one-hop topology, chain topology, cross topology, and grid topology.

We will also test our protocol on two different protocols: the TCP Reno and the TCP Vegas, in order to verify our theory. Compared to the TCP Reno, the TCP Vegas is a proactive implementation of TCP; this is because it detects congestion before it has fully developed, while the TCP Reno detects the congestion once it has actually happened. We try to illustrate in this section that both implementations can be improved based on our theory. Our protocol operates in the MAC layer, so it will not affect any TCP protocol on the transport layer. Regardless of which TCP implementation is being used, our protocol can improve its performance.

#### One hop Scenario

In the one-hop scenario, the receiver works as a coordinator for the network, and it can probe nearby terminals and assign the corresponding lower bound of contention windows to nearby terminals in order to infer the corresponding throughput. The receiver will accept TCP transmissions from multiple sources. The criteria measured in our simulation is TCP throughput. We compare the performance of our algorithm to the performance of normal TCP Reno in wireless networks. The results indicate that our algorithm significantly improves the performance of TCP, which proves our theory that TCP throughput is affected by contention over wireless networks. Because the contention in wireless networks is more severe than that in the wired networks, its impact on TCP throughput has been enlarged. Limited by the overall network conditions, the TCP throughput of each terminal decreases by the increment of the network size. However, the decrement is not linear because the network capacity is not fully occupied. More

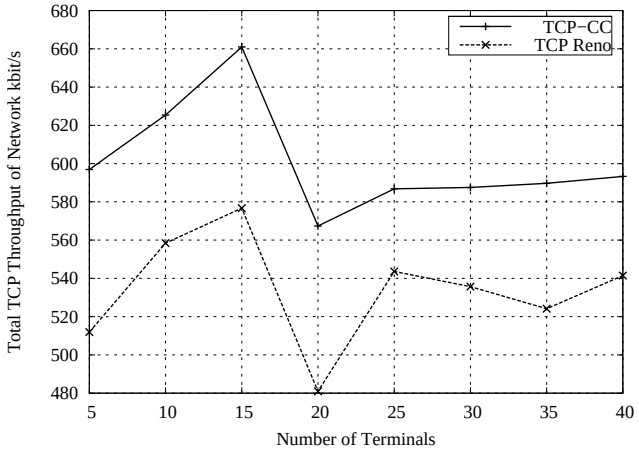



Figure 6.21: TCP Reno Throughput under One Hop Scenario in Different Sizes

segments added into the network increase the network utilization. Our algorithm can achieve an average 10% additional overall TCP throughput than TCP Reno, which is indicated in Figure 6.21.

In order to verify our algorithm, we test our algorithm in TCP Vegas as well. It shows similar results in Figure 6.22. When the size of the network is small, for example, 5 hops, TCP Vegas sufficiently uses the network resources. After the network resources are sufficiently used, the increasing network size degrades the performance of TCP Vegas. Our algorithm improves the performance of TCP Vegas in both a small network size and a large network size. In a small network size, our cross-layer algorithm shortens the time for deferring and backoff; and the network resources are sufficiently occupied. When the network size increases, the degradation of TCP Vegas is absorbed by the contention control of our algorithm; degradation is controlled in 3% of the simulation.

Even though TCP Vegas utilizes more resources than TCP Reno, the oscillation of TCP Vegas is more significant than that of TCP Reno, because of its sensitive congestion control. Our algorithm does not only improve the TCP throughput, but also slows down the performance degradation of TCP Vegas when the network size increases.

### Multi-hop Scenarios

In multi-hop scenarios, we compare our TCP-CC algorithm with the TCP Reno to the normal TCP Reno protocol in chain topology, cross topology, and grid topology within multi-hop scenarios. We run the simulation in the chain topology from 5 hops to 30 hops, in cross topology from 5 hops to 31 hops, and in grid topology from $3 \times 3$ to

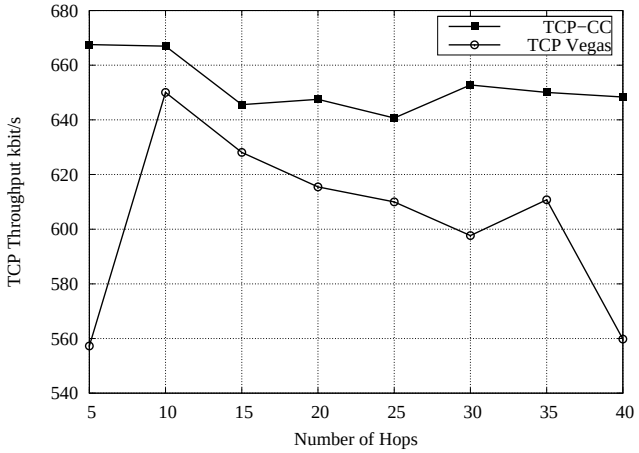
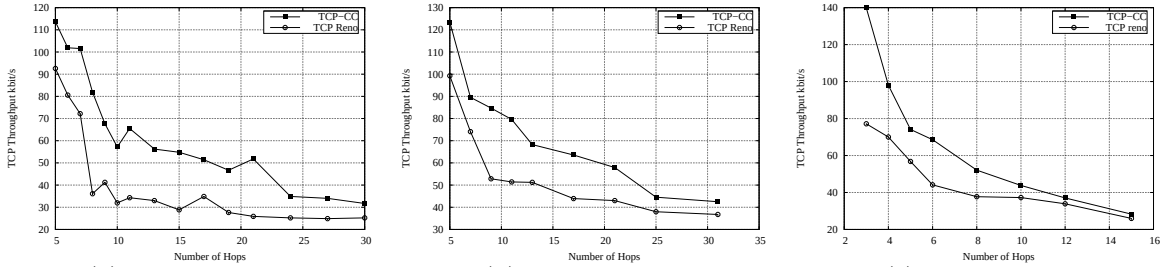


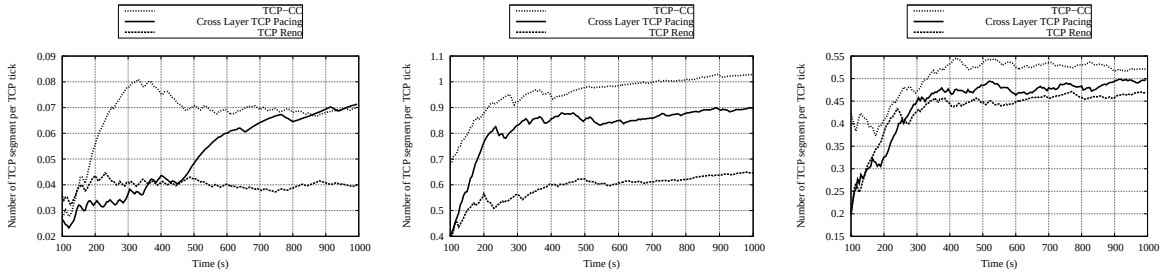

Figure 6.22: TCP Vegas Throughput under One Hop Scenario in Different Sizes




(a) chain topology (b) cross topology (c) grid topology

Figure 6.23: Overall TCP Reno Throughput under Multi-hop Scenarios

$15 \times 15$. We measure the overall TCP throughput of the network through the successfully received TCP segments, which are the accumulation of the throughput of all TCP sources. Figure 6.23(a) indicates that our protocol can achieve a higher TCP throughput in chain topology than the normal TCP Reno, even when the number of hops increases. Our protocol is able to pace the TCP traffic flow by inserting a delay interval into the MAC traffic flow. Therefore, the physical traffic flow in the network is eventually well paced. The paced traffic flow leads to a burstiness-free network with less overall contention. Cross topology Figure 6.23(b) and grid topology Figure 6.23(c) give the same positive results as chain topology. In grid topology, the improvement of our algorithm decreases. The reason is that the total throughput of TCP Reno is actually very low while the network size gets too large; also, TCP Reno has a high tolerance level for high contention networks. As a reactive implementation of TCP, TCP Reno will not easily treat TCP segment loss, caused by contention, as congestion.



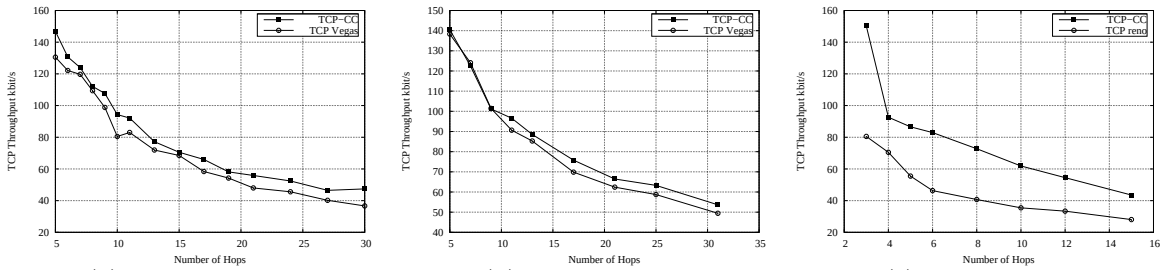

(a) chain topology (b) cross topology (c) grid topology

Figure 6.24: Speed of the improvement process of TCP Reno in Multi-hop Scenario

We also run simulations to measure the efficiency of the protocols. We run the simulation on a 10-hops chain topology, 10-hops cross topology, and $10 \times 10$ grid topology. Figure 6.24 illustrates the efficiency of the cross-layer pacing technique in different topology scenarios, both with and without the initial lower bound of contention window $CWMin$. Figure 6.24(a) illustrates the efficiency of a chain topology with 10-hops with different protocols. Initial $CWMin$ is able to optimize the TCP throughput within a short period of time, because the initial value of the $CWMin$ is very close to the value that maximize the throughput. As seen in Figure 6.24(b), we test the cross topology in a 10 hops scenario, and see that the TCP throughput increases very quickly and remains stable with initial $CWMin$. Grid topology Figure 6.24(c) demonstrates the same results. With initial $CWMin$, our cross-layer TCP pacing protocol is much closer to the optimization point and reduces the oscillation of the contention window size.

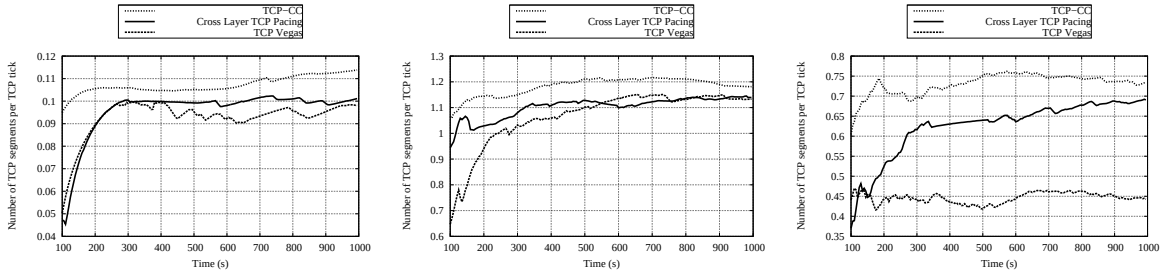
We also compare TCP-CC implemented by TCP Vegas to the normal TCP Vegas in different topology, as shown in Figure 6.25. Our protocols achieve improvements of up to 22% in chain topology, up to 7% in cross topology, and up to 44% in grid topology compared to that of the normal TCP Vegas. In chain topology and cross topology, the contention level is not as high as in grid topology. The available improvement is limited. Because TCP Vegas is a proactive implementation, a high contention levels will cause it to degrade. It considers the TCP segment loss as being caused by congestion, but it is actually caused by contention. It leads to the TCP Vegas unnecessarily decreasing congestion windows. Our protocol reduces the contention level in MAC which eventually avoids the situation described, and it leads to a significant improvement in the performance of TCP Vegas in grid topology, which is very similar to the normal network topology.

To further illustrate the performance of our protocol, we show the runtime detail of




(a) chain topology (b) cross topology (c) grid topology

Figure 6.25: Overall TCP Vegas Throughput under Multi-hop Scenarios




(a) chain topology (b) cross topology (c) grid topology

Figure 6.26: Speed of the improvement process of TCP Vegas in Multi-hop Scenario

the three protocols: the TCP Vegas, our TCP pacing algorithm and TCP-CC. Results are shown in Figure 6.26 with 10-hops chain topology, 10-hops cross topology and $10 \times 10$ grid topology. The TCP-CC with the initial $CWMin$ achieves a very high throughput at the beginning of a simulation, because it avoids the first burstness caused by the first set of TCP segments pouring into the networks. In contrast, our TCP pacing algorithm and TCP Vegas suffer a degradation at the beginning of the simulation. As we demonstrated in the above paragraph, in the chain topology and the cross topology, the available space for improvement is limited. The throughput of our TCP pacing algorithm is very similar to that of TCP Vegas. In the grid topology, because of the high contention level, our algorithm retrieves a significant improvement in terms of TCP throughput than that of TCP Vegas.

To verify our algorithm, we tested it in both TCP Reno and TCP Vegas. Results show that our algorithm can achieve an improvement in both protocols. However, there is a slight difference: our TCP pacing can achieve significant improvement on chain topology and cross topology in TCP Reno; however, in contrast, it can achieve significant improvements in grid topology on TCP Vegas. This difference is caused by the proactive

and reactive characteristics of each protocol.

### 6.2.7 Summary

Most of the studies on TCP optimization on wireless networks are focus on the modification of TCP protocol. However, the drawback of TCP within wireless networks is mainly caused by unstable network conditions, as well as the fact that lower layer parameters are totally hidden from TCP. One of the optimized solutions to this problem is utilizing a cross-layer technique. In this paper, we measured and modeled how the contention window size in the MAC layer affects the TCP throughput. Based on the simulation results, we propose a TCP-CC protocol which achieves both one-hop improvement and a cross-layer pacing in multi-hop TCP transmissions. Our solution aims to modify the contention window mechanism to improve TCP performance without any changes to the TCP protocol. The simulation results illustrate that our protocol is able to optimize the TCP throughput on both TCP Reno and TCP Vegas. The initial lower bound of the contention window size $CWMin$ is able to increase and enhance the process of improvement.

# Chapter 7

# Conclusion and Future Work

In this dissertation, the problems of the video sharing on VANET are addressed. In the following, the contributions of this research and our future work are presented.

## 7.1 Our Contributions

Our contributions include three parts.

**Multi-path Error Recovery Video Streaming (MERVS) on VANETs**

A Multi-path Error Recovery Video Streaming (MERVS) technique is proposed. MERVS splits a video stream into primary and secondary streams. Primary stream is only composed of I-frames, and secondary stream is composed of the other frames. The primary stream is transmitted over the TCP protocol, in order to ensure the transmission of the I-frames, which is the significant part of a video stream. The secondary stream is transmitted over UDP, which is not guaranteed but efficient enough. Several delay improvement techniques are integrated to improve the performance of the primary stream, in order to improve the performance of MERVS. Link disjoint and node disjoint is also applied to MERVS, in order to reduce the effect of the interference and collision.

**Cross Layer Routing Scheme**

A transmission path, with less estimated transmissions, is able to improve the quality of a transmitted video. Because there are two different protocols are selected to transmit the video stream simultaneously, different improvement approaches are proposed. One

is the cross layer delay sensitive wireless routing metric for UDP channel, which can select the lowest delay for UDP transmission. Another one is the TCP-ETX, which is a cross layer metric for TCP improvement. TCP-ETX calculate the expected number of transmissions to successfully deliver a TCP segment, in order to evaluate the quality of a path to the TCP transmissions.

**MAC Layer Improvement**

Because of the selection of the IEEE802.11, the mechanism of the contention window in IEEE802.11 is able to be improved to achieve higher performance of the streaming. Most recent QoS techniques for ad hoc networks rely on basic QoS classifications such as Enhanced Distributed Channel Access (EDCA) and Hybrid Coordination Function Channel Access (HCCA) with pre-fixed contention window range for specific services, which without considering overall resource consumption and the external correlation among transmitting source nodes. A cross layer QoS insurance technique is proposed to ensure the UDP transmissions, which applies dynamic and automatic QoS assignments with low complexity by considering the restrictions among various requirements simultaneously. TCP suffers significant degradation over wireless networks. By analyzing the relation of TCP throughput and the contention window size, A TCP-CC is proposed for TCP throughput improvement, which is a cross layer TCP pacing protocol by contention control.

## 7.2 Future Work

In the future, we plan to extend our research based on the following aspects.

### 7.2.1 Optimization of Guarantee Channel

The handshake feature of TCP causes extra delay on the connection and reconnection of the TCP communication. There are several existing techniques to offer TCP fast open for shorten the time to establish the TCP connections[86]. The detection of the disconnection is also need to be improved in TCP for fast reconnection of the broken connection to the source. A multi-source solution also can be proposed to avoid this problem. The reliability level of the TCP transmission can also be adjusted based on the quality of the channel in order to maintain the TCP connection and reduce the time to wait for a TCP failure.

### 7.2.2 Zone Disjoint

The major cause of the collision and interference is the routing coupling on the two traffics. In order to prevent the coupling of the two routes, the cause of the collision and interference should be carefully considered. Communication interference is mainly caused by the signal disruption of two close signal sources. Therefore, when choosing the route for TCP and UDP transmission in MERVS, we should consider the interference range of the signal to avoid the interference. The node disjoint and link disjoint algorithm is able to avoid the collision and one hop interference. However, the existing two hop interference still need to be solved in the future. Zone disjoint is an approach to solve the problems of interference, but because of the limited width of a paved road zone disjoint algorithm is difficult split the area of a paved road in VANET.

### 7.2.3 Localization and Quality Assurance Routing

When achieving route coupling prevention, the route should be carefully chosen as well. Most of the relaying algorithm for video streaming in VANETs is based on the localization information [20, 22, 33, 21, 67], because the existing of GPS and other location information devices [26]. Localization routing protocols help the service to discover a shortest route faster, comparing to the traditional routing protocols with broadcasting routing advertisements. However, most of the localization routing does not consider the quality of the link, because they assume that the link quality of all nodes are the same. This assumption is not always correct in the realistic situation, because of the various quality and workload of the devices. We would like to do research on integration of the localization and quality assurance routing for video streaming in VANETs in the future work. Some researches on quality insurance video streaming have already published, like [59, 73].

### 7.2.4 Security

Because wireless network is an open network, the security issue should be carefully considered. Trust management system is require in VANETs to insure the information is correct and harmless [9, 25]. Malicious activities or policy violations should be monitored in vehicles or in the RSU [17, 19, 18], to insure the networks is secured. Anonymous techniques is also another way to insure the security of the networks, such as [14, 16, 15]

### 7.2.5 Video Streaming on Sensor Networks

Besides VANETs, our algorithm should be able to apply to wireless sensor networks. The main problems of sensor networks are less processing power and short battery life. A light weight energy-aware routing protocol are required to apply our protocols in wireless sensor networks, such as [12, 107, 13, 70]. TCP might not be suitable for sensor networks, because of the heavy processing and battery consumption. Other techniques might be required to insure the video transmissions, such as [24]. Fault tolerance techniques should also be integrated, such as [23, 35, 3].

# Bibliography


[1] Boukerche A. *Handbook of Algorithms for Wireless Networking and Mobile Computing*. CRC press, Boca Raton, United States, 2005.

[2] A. Aggarwal, S. Savage, and T. Anderson. Understanding the performance of TCP pacing. In *Proceedings IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1157 –1165 vol.3, Mar 2000.

[3] T. Antoniou, I. Chatzigiannakis, G. Mylonas, S. Nikoletseas, and A. Boukerche. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. In *Proceedings of the 37th annual symposium on Simulation*, page 43. IEEE Computer Society, 2004.

[4] M. Asefi, J. W. Mark, and X. Shen. A mobility-aware and quality-driven retransmission limit adaptation scheme for video streaming over VANETs. *IEEE Transactions on Wireless Communications*, 11(5):1817–1827, 2012.

[5] M. Barni. *Document and Image Compression*. Signal Processing and Communications. Taylor & Francis, 2006.

[6] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 63–68, Barcelona, Spain, October 2011.

[7] M. Bharatiya and S. Prasad. MILD based sliding contention window mechanism for QoS in wireless LANs. In *Proc. IEEE International Workshops on Wireless Communication and Sensor Networks(WCSN'08)*, pages 105 – 110, 2008.

[8] G. Bhutani. A near-optimal scheme for TCP ACK pacing to maintain throughput in wireless networks. In *2010 Second International Conference on Communication Systems and Networks (COMSNETS)*, pages 1 –7, Jan. 2010.

[9] A. Boukerch, L. Xu, and K. EL-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, 30:2413–2427, 2007.

[10] A Boukerche. *Algorithms and protocols for wireless, mobile Ad Hoc networks*, volume 77. John Wiley & Sons, 2008.

[11] A. Boukerche. *Algorithms and protocols for wireless sensor networks*, volume 62. John Wiley & Sons, 2008.

[12] A. Boukerche, X. Cheng, and J. Linus. Energy-aware data-centric routing in microsensor networks. In *Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 42–49. ACM, 2003.

[13] A. Boukerche, X. Cheng, and J. Linus. A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, 11(5):619–635, 2005.

[14] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. A novel solution for achieving anonymity in wireless ad hoc networks. In *Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 30–38. ACM, 2004.

[15] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 618–624. IEEE, 2004.

[16] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *computer communications*, 28(10):1193–1203, 2005.

[17] A. Boukerche, K. RL Jucá, J. B. Sobral, and M. SMA Notare. An artificial immune based intrusion detection model for computer and telecommunication systems. *Parallel Computing*, 30(5):629–646, 2004.

[18] A. Boukerche, R. B Machado, K. RL Jucá, J. B. M Sobral, and M. SMA Notare. An agent based and biological inspired real-time intrusion detection and security model for computer network operations. *Computer Communications*, 30(13):2649–2660, 2007.

[19] A. Boukerche and M SMA Notare. Behavior-based intrusion detection in mobile phone systems. *Journal of Parallel and Distributed Computing*, 62(9):1476–1490, 2002.

[20] A. Boukerche, H. ABF Oliveira, E. F Nakamura, and A. AF Loureiro. Localization systems for wireless sensor networks. *wireless Communications, IEEE*, 14(6):6–12, 2007.

[21] A. Boukerche, H. ABF Oliveira, E. F Nakamura, and A. AF Loureiro. Secure localization algorithms for wireless sensor networks. *Communications Magazine, IEEE*, 46(4):96–101, 2008.

[22] A. Boukerche, H. ABF Oliveira, E. F Nakamura, and A. AF Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications*, 31(12):2838–2849, 2008.

[23] A. Boukerche, R. WN Pazzi, and R. Araujo. Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments. *Journal of Parallel and Distributed Computing*, 66(4):586–599, 2006.

[24] A. Boukerche, R. WN Pazzi, and R. B. Araujo. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 157–164. ACM, 2004.

[25] A. Boukerche and Y. Ren. A trust-based security system for ubiquitous and pervasive computing environments. *Computer Communications*, 31(18):4343–4351, 2008.

[26] A. Boukerche and S. Rogers. Gps query optimization in mobile and wireless networks. In *Computers and Communications, 2001. Proceedings. Sixth IEEE Symposium on*, pages 198–203. IEEE, 2001.

[27] A. Boukerche, B. Turgut, N. Aydin, M. Z Ahmad, L. Bölöni, and D. Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032–3080, 2011.

[28] R. Braden, T. Faber, and M. Handley. From protocol stack to protocol heap - role-based architecture. In *In 1st ACM Workshop on Hot Topics in Networks*, pages 17–22, 2002.

[29] P. Bucciol, J.-L. Zechinelli-Martini, and G. Vargas-Solar. Optimized transmission of loss tolerant information streams for real-time vehicle-to-vehicle communications. In *2009 Mexican International Conference on Computer Science (ENC)*, pages 142–145, 2009.

[30] D. Chen, H. Ji, and V.C.M. Leung. Distributed best-relay selection for improving TCP performance over cognitive radio networks: A cross-layer design approach. *IEEE Journal on Selected Areas in Communications*, 30:315–322, February 2012.

[31] Y. Chetoui and N. Bouabdallah. Adjustment mechanism for the IEEE 802.11 contention window: An efficient bandwidth sharing scheme. *Computer Communications*, 30:2686–2695, September 2007.

[32] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom '03, pages 134–146, New York, NY, USA, 2003. ACM.

[33] H. ABF De Oliveira, A. Boukerche, E. F. Nakamura, and A. AF Loureiro. An efficient directed localization recursion protocol for wireless sensor networks. *Computers, IEEE Transactions on*, 58(5):677–691, 2009.

[34] J. Dong, R. Curtmola, and C. Nita-Rotaru. Secure high-throughput multicast routing in wireless mesh networks. *IEEE Transactions on Mobile Computing*, 10(5):653–668, May 2011.

[35] M. Elhadef, A. Boukerche, and H. Elkadiki. A distributed fault identification protocol for wireless and mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 68(3):321–335, 2008.

[36] S.M. ElRakabawy and C. Lindemann. A practical adaptive pacing scheme for TCP in multihop wireless networks. *IEEE/ACM Transactions on Networking*, 19(4):975 –988, Aug 2011.

[37] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – http/1.1, 1999.

[38] S. Floyd, M. Allman, A. Jain, and P. Sarolahti. Quick-Start for TCP and IP. RFC 4782 (Experimental), Jan 2007.

[39] S. Fortin-Parisi and B. Sericola. A markov model of TCP throughput, goodput and slow start. *Perform. Eval.*, 58(2+3):89–108, November 2004.

[40] F. Foukalas, V. Gazis, and N. Alonistioti. Cross-layer design proposals for wireless mobile networks: a survey and taxonomy. *Communications Surveys Tutorials, IEEE*, 10(1):70–85, 2008.

[41] Z. Fu, H. Luo, P. Zerfos, S. Lu, and L. Zhang. Mario gerla: The impact of multihop wireless channel on TCP performance. *IEEE Transactions on Mobile Computing*, 4:209–221, 205.

[42] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla. The impact of multihop wireless channel on TCP throughput and loss. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1744 – 1753 vol.3, Mar 2003.

[43] D. Guo and X. Wang. Bayesian inference of network loss and delay characteristics with applications to TCP performance prediction. *IEEE Transactions on Signal Processing*, 51:2205 – 2218, August 2003.

[44] S. Hua, Y. Guo, Y. Liu, H. Liu, and S.S. Panwar. Scalable video multicast in hybrid 3G/ad-hoc networks. *IEEE Trans on Multimedia*, 13(2):402–413, 2011.

[45] P. Huang, C. Wang, and L. Xiao. Improving end-to-end routing performance of greedy forwarding in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(3):556 –563, Mar 2012.

[46] G. Jakllari, S. Eidenbenz, N. Hengartner, S.V. Krishnamurthy, and M. Faloutsos. Link positions matter: A noncommutative routing metric for wireless mesh networks. *IEEE Transactions on Mobile Computing*, 11(1):61 –72, Jan. 2012.

[47] N. Javaid, A. Bibi, and K. Djouani. Interference and bandwidth adjusted ETX in wireless multi-hop networks. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pages 1638 –1643, Dec. 2010.

[48] S. Jung, J. Sung, Y. Bang, M. Kserawi, H. Kim, and J.-K.K. Rhee. Greedy local routing strategy for autonomous global load balancing based on three-dimensional potential field. *Communications Letters, IEEE*, 14(9):839 –841, Sep 2010.

[49] T. Kim, S. Kim, J. Yang, S. Yoo, and D. Kim. Neighbor table based shortcut tree routing in zigbee wireless networks. *Paral and Dist Systems, IEEE Trans*, 25(3):706–716, 2014.

[50] D. Koutsonikolas, Chih-Chun Wang, and Y.C. Hu. Efficient network-coding-based opportunistic routing through cumulative coded acknowledgments. *IEEE/ACM Transactions on Networking*, 19(5):1368 –1381, Oct. 2011.

[51] S. Krco and M. Dupcinov. Improved neighbor detection algorithm for AODV routing protocol. *Communications Letters, IEEE*, 7(12):584–586, 2003.

[52] M. Krishnan and A. Zakhor. Throughput improvement in 802.11 WLANs using collision probability estimates in link adaptation. In *Proc. IEEE International Workshops on Wireless Communications and Networking (WCNC'10)*, pages 1 – 6, 2010.

[53] A. Ksentini, A. Nafaa, A. Gueroui, and M. Naimi. Determinist contention window algorithm for IEEE 802.11. In *Proc. IEEE International conference on Personal, Indoor and Mobile Radio Communications (PIMRC'05)*, pages 2712 – 2716, 2005.

[54] M. Kserawi, S. Jung, D. Lee, J. Sung, and J.-K.K. Rhee. Multipath video real-time streaming by field-based anycast routing. *IEEE Transactions on Multimedia*, 16(2):533–540, Feb 2014.

[55] W.-H. Kuo, W. Liao, and T. Liu. Adaptive resource allocation for layer-encoded IPTV multicasting in IEEE 802.16 wimax wireless networks. *IEEE Trans on Multimedia*, 13(1):116–124, 2011.

[56] S.-B. Lee, G.-S. Ahn, and A.T. Campbell. Improving UDP and TCP performance in mobile ad hoc networks with INSIGNIA. *Communications Magazine, IEEE*, 39(6):156 –165, Jun 2001.

[57] C. Li, J. Zou, H. Xiong, and C. W. Chen. Joint coding/routing optimization for distributed video sources in wireless visual sensor networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(2):141–155, Feb 2011.

[58] X. Li and J. Cai. Robust transmission of JPEG2000 encoded images over packet loss channels. In *2007 IEEE International Conference on Multimedia and Expo*, pages 947–950, 2007.

[59] Y. Li and A. Boukerche. Qugu: A quality guaranteed video dissemination protocol over urban vehicular ad hoc networks. *ACM Trans. Multimedia Comput. Commun. Appl.*, 11(4):55:1–55:23, June 2015.

[60] C.-H. Lin, C.-K. Shieh, and W.-S. Hwang. An access point-based FEC mechanism for video transmission over wireless LANs. *IEEE Transactions on Multimedia*, 15(1):195–206, 2013.

[61] J. D. S. Lorente. *Recent Advances on Video Coding*. InTech, 2011.

[62] C. Luo, F.R. Yu, H. Ji, and V.C.M. Leung. Cross-layer design for TCP performance improvement in cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 59:2485–2495, January 2010.

[63] C.-Y. Luo, N. Komuro, K. Takahashi, and T. Tsuboi. Paced TCP: A dynamic bandwidth probe TCP with pacing in AD HOC networks. In *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007.*, pages 1 –5, Sep 2007.

[64] H. R. Maamar, G. Romn-Alonso, A. Boukerche, and E. M. Petriu. Energy management control for supplying partner selection protocol in mobile peer-to-peer three-dimensional streaming. *Concurrency and Computation: Practice and Experience*, 25(5):752–769, 2013.

[65] H.R. Maamar, A. Boukerche, and E. Petriu. Streaming 3d meshes over thin mobile devices. *Wireless Communications, IEEE*, 20(3):136–142, June 2013.

[66] H.R. Maamar, A. Boukerche, and E.M. Petriu. 3-d streaming supplying partner protocols for mobile collaborative exergaming for health. *Information Technology in Biomedicine, IEEE Transactions on*, 16(6):1079–1095, Nov 2012.

[67] G. Maia, L. Villas, A. Carneiro Viana, A. L. L. Aquino, A. Boukerche, and A. A. F. Loureiro. A rate control video dissemination solution for extremely dynamic vehicular ad hoc networks. *Perform. Eval.*, 87(C):3–18, May 2015.

[68] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, L. Stibor, C. Contention, and F. Poll. IEEE 802.11e wireless LAN for quality of service, 2003.

[69] G. Marfia and M. Roccetti. TCP at last: reconsidering TCP's role for wireless entertainment centers at home. *IEEE Transactions on Consumer Electronics*, 56:2233–2240, November 2010.

[70] A. Martirosyan, A. Boukerche, and R. WN Pazzi. A taxonomy of cluster-based routing protocols for wireless sensor networks. In *Parallel Architectures, Algorithms, and Networks, 2008. I-SPAN 2008. International Symposium on*, pages 247–253. IEEE, 2008.

[71] Y. Matsushita, T. Matsuda, and M. Yamamoto. TCP congestion control with ACK-pacing for vertical handover. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1497 – 1502 Vol. 3, Mar 2005.

[72] M. Mishra and A. Sahoo. A contention window based differentiation mechanism for providing QoS in wireless LANs. In *Proc. IEEE International conference on Information Technology (ICIT'06)*, pages 72 – 76, 2006.

[73] F. Naeimipoor and A. Boukerche. A hybrid video dissemination protocol for vanets. In *Communications (ICC), 2014 IEEE International Conference on*, pages 112–117, June 2014.

[74] K. Nahm, A. Helmy, and C.-C.J. Kuo. Cross-layer interaction of TCP and ad hoc routing protocols in multihop IEEE 802.11 networks. *IEEE Transactions on Mobile Computing*, 7(4):458 –469, Aprl 2008.

[75] NS2. The network simulator ns-2, 2007.

[76] M.Z. Oo and M. Othman. How good delayed acknowledgement effects rate-based pacing TCP over multi-hop wireless network. In *2009 International Conference on Signal Processing Systems*, pages 464 –468, May 2009.

[77] O. Oyman and S. Singh. Quality of experience for HTTP adaptive streaming services. *Comms Mag, IEEE*, 50(4):20–27, 2012.

[78] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose. Modeling TCP reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8:133–145, 2000.

[79] E. Park, D. Kim, H. Kim, and C. Choi. A cross-layer approach for per-station fairness in TCP over WLANs. *IEEE Transactions on Mobile Computing*, 59:898–911, July 2008.

[80] R. WN Pazzi and A. Boukerche. PROPANE: A progressive panorama streaming protocol to support interactive 3D virtual environment exploration on graphics-constrained devices. *TOMCCAP*, 11(1):5:1–5:22, 2014.

[81] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing, 2003.

[82] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, SIGCOMM '94, pages 234–244, New York, NY, USA, 1994. ACM.

[83] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, October 1994.

[84] J. Postel. User datagram protocol. RFC 768, Internet Engineering Task Force, August 1980.

[85] J. Postel. Transmission control protocol. RFC 793, Internet Engineering Task Force, Sep 1981.

[86] N. Qadri, M. Altaf, M. Fleury, M. Ghanbari, and H. Sammak. Robust video streaming over an urban VANET. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009.*, pages 429–434, 2009.

[87] F. Ren and C. Lin. Modeling and improving TCP performance over cellular link with variable bandwidth. *IEEE Transactions on Mobile Computing*, 10(8):1057–1070, 2011.

[88] F. Ren and C. Lin. Modeling and improving TCP performance over cellular link with variable bandwidth. *IEEE Transactions on Mobile Computing*, 10:1057–1070, August 2011.

[89] C. Rezende, A. Mammeri, A. Boukerche, and A. A. F. Loureiro. A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection. *Ad Hoc Netw.*, 17:1–17, June 2014.

[90] C. Rezende, H.S. Ramos, R.W. Pazzi, A. Boukerche, A.C. Frery, and A. A F Loureiro. VIRTUS: A resilient location-aware video unicast scheme for vehicular networks. In *2012 IEEE International Conference on Communications (ICC)*, pages 698–702, 2012.

[91] C. G. Rezende, A. Boukerche, M. Almulla, and A. A. F. Loureiro. The selective use of redundancy for video streaming over vehicular ad hoc networks. *Computer Networks*, 81:43–62, 2015.

[92] C. G. Rezende, A. Boukerche, H. S. Ramos, and A. A. F. Loureiro. A reactive and scalable unicast solution for video streaming over vanets. *IEEE Trans. Computers*, 64(3):614–626, 2015.

[93] C. (Babis) Samios and M. K. Vernon. Modeling the throughput of tcp vegas. *SIGMETRICS Perform. Eval. Rev.*, 31(1):71–81, June 2003.

[94] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications, 2003.

[95] V. Sharma, K. Kar, K.K. Ramakrishnan, and S. Kalyanaraman. A transport protocol to exploit multipath diversity in wireless networks. *IEEE/ACM Transactions on Networking*, 20(4):1024–1039, Aug 2012.

[96] K. Shi, Y. Shu, O. Yang, J. Wang, and J. Luo. Improving TCP performance for EAST experimental data in the wireless LANs. *IEEE Transactions on Nuclear Science*, 58(4):1825–1832, 2011.

[97] K. Shi, Y. Shu, O. Yang, J. Wang, and J. Luo. Improving TCP performance for EAST experimental data in the wireless LANs. *IEEE Transactions on Nuclear Science*, 58:1825–1832, August 2011.

[98] S. Sinha. A TCP tutorial, 1998.

[99] F. Soldo, C. Casetti, C. Chiasserini, and P.A. Chaparro. Video streaming distribution in VANETs. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1085–1091, 2011.

[100] W. Song and W. Zhuang. Performance analysis of probabilistic multipath transmission of video streaming traffic over multi-radio wireless devices. *IEEE Transactions on Wireless Communications*, 11(4):1554–1564, Apr 2012.

[101] V. Srivastava and M. Motani. Cross-layer design: a survey and the road ahead. *Communications Magazine, IEEE*, 43(12):112–119, 2005.

[102] N. Thomos, N.V. Boulgouris, and M.G. Strintzis. Optimized transmission of JPEG2000 streams over wireless channels. *IEEE Transactions on Image Processing*, 15(1):54–67, 2006.

[103] K. Thornberry. Patent application title: Separate video file for I-frame and non-I-frame data to improve disk performance in trick play, 11 2013.

[104] M.-F. Tsai, C.-K. Shieh, T.-C. Huang, and D.-J. Deng. Forward-looking forward error correction mechanism for video streaming over wireless networks. *Systems Journal, IEEE*, 5(4):460–473, 2011.

[105] MF. Tsai, N. Chilamkurti, J Park, and C. Shieh. Multi-path transmission control scheme combining bandwidth aggregation and packet scheduling for real-time streaming in multi-path environment. *Comms, IET*, 4(8):937–945, 2010.

[106] J. Vella and S. Zammit. A survey of multicasting over wireless access networks. *Communications Surveys Tutorials, IEEE*, 15(2):718–753, 2013.

[107] L. Villas, A. Boukerche, H. S Ramos, H. ABF de Oliveira, R. B. de Araujo, A. Loureiro, et al. Drina: a lightweight and reliable routing approach for in-network aggregation in wireless sensor networks. *Computers, IEEE Transactions on*, 62(4):676–689, 2013.

[108] T. Volkert, M. Osdoba, A. Mitschele-Thiel, and M. Becke. Multipath video streaming based on hierarchical routing management. In *2013 27th Int'l Conf on Advanced Information Networking and Applications Workshops*, pages 1107–1112, 2013.

[109] H. Vu and T Sakurai. Collision probability in saturated IEEE 802.11 networks. In *Australian Telecommunication Networks & Applications Conference (ATNAC'06)*, pages 1 – 5, 2006.

[110] C. Wang, B. Li, and L. Li. A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. *IEEE Transactions on Vehicular Technology*, 53:1235, July 2004.

[111] C. Wang, T. Lin, and J. Chen. A cross-layer adaptive algorithm for multimedia QoS fairness in WLAN environments using neural networks. *Communications*, 1:858 – 865, November 2007.

[112] R. Wang, M. Almulla, C. Rezende, and A. Boukerche. Video streaming over vehicular networks by a multiple path solution with error correction. In *Communications (ICC), 2014 IEEE International Conference on*, pages 580–585, June 2014.

[113] R. Wang, C. Rezende, H.S. Ramos, R.W. Pazzi, A. Boukerche, and A. A F Loureiro. LIAITHON: A location-aware multipath video streaming scheme for urban vehicular networks. In *2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 000436–000441, 2012.

[114] W. Wang, X. Liu, and D. Krishnaswamy. Robust routing and scheduling in wireless mesh networks under dynamic traffic conditions. *IEEE Transactions on Mobile Computing*, 8(12):1705–1717, 2009.

[115] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication, special issue on Objective video quality metrics*, 19(2):121–132, Feb 2004.

[116] C.-Y. Wee, R. Paramesran, and R. Mukundan. Quality assessment of gaussian blurred images using symmetric geometric moments. In *14th International Conference on Image Analysis and Processing, 2007. ICIAP 2007.*, pages 807–812, Sep 2007.

[117] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[118] H. Wu, M. Yang, and K. Ke. The design of QoS provisioning mechanisms for wireless networks. In *Proc. IEEE International Workshops on Pervasive Computing and Communications (PERCOM'10)*, pages 756 – 759, 2010.

[119] M. Wu, S. Makharia, H. Liu, D. Li, and S. Mathur. IPTV multicast over wireless LAN using merged hybrid ARQ with staggered adaptive FEC. *IEEE Transactions on Broadcasting*, 55(2):363–374, Jun 2009.

[120] H. Xie, A. Boukerche, and A. Loureiro. A multi-path video streaming solution for vehicular networks with link disjoint and node-disjoint. *Parallel and Distributed Systems, IEEE Transactions on*, PP(99):1–1, 2014.

[121] H. Xie, A. Boukerche, and A. Loureiro. Mervs: A novel multi-channel error recovery video streaming scheme for vehicle ad-hoc networks. *Vehicular Technology, IEEE Transactions on*, PP(99):1–1, 2015.

[122] M. Xing and L. Cai. Adaptive video streaming with inter-vehicle relay for highway VANET scenario. In *2012 IEEE International Conference on Communications (ICC)*, pages 5168–5172, 2012.

[123] H. Zhang, L. Sang, and A. Arora. Comparison of data-driven link estimation methods in low-power wireless networks. In *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09.*, pages 1 –9, Jun 2009.

[124] L. Zhang, G. Wang, and W. Dong. Adaptive contention window adjustment for 802.11-based mesh networks. In *Proc. IEEE International conference on Wireless Communication, Networking and Mobile Computing (WiCOM'08)*, pages 1 – 4, 2008.

[125] R. Zhang and M.K. Tsatsanis. Network-assisted diversity multiple access in dispersive channels. *IEEE Transactions on Communications*, 50(4):623–632, 2002.

[126] X. Zhang, W. Zhu, N. Li, and D. Sung. TCP congestion window adaptation through contention detection in ad hoc networks. *IEEE Transactions on Vehicular Technology*, 59:4578–4588, November 2010.

[127] Z. Zhu, S. Li, and X. Chen. Design QoS-aware multi-path provisioning strategies for efficient cloud-assisted SVC video streaming to heterogeneous clients. *IEEE Transactions on Multimedia*, 15(4):758–768, Jun 2013.

[128] H. Zimmermann. OSI reference model–the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432, 1980.

[129] J. Zou, H. Xiong, C. Li, L. Song, Z. He, and T. Chen. Prioritized flow optimization with multi-path and network coding based routing for scalable multirate multicasting. *IEEE Trans on Circuits and Systems for Video Technology*, 21(3):259–273, 2011.