uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Hanliu Chen

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Performance Evaluation and Analysis of Datagram Congestion Control Algorithms in IP Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Prof. O. Yang

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Prof. S. Shirmohammadi

Prof. C.H. Lung

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Performance Evaluation and Analysis of Datagram Congestion Control Algorithms in IP Networks

By

Hanliu Chen

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

Master of Applied Science

Master Program in Electrical Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

May 9, 2008

Your file Votre référence
ISBN: 978-0-494-50863-3
Our file Notre référence
ISBN: 978-0-494-50863-3

Abstract

Providing a TCP-friendly congestion control protocol to support multimedia applications has been a great challenge for many researchers. DCCP (Datagram Congestion Control Protocol) is a promising protocol that was proposed recently. This thesis proposes to use VCP (Variable-structure congestion Control Protocol) to overcome the shortcomings of RED (Random Early Detection). We make mathematical analysis and run OPNET simulation to validate DCCP/VCP congestion control strategy. We have also conducted some performance analysis and made comparison with some other related traffic congestion control strategies under both wired and WiMAX environments. The simulation results show that DCCP/VCP can maintain a shorter buffer queue size than the other congestion control strategies. DCCP/VCP also provides zero packet drop rate which solves the random packet loss problem in RED router.

Acknowledgements

I would like to sincerely thank my supervisor, Dr. Oliver Yang, for his research guidance and suggestions throughout the research. I also wish to extend my thanks to Dr. Yang Hong for his kind help.

I would like to express my deep gratitude to my family, especially my mother Huarong Zang and my father Jijiang Chen, for their endless support and love.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

		Section of 1st appearance
ACK	Acknowledgement	1.2.1.1
ADSL	Asymmetric Digital Subscriber Line	1.2.1.2
AI	Additive Increase	2.4
AQM	Active Queue Management	1.1
ATM	Asynchronous Transfer Mode	1.2.1.2
BS	Base Station	2.5
DCCP	Datagram Congestion Control Protocol	1.2.1
CCID	Congestion Control Identifier	1.2.1.1
CWND	Congestion Window Size	1.2.1.1
DH	Drop Head	1.2.2.1
DT	Drop Tail	1.2.2.1
ECN	Explicit Congestion Notification	1.2.2.1
FDD	Frequency Division Duplex	2.5
FIFO	First in First out	2.3
FSM	Finite State Machine	3.5
HTTP	Hypertext Transfer Protocol	3.3.2
IT	Information Technology	1.1
MAC	Medium Access Control	2.5
MD	Multiplicative Decrease	2.4
MI	Multiplicative Increase	2.4
OFDM	Orthogonal Frequency Division Multiplexing	2.5
OFDMA	Orthogonal Frequency Division Multiplexing Access	2.5
PDA	Personal Digital Assistants	1.1
QAM	Quadrature Amplitude Modulation	2.5
QPSK	Quadrature Phase Shift Keying	2.5
QoS	Quality of Service	1.2
RED	Random Early Detection	1.1
RTPD	Round Trip Propagation Delay	3.1
RTT	Round Trip Time Delay	5.4
SC	Single Carrier	2.5
SS	Subscriber Station	2.5
SSTHRESH	Slow Start Threshold	3.1.2
TCP/IP	Transmission Control Protocol / Internet Protocol	1.1
TDD	Time Division Duplex	2.5
TDMA	Time Division Multiplex Access	2.5
TFRC	TCP Friendly Rate Control	1.2.1.2
UDP	User Datagram Protocol	1.2.1.1
VCP	Variable-structure congestion Control Protocol	1.2.2.1
WiMAX	Worldwide Interoperability for Microwave Access	1.2.3
XCP	Explicit Congestion Control Protocol	1.2.2.1

List of Notations and Symbols

		Section of 1st appearance
b	Number of packets acknowledged by an ACK in DCCP-CCID3	3.3.1
C	Link capacity in bits per second	3.1.0
$cwnd$	Congestion window size	3.2.0
k_q	VCP constant to control how fast to drain the link steady queue	3.2.0
N	Number of DCCP flows	3.5.0
p	Loss event rate in DCCP-CCID3	3.3.1
$pipe$	Number of outstanding packets in the network	3.3.1
q_l	Persistent queue size	3.2.0
q_t	Queue size at time t	4.2.1
R	Round trip time delay	3.3.1
R_p	Propagation delay	4.3.2
R_q	Queuing delay	4.3.2
t_p	Link load factor measurement interval	3.2.0
t_q	Link queue sampling interval	3.2.0
T	Time delay in the destination node to send our an ACK	4.3.2
T_{avg}	Average of the time period T	4.3.2
T_{RTO}	Source retransmission timeout in DCCP-CCID3	3.3.1
s	Packet size	3.3.1
$ssthresh$	Slow Start Threshold	3.3.1
$wi(t)$	Flow i's congestion window size at time t	4.3.3
a	AI parameter	3.2.0
s_{AI}	AI scaling limiter	3.2.0
s_{MI}	MI scaling limiter	3.2.0
β	MD parameter	3.2.0
γ	Link target utilization	3.2.0
λ_l	Incoming traffic size during each t_p period	3.2.0
μ	MI parameter	3.2.0
ρ	Traffic load factor in VCP	2.4.0
$\rho(t)$	Traffic load factor in the router at time t	4.1.0
t_{i1}	Path delay from Source i to the bottleneck router	4.1.0
t_{i2}	Path delay from the bottleneck router back to Source i through the destination	4.1.0
ζ	Acknowledgement ratio	2.2.1

Chapter 1
Introduction

1.1 Overview

Over the past several decades, computer technologies have made spectacular development, due to the rapid advancement on both hardware and software. During the 1970s and 1980s, the technology, products, and companies of the now combined computer-communications industry were profoundly changed by a merger of the fields of computer system and data communications. From the 1990s, computer networks began to deliver Internet services to private individuals at home. Entering the 21st century, wireless communication technology is generally used for mobile IT (Information Technology) equipment, such as cellular telephone, PDA (Personal Digital Assistants) and wireless networking. The boom of internetworking has required high efficient traffic engineering mechanisms to manage the traffic exposed to the network, and is stressing network traffic and resources for congestion control implementation. TCP/IP (Transmission Control Protocol / Internet Protocol), the dominant network protocol suite, plays a very important role in the network traffic control nowadays. TCP Reno algorithm [Jaco90] has been the most-adopted traffic control algorithm in computers. On the other hand, RED (Random Early Detection) [FlJa93], the most popular AQM (Active Queue Management) algorithm, works in the router to cooperate with the TCP algorithm working in the source and destination to ensure that congestion is avoided. TCP/RED is being used to be the dominant networking strategy on traffic congestion avoidance.

In recent years, real-time multimedia communication has become one of the major data streams over the Internet. A high bandwidth network with high utilization, low packet drop rate and small time delay is required. In current TCP/IP networks, TCP relies on packet drops in RED router as the indication of congestion. When people use real-time multimedia transmission such as video conferencing, this feature may cause many packet losses, unstable queue size, large time delay and severely suffered throughput, which limit people to achieve the healthy real-time multimedia communications. To overcome the weakness of these

algorithms, some new AQM and transport layer protocols were developed in recent years. However, they have various limitations which cannot solve all of the performance problems mentioned above. Considering that the available Internet bandwidth to service people cannot be enlarged illimitably and that the service demands are increasing at an exponential rate, the traffic and congestion control mechanism over both wired and wireless links will be a key factor in the Internet service quality, and hence deserves further studies.

1.2 Classification of Congestion Control Mechanisms

Due to the congestion collapse experienced by the Internet, a large number of widely dispersed Internet sites experienced simultaneous slowdown or cessation of networking services for prolonged periods, which made the need for Internet congestion control originally apparent. For the purpose to conduct our research on Internet congestion control, we put an insight into the related literatures in this area. We classify most of the relative proposed algorithms in a logical gradation. We also list the merits and short-comings of each technology.

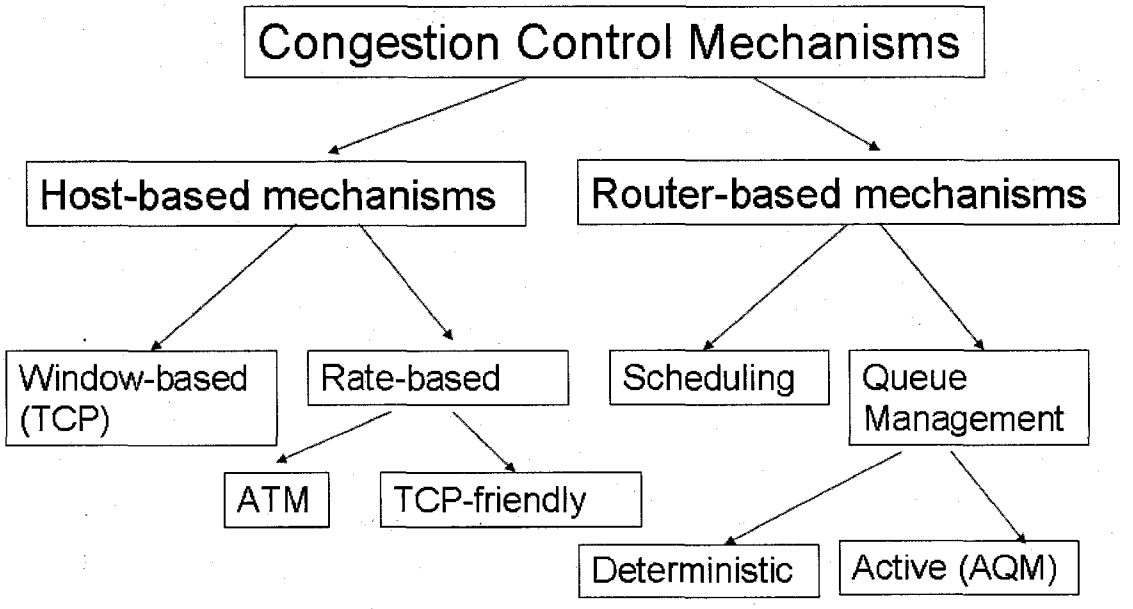


Figure 1-1: Literature Classification of Congestion Control Mechanisms

Figure 1-1 shows the literature classification according to where the algorithm functions and its mechanism. Generally, the major congestion control algorithms which have

been proposed so far regarding the TCP/IP network fall into two main categories. They are host-based mechanisms and router-based mechanisms. The source node controls its window size or sending rate according to the information it gets from the destination node in the host-based control (or end-to-end control) category, e.g. [MaSe97]. In the router-based control mechanisms, the router regulates its queue size to an acceptable level by putting the information of either the router buffer status or the recommended source transmission rate into the packet header. The source then uses this information to adjust its window size to prevent congestion from happening in the router. The co-operation of host-based and router-based mechanisms in network traffic control allows people to achieve high QoS (Quality of Service) requirements.

1.2.1 Host-based Mechanisms

Host-based mechanisms are required to avoid congestion collapse, achieve source sharing fairness, minimize loss and delay, and maximize throughput. The widely adopted TCP and the recently proposed DCCP (Datagram Congestion Control Protocol) [KoHa06] (to be described in Section 2.2) are the typical host-based mechanisms. In the host-based congestion control mechanisms, there are two categories according to the basic congestion control functions of a source node. They are the window-based control and the rate-based control. Window-based control mechanism host adjusts its window size achieve congestion avoidance. On the other hand, rate-based control mechanism host changes its data sending rate according to the information its gets from the destination node to avoid router buffer overflow.

1.2.1.1 Window-based

The typical window-based congestion control mechanisms are TCP and its related TCP-like control algorithms, such as DCCP-CCID2 (Congestion Control Identifier) [FlKo06b]. In 1987, a collection of several algorithms called Slow-start [Jaco88] is developed by Van Jacobson and Mike Karels. This was the initial version of the later widely adopted TCP congestion control policy. Successful Internet experiences with the Slow-start algorithms (Slow Start, Congestion Avoidance and Fast Retransmit) led them to be mandatory for TCP-Tahoe. Before Tahoe, TCP implementations applied a Go-Back-N model [Sast75], which

used cumulative positive ACK (acknowledgement) and required a retransmit timer expiration to re-send lost packets. The Tahoe implementation [Jaco88] followed this model and added the Slow-start algorithms (dictated by network condition) in the purpose of controlling the sender's window size according to the congestion level in the network.

Retaining the Slow-start algorithms incorporated in Tahoe, a new feature in the Fast Retransmit operation called Fast Recovery [Jaco90] is added in TCP-Reno [Stev94, Stev97, AlPa99]. With the Fast Recovery algorithm, the sender halves its congestion window size instead of resetting it to one after Fast Retransmit. Instead of slow-starting in Tahoe, Reno uses additional incoming duplicate ACKs to clock subsequent outgoing packets. Reno's Fast Recovery algorithm can improve the link utilization in the case that only one packet is dropped in a certain window. However, it suffers from performance problems when many packets are dropped from one window. In this case, the sender will have to wait for retransmit timer expiration before it can send packets again after the first fast retransmission.

A variation of Reno was proposed - New-Reno TCP [Hoe96, FlHe99] to solve the performance problems of Reno in the case of multiple-packet loss from one window by making a small change of Reno's Fast Recovery algorithm. New-Reno does not take TCP out of Fast Recovery state upon receiving partial ACKs as Reno does. Instead, it assumes that the packets immediately following the acknowledged one have been lost. Hence, it retransmits packets without waiting for the occurrence of retransmission timeout, which increases the sender's throughput significantly when multiple-packet loss happens in one window during the data transmission.

In addition to New-Reno, there are a few other Reno extensions proposed in recent years, such as FACK (Forward Acknowledgement) TCP [MaMa96a], SACK (Selected Acknowledgement) TCP [MaMa96b], and Vegas TCP [BrOM94]. A lot of work has been done on the TCP model and flow analysis, as well as the comparison of the algorithms mentioned above [FaFl96, BaPa97, MaSe97, and Morr97].

DCCP proposed by Kohler et al. in July 2001 aimed at implementing congestion control on a UDP-like no data retransmission protocol.DCCP-CCID2 algorithm is based on the AIMD (Additive Increase Multiplicative Decrease) algorithm for window-based flow control. As for TCP, the window size of the algorithm is given as the CWND (Congestion

Window Size) which is equal to the maximum number of out stage packets allowed in the network at any time.

While we appreciate the performance improvement in network congestion control achieved with various TCP and relative TCP-like algorithms, they all have the following two problems in common: (1) TCP sources cannot maintain a relatively stable transmission rate; and (2) they cannot improve the queue size stability in the network gateways.

1.2.1.2 Rate-based

To overcome the shortcomings of window-based control mechanism and achieve the co-existence with the widely used TCP algorithm, a suite of congestion control mechanisms called rate-based control mechanism which are able to modify their data transfer rates according to the available bandwidth within the network was proposed. The typical rate-based control algorithms are ATM (Asynchronous Transfer Mode) and TCP-friendly congestion control mechanisms, e.g. TFRC (TCP Friendly Rate Control) [HaFl03], DCCP-CCID3 [FlKo06c].

As a connection-oriented technology, ATM has proved very successful in the wide-area network scenario and numerous telecommunication providers have implemented ATM in their wide-area network cores. Also many ADSL (Asymmetric Digital Subscriber Line) implementations use ATM. However, ATM has failed to gain wide use as a LAN technology, and its complexity has held back its full deployment as the single integrating network technology in the way that its inventors originally intended.

TFRC is a TCP friendly rate-based congestion control protocol, which intends to compete fairly for bandwidth with TCP flows. The data sending rate is adjusted in response to the level of congestion as it is indicated by the loss rate. This adjustment is "gentle", that is, its instantaneous throughput has a much lower variation over time comparing with TCP. The smoothing of the data sending rate makes TFRC especially suitable for streaming media or other applications which require a smooth transmission throughput. However, the tradeoff for the smoothness is that the protocol becomes less responsive to bandwidth availability. Furthermore, TFRC is designed for applications that use fixed sized packets. In case of applications with a various packet size (e.g., audio applications), TRFC becomes not

compatible. A TFRC variant named TFRC-SP (TFRC Small-Packet) [FlKo06a] is proposed to be a substitution.

Without doubt, rate-based control mechanisms provides smooth throughput on data communications, while they still cannot solve well the bottleneck queue size stability problem. Rate-based control also causes serious problems on resource sharing fairness.

1.2.2 Router-based Mechanisms

For TCP Reno sources, the congested bottleneck router acts like a black box. The TCP source can only make its adjustment about the network work status by the implicit information it receives from the router (i.e. dropped packets). On the other hand, TCP Reno sources are also the black boxes to the router. The router cannot help the TCP sources explicitly to set their congestion control schemes more efficiently. For this reason, the TCP Reno and the router combination can never achieve the optimal performance for link utilization and gateway router queue size.

Many researchers have noticed this problem and proposed new schemes that explicitly convey the information of the router buffer occupancy in the TCP header to the TCP senders to let the senders make precise regulation of their data transmission rate. This group of congestion methods is called the router-based mechanisms because they all fall into the concept that the information about the congested router is fed back to the source.

Like the host-based mechanisms, router-based mechanisms can be claimed under window-based and rate-based. The most famous window-based mechanism RED [FlJa93], as well as the VCP algorithm are considered in this thesis. The most famous rate-based router mechanism is XCP [KaHa02]. The good results of PI-Rate algorithm [HoYa06] also make it a promising candidate to support streaming traffic.

Since the allocation of the limited amount of resources such as buffer queue size and output port bandwidth of the router can cause many problems. Scheduling is also a part of router-based mechanism e.g. [Hoe96]. A queue scheduling discipline allows the router to select the next packet that is transmitted on a port, hence manage access to a fixed amount of output bandwidth. Congestion occurs when packets arrive at an output port faster than they can be transmitted. The packets will be dropped due to the router buffer exhaustion. However, we do not focus on scheduling in this thesis, as we are only interested in queue management.

1.2.2.1 Queue Management

A properly designed queue management method for the Internet gateway is very important to the network performance. Good queue management techniques can utilize the link capacity more efficiently and make the network traffic stable. Otherwise, the buffer of the router can be easily overloaded or go into idle state. In the former case, a large number of packets are lost and require retransmission, which will reduce the source transmission rate and may result in the system to be in the latter case. Serious packet losses can even cause network collapse and shut down the whole network.

The simplest and most commonly used algorithm in the Internet gateway is DT (Drop Tail), which drops any incoming packets from the tail of the buffer if it has been full. This algorithm has the advantage of simplicity, suitability to heterogeneity and the nature of decentralization, but it has some serious disadvantages. It lacks fairness and cannot protect the responsive sources against non-responsive sources, which do not reduce their data transmission rate to respond to the congestion signals sent from the gateway router. Furthermore, its operation mechanism makes it difficult to balance between the link utilization and the transmission delay because it can easily trigger the global synchronization, which in turn may cause all the sources to increase transmission rate and result in congestion at once. Then, they will all decrease their rate and cause many links underutilized as a result. Similar to DT, DH (Drop Head) algorithm has the same advantages and disadvantages.

AQM (Active Queue Management) was proposed to solve the problems caused by the traditional queue management algorithms. The most famous AQM algorithms are RED (Random Early Detection) and its ramification. RED is widely adopted by the industry to work together with TCP source-based congestion control. However, the buffer queue instability and unnecessarily dropping data packets are the main problems caused by RED.

XCP (Explicit Congestion Control Protocol) [KaHa02] addresses the above problem. However, it requires multiple ECN (Explicit Congestion Notification) bits to encode the congestion control information exchanged between routers and end-hosts. Unfortunately, there are only two ECN bits allowed in the IP header currently. Solving this problem involves a non-trivial and time-consuming standardization process.

VCP (Variable-structure congestion Control Protocol) [XiSu05] (to be described in Section 2.4) is proposed to achieve high link utilization, low buffer queue length, negligible packet loss rate by leveraging the existing only two ECN bits. It has been verified that it can achieve comparable good performance to XCP with simple operation.

1.2.3 Congestion Control over WiMAX Links

During the past few years several wireless technologies have been developed and deployed. Broadband wireless access is gaining a great deal of interest from the networking research community. The recently standardized IEEE 802.16 WiMAX (Worldwide Interoperability for Microwave Access) [IeWi05] presents interesting perspectives, notably due to its capacity to offer consistent bandwidth and therefore consistent QoS. However, the behavior of network protocols, such as congestion control algorithms, has not been well studied in WiMAX applications. This is a problem that could slow down the widespread deployment of WiMAX.

Most of the wireless technologies have incorporated TCP/IP into their protocol stacks. However, TCP was designed for wired networks. Its sliding window and congestion avoidance mechanisms were designed to avoid routers congestion. However, in a typical wireless communication system, it is common that packets are lost because of the link error problem. When packets are dropped for reasons other than congestion, the TCP sources unnecessarily reduce their source throughput. For this reason, the network performances suffer from the link-error-caused packet losses.

To solve the performance degradation of TCP over wireless networks, several schemes have been proposed to alleviate the effects of non-congestion-related packet losses on TCP performance over networks that contain wireless links. Snoop [BaSe95] is one of these schemes that address the performance problem of TCP over wireless links. The Snoop protocol introduces a new module, which is called the snoop agent at the base station, so that the retransmission due to packet loss is only conducted over the wireless link and the wired links do not pass the duplicate ACKs to the TCP source. Hence, the TCP source does not reduce window size unnecessarily.

Another well-known scheme is TCP Westwood [CaGe00, CaGe01, and GeSa01], which uses the bandwidth estimation to determine the congestion window size value at the

TCP sources when the fast retransmit function is activated. The throughput of the TCP senders can be improved with TCP Westwood. TCP Westwood has the advantage that it only needs the sender-side modification of current congestion control schemes and is totally transparent to the router, but is cannot help to stabilize the queue size in the router.

The algorithms aimed at improving the TCP performance for network containing wireless links are classified into three groups [BaPa97]: end-to-end proposals, split-connection proposals and link-layer proposals. The group of end-to-end algorithms allows the TCP sender to tell the difference between congestion-caused packet loss and wireless-link-related loss. The split-connection group completely hides the wireless link from the source by terminating the TCP connection at the base station and uses a separate reliable connection between the station and the destination. The third group of algorithms, link-layer group, hides the link-error-caused packet losses from the TCP source by using local retransmissions over wireless links. Local retransmission is conducted with the techniques that can increase the performance significantly. The work of [BaPa97] also provides performance analysis and comparison of TCP over wireless links for these three groups of algorithms. The link layer proposals have the problems of the competing retransmissions between the two layers and the negative effect of link layer protocol on the TCP fast retransmission. The others only concern the TCP performance, and cannot benefit the gateway performance at the same time.

1.3 Thesis Motivation

Since V. Jacobson proposed TCP-Tahoe in 1988, the TCP congestion control algorithms have been upgraded several times, including Reno, New-Reno, SACK, FACK, Vegas, and Westwood as reviewed before. Each of these new TCP mechanisms has often improved the TCP throughput and congestion control performance to a higher level. However, the absolute reliable TCP cannot satisfy the requirement of real-time multimedia communications. UDP (User Datagram Protocol) was proposed to mitigate delay jitter effectively without congestion control and packet retransmission. However, this would cause too many packets dropped and lost, thus deteriorating the quality of multimedia traffic. To decouple loss recovery from congestion control, DCCP (Datagram Congestion Control Protocol) [KoHa06] provides congestion control without packet retransmission for streaming media.

Since there is no data retransmission in DCCP, an efficient network operation with fair bandwidth allocation and minimized packet loss becomes essential. Also, the DCCP source throughput has too much fluctuation. The early congestion notification in RED relies on the unnecessary loss of packets, which wastes network resources, increases timeouts, causes too much fluctuation on source sending rate and may cause an unhealthy multimedia communication environment.

The current TCP/RED control strategy suffers from performance degradation when the network involves communications over wireless channels. WiMAX, as one of the most promising wireless techniques, has much greater bandwidth than the other wireless networks. However, currently the congestion control performance of different classical mechanisms on WiMAX applications is still not well investigated yet. The network performance of WiMAX networks may suffer from the high mobility and wireless bandwidth environment. Stabilizing the buffer queue and limiting packet loss from congestion becomes more difficult than before.

Based on the above, it is desirable to have a traffic and congestion control scheme that satisfies the following four requirements: (1) can stabilize the queue size in the router; (2) can avoid the unnecessary packet loss; (3) can achieve the global fairness and smoothing the source throughput, and (4) can improve the network performance over wireless links. The motivation of this thesis is to propose a control strategy with easy implementation using the available two ECN bits in the IP header.

1.4 Objectives

In general, we are interested in designing a network traffic and congestion control strategy that can meet the conditions mentioned in the last section. However, the research in this thesis will focus on the following aspects:

1. Propose and implement a traffic and congestion control strategy to support a healthy real-time multimedia communications.
2. Mathematically analyze properties of the proposed traffic and congestion control strategy.
3. Compare the performance of the new traffic and congestion control strategy with some other common strategies in the wired and WiMAX networks environment.

1.5 Methodologies

It is noted that the source-based multimedia communication control mechanism DCCP-CCID2 is a window-based congestion control scheme. VCP that conducts congestion control according to the traffic load of the congested router also intends to avoid congestion by changing the source node's window size. The MIAIMD (Multiplicative Increase Additive Increase Multiplicative Decrease) window adjusting algorithm of VCP can help the window-based DCCP-CCID2 change the source window size more efficiently with respect to the congestion avoidance issue. Thus, to achieve the goals mentioned in the last section, we would like to implement the VCP algorithm to control a network with a large number of DCCP-CCID2 sources. This thesis focuses on DCCP-CCID2. For simplicity reasons, we use DCCP for DCCP-CCID2.

We shall use mathematical analysis to provide more insight into the DCCP/VCP network operation, by focusing on the equilibrium properties for DCCP/VCP networks. We first use a simplified continuous-time fluid model for our network operation to conduct our mathematical analysis. The steady state equilibrium window size and source sending rate will be investigated by our mathematical model.

We shall use OPNET [Opne05] simulation to conduct performance evaluation of the proposed congestion control strategy DCCP/VCP. OPNET is an engineering system capable of simulating large communication networks with detailed protocol modeling and performance analysis. It is a mature commercial tool and has a complete user-oriented procedure for installation and simulation development, which makes it the best choice for the simulation work in this thesis. Unlike a mathematical level simulator (e.g. Matlab), OPNET is a packet level simulation tool whose simulation and measurement are based on the real data packets sending and receiving. We shall perform a great deal of simulation work needs to be done to study the network behavior in different scenarios.

We shall study the congestion control algorithms in three different bottleneck network models: the wired network model, the WiMAX network model, and the mixed (wired/WiMAX) network model to see how the congestion control strategy perform in different communication environments. We create bottleneck network to study congestion and evaluate the performance of congestion control in terms of source throughput, queue size, and packet drop rate by simulations.

1.6 Thesis Contributions

The following are the contributions of this thesis:

1. Proposal of DCCP/VCP network operation strategy for real-time multimedia network traffic and congestion control.
2. Providing a fluid model based mathematical analysis of equilibrium properties for DCCP/VCP network.
3. Performance analysis and comparison of TCP and DCCP AQM algorithms in wired and WiMAX network environments.
4. Development of the OPNET simulation models of large scale DCCP-CCID2, DCCP-CCID3 and WiMAX bottleneck network models.
5. Implementation of various congestion control algorithms, especially VCP algorithm, for analysis and comparison.

1.7 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 provides the preliminary background involved in this thesis. Chapter 3 describes the network operation, models and assumptions for the wired network model, the WiMAX network model and the mixed (wired/WiMAX) network model, the long-lived DCCP flow, and the short-lived DCCP flow, the definition of the performance measures. In Chapter 4, we provide the control mechanism and implementation of DCCP/VCP. Mathematical analysis based on a fluid model is also conducted. Chapter 5 presents the performance evaluation and comparison for TCP and DCCP AQM congestion control strategies in the wired network environment. The performance evaluation and comparison for TCP and DCCP AQM congestion control strategies in the WiMAX and mixed network environments are presented in Chapter 6. Chapter 7 concludes the thesis and provides some suggestions for the potential future work.

Chapter 2
Background

This chapter discusses the detailed description of the main related background to this thesis. The newly proposed source-based DCCP traffic and congestion control algorithms (i.e. DCCP-CCID2 and DCCP-CCID3) are detailed in the chapter. Additionally, this chapter provides the properties of the VCP controller that we implemented in this thesis. The detailed functionalities about VCP algorithm will be given. The final section gives a brief overview of the WiMAX standard as described in [IeWi05].

2.1 TCP-based Congestion Control and Multimedia Transmission

Historically, the great majority of Internet traffic has used the reliable source-based traffic and congestion control TCP, with UDP making up most of the remainder. TCP's three connection steps, notification for connection, data transmission, and disconnection, with its data retransmission mechanism, provide absolute reliable Internet communication and congestion control scheme. However, in the last few years there has been a steady growth of multimedia applications such as Internet telephony, streaming video and on-line games, which generate long-lived flows of UDP datagrams and share a preference for timeliness over reliability. For these applications, data not delivered within some deadline (typically a small number of round-trip times) will not be useful at the receiver. TCP can introduce arbitrary delay because of its reliability and in-order delivery requirements; thus, these applications use non-congestion-controlled UDP instead. This lack of congestion control poses a threat to the network: if these applications' usage continued to grow, the Internet might be at real risk of congestion collapse.

2.2 DCCP Traffic Flows

DCCP, the Datagram Congestion Control Protocol, was first introduced as the Datagram Control Protocol (DCP) in the IETF [KoHa01]. IETF changed the name to DCCP and created a new DCCP working group [IeDc02]. DCCP is a new transport protocol in the

TCP/UDP family that provides a congestion-controlled flow of "unreliable" datagrams. Delay-sensitive applications, such as streaming media and telephony, prefer timeliness to reliability. DCCP was proposed to make it easy to deploy these applications without risking congestion collapse. Furthermore, DCCP can choose dynamically from different congestion control algorithm in the UDP-like transport layer to support multimedia transmission. This highlighted direction has been documented in [KoHa06] for future congestion control, and has attracted the interest of many researchers since its publication.

Currently, there are multiple prototype implementations, for example for the Linux and FreeBSD operating systems. One of the most attractive features of DCCP is dynamic selection among multiple congestion control algorithms. Of these algorithms, CCID2 [FlKo06b] and CCID3 [FlKo06c] are currently being standardized. The CCID2 algorithm is a TCP-like congestion control algorithm that is designed to satisfy such multimedia applications have bursty flows. The CCID3 algorithm is equivalent in many ways to that of TFRC, which has been evaluated by many researchers. This algorithm can provide smooth flow throughput to the applications that is sensitive to sudden flow throughput bursts. We mainly focus on DCCP-CCID2 in this thesis.

2.2.1 DCCP-CCID2

As a TCP-like Congestion Control, DCCP-CCID2 is appropriate for the applications that simply need to transfer as much data as possible in as short a time as possible and flows that would like to receive as much bandwidth as possible over the long term, consistent with the use of source-based congestion control. DCCP-CCID2 flows must also tolerate the large sending rate variations characteristic of AIMD congestion control, including halving of the congestion window in response to a congestion event.

DCCP-CCID2 algorithm is based on the AIMD algorithm for window-based flow control. As for TCP, the window size of the algorithm is given as the CWND (Congestion Window Size) which is equal to the maximum number of out stage packets allowed in the network at any time. Sender hosts adjust the congestion window size through congestion estimation according to the sequence of the ACK receptions. The congestion window size is increased by one packet in the following cases: (1) every acknowledged packet arrives in a slow-start phase, and (2) every window of data is acknowledged without lost packets in a

congestion-avoidance phase. On the other hand, the congestion window size is halved when the sender can infer that there are lost of packets due to duplicate acknowledgments, which is equivalent to TCP. If an ACK packet does not arrive at the sender before the timeout timer expiry (i.e., when an entire window of packets is lost), the sender resets the congestion window size to one. In this way, the sending rate is adjusted to avoid network congestion.

DCCP-CCID2 introduces a new parameter, ACK Ratio (Acknowledgement Ratio) ζ, to avoid the congestion on the reverse path and make ACK streams friendly with respect to TCP. The ACK Ratio ζ is defined as the number of data packets to be covered by an acknowledgement from the receiver. That is, the DCCP destination node acknowledges the received packets by sending one ACK back to the source node per ζ received data packets. ACK Ratio cannot be either smaller than 2 or exceed the congestion window size.

Table 2-2 below lists the main differences between DCCP-CCID2 and TCP.

Table 2-1 Main differences between DCCP-CCID2 and TCP

	TCP	DCCP-CCID2
Congestion control to the reverse path (ACK)?	No	Yes
Retransmission of data packets?	Yes	No
Units of the congestion control parameters (e.g. *cwnd*)	Bytes	Packets

2.2.2 DCCP-CCID3

DCCP-CCID3 uses another completely different approach (i.e. TFRC) than the TCP-like congestion control in DCCP-CCID2. Since DCCP-CCID 2's AIMD algorithm halves the sending rate in response to each congestion event and thus cannot provide a relatively smooth sending rate, TFRC congestion control algorithm is used in DCCP-CCID3. It makes DCCP-CCID3 appropriate for flows that prefer to minimize abrupt changes in the sending rate, including streaming media applications with small or moderate receiver buffering before playback.

The DCCP-CCID3 sender uses a sending rate in stead of congestion window size, and the receiver sends feedback to the sender roughly once per round-trip time reporting the loss event rate calculated by the receiver. The sender uses the reported loss event rate to

determine its sending rate. If the sender receives no feedback from the receiver for several round trip times, then the sender halves its sending rate.

The price of having smooth throughput while competing fairly for bandwidth with TCP in DCCP-CCID3 is that, the TFRC algorithm in CCID3 responds slower to changes in available bandwidth than do CCID2 or TCP.

2.3 AQM

AQM that dynamically manages the queue size in the gateway router has been proposed to solve the problems caused by the gateway congestion control algorithm of DT/DH plus FIFO. AQM algorithms drop incoming packets randomly according to the output of some probability function instead of waiting until buffer overflow to drop the packets in DT/DH. The key idea of AQM algorithms is to convey congestion notification early to the network traffic sources in order for the sources to reduce their transmission rate before the router buffer gets overflow and has to drop all the incoming packets. Therefore, AQM algorithms should be able to achieve better queue size control performance, higher link utilization and shorter traffic delay comparing with the static queue management methods such as DT and DH.

The well-known and most widely adopted AQM algorithm, Random Early Detection (RED) (Ref: Appendix 1) proposed by [FlJa93] has attracted a lot of researchers to work on its theoretical analysis and performance evaluation [FiBo00, TiMa01, HoMi01a, KuLa01, LoPa02]. In RED, there are some important parameters need to be set properly to make it perform well. While RED outperforms DT or DH significantly, it still has the following obvious problems. First, early congestion notification is built on the sacrifice of unnecessary packet loss. Second, RED is unable to control the queue size very well because the queue sizes under the control of RED may oscillate dramatically, which is one phenomenon of network instability. There has been various improvement ranging from SRED [FeKa99] to some recent work rate control RED [ZhYa03] and PI-RED [HoMi01b].

2.4 VCP

VCP (Variable-structure congestion Control Protocol) is proposed [XiSu05] to overcome the shortcomings of the traditional AQM algorithms (i.e. RED and its variants) and the

operational complexities of the XCP algorithm. The design guideline of the new router-based congestion control algorithm was to develop a simple congestion control mechanism that can scale to high BDP (Bandwidth-Delay Product) networks. Thus, an XCP-like congestion control protocol, that is called Variable-structure congestion Control Protocol, is proposed by decoupling efficiency control and fairness control with using a traffic load factor as the congestion signal.

VCP requires explicit feedback from the gateway router about the status of congestion. It relies on the two ECN bits that are available in current implementations of the IP header. Each gateway router in the network calculates a traffic load factor ρ that is used to classify the level of congestion into three states: low-load ($0 \leqq \rho < 80\%$), high-load ($80\% \leqq \rho < 1$), and overload ($\rho \geqq 1$). The routers are responsible to encode the current congestion state into the two ECN bits: $(01)_2$, $(10)_2$ and $(11)_2$, for low-load, high-load and overload respectively. Extracting the ECN bits from the packets sent by the router, the receiver sends the congestion information back to the sender via ACK packets.

Reserving the original AIMD algorithm, a new algorithm called MI (Multiplicative Increase) is introduced by VCP. Based on the traffic load states reported by the receiver, the sender uses one of the following algorithms: MI in the low-load state, AI in the high-load state, and MD in the overload state. By using MI in the low-load region, flows can exponentially ramp up their bandwidth to improve network utilization. Once high utilization is attained, AIMD provides long-term fairness amongst the competing flows. It is ensured that the flow throughput is smooth and the link utilization is full used and fairly shared by each of the flows traversing the VCP router.

From a practical point of view, VCP has two advantages: (1) VCP does not require any modification to the IP header since it can reuse the two ECN bits in a way that is compatible with the ECN proposal, and (2) VCP is a simple protocol with low algorithmic complexity. The functionalities of the source and destination nodes algorithm under VCP are similar to that of TCP. As a router-based congestion control mechanism, VCP algorithm maintains no per-flow state, and it has very low computation complexity.

2.5 WiMAX Nodes

IEEE Standard 802.16e WiMAX [IeWi05], completed in 2005 and published on February 2006, defines the Wireless MAN air interface specification for wireless metropolitan area networks. The completion of this standard heralds the entry of broadband wireless access as a major new tool in the effort to link homes and businesses to core telecommunications networks worldwide.

The physical layer specification defined for 10–66GHz uses burst SC (Single-Carrier) modulation with adaptive burst profiling in which transmission parameters, including the modulation and coding schemes, may be adjusted individually to each SS (Subscriber Station) on a frame-by-frame basis. The 2–11GHz band, that includes both licensed and license-exempt, was designed to use OFDM (Orthogonal Frequency Division Multiplexing) and OFDMA (Orthogonal Frequency Division Multiplexing Access) channel access schemes. IEEE Standard 802.16e WiMAX [IeWi05] supports high mobility as well as high bandwidth. Hence power becomes an important issue. The transmission power ranging from 63.5dbm (2238.72W) to 64dbm (2511.89W) is expected. Power saving mode such as sleep request/ response message when the WiMAX node is in idle mode is also included in IEEE Standard 802.16e. Modulation schemes ensure that a quality signal is delivered over the distance by decreasing throughput. WiMAX is able to dynamically shift modulations from 64-QAM (Quadrature Amplitude Modulation) to QPSK (Quadrature Phase Shift Keying) via 16-QAM, displaying its ability to overcome QoS issues with dynamic bandwidth allocation over the distance between the BS (Base Station) and the SS (Subscriber Station).

WiMAX also incorporates a number of time-proven mechanisms to ensure good QoS. Most notable are TDD (Time Division Duplex) and FDD (Frequency Division Duplex). The WiMAX standard provides flexibility in spectrum usage by supporting both FDD and TDD. It can operate in both FDD/OFDM and TDD/OFDM modes. Channel bandwidths of 20 or 25MHz (typical U.S. allocation) or 28MHz (typical European allocation) are specified in WiMAX.

The WiMAX MAC (Medium Access Control) protocol was designed for point to multipoint broadband wireless access applications. It addresses the need for very high bit rates, both uplink and downlink. Access and bandwidth allocation algorithms must accommodate hundreds of terminals per channel, with terminals that may be shared by

multiple end users. The services required by these end users are varied in their nature and include legacy TDM (Time Division Multiplexing) voice and data, IP connectivity, and packetized VoIP (Voice over IP). To support this variety of services, the WiMAX MAC layer must accommodate both continuous and bursty traffic. Additionally, these services expect to be assigned QoS in keeping with the traffic types. Thus, TDMA (Time Division Multiplex Access) is used as the medium access method in WiMAX MAC layer.

WiMAX MAC ensures a number of QoS measures not seen in other wireless standards. Perhaps its greatest value is providing for dynamic bandwidth allocation. This is done by the dynamic changing of uplink and downlink subframes in one frame. In the case of TDD that we used in this thesis, the uplink and downlink transmission occur at different times and usually share the same frequency. A TDD frame has a fixed duration and contains one downlink and one uplink subframe. The TDD framing is adaptive in that the bandwidth allocated to the downlink versus the uplink can vary.

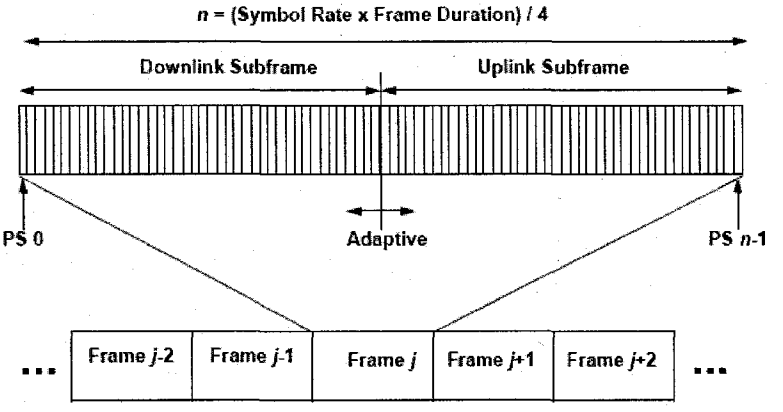


Figure 2-1: WiMAX TDD Frame Structure

Figure 2-1 shows the frame structure of WiMAX TDD frame structure. There are two parts in a frame, which are downlink subframe and uplink subframe. The length of each of the two subframes is adaptive. Each subframe is divided into many minislots. Since TDD sends frame according to TDM/TDMA. The minislots shares the same frequency but divided by time.

In TDD, the downlink subframe uses TDM portions to transmit data packets. However, in the uplink subframe, the data packets are transmitted based on TDMA to

support sudden burst. The subchannels in the frame share the same frequency but are separated by time. In order to support different kind of services (e.g. data and voice), WiMAX MAC accommodates both continuous and bursty traffic and assigns different QoS parameters to each service.

2.6 Concluding Remarks

As we discussed in this chapter, the widely adopted source-based Internet traffic and congestion control scheme TCP and the most popular router-based AQM algorithm RED are detailed. However, the co-operation of TCP/RED control strategy has its limitations. The reliable TCP scheme cannot well satisfy the requirement of real-time multimedia communications. UDP is used to support such multimedia communications as Internet telephony, streaming media and online gaming, but its lack of congestion control may cause the corruption of Internet.

The newly proposed DCCP intends to support the rapid increasing requirement of multimedia communication with its inbuilt congestion control scheme and UDP-like no data transmission mechanism. However, no matter DCCP/RED may cause random packet loss without retransmitting. VCP is introduced to solve this problem.

In this thesis, DCCP-CCID2, DCCP-CCID3 and VCP algorithms are implemented and analyzed. We mainly focus on DCCP-CCID2. The performance of DCCP/VCP is compared with that of TCP/RED, DCCP/RED and TCP/VCP. The congestion control over WiMAX links is another issue that we take into account in this thesis. DCCP/VCP and DCCP-CCID3 are addressed according to this issue.

Chapter 3
Network Operation, Models and Assumptions

This chapter details the general communication network operation mechanism as well as the network configuration used for the performance evaluation of the algorithms. The different network models, the traffic flow models, the OPNET simulation models, and the assumptions used in this thesis are also introduced.

3.1 Network Topology

In this thesis, we use wired and WiMAX network to evaluate the performance of our congestion control strategy. This section provides the details of the topologies of the two networks. We conceive a general bottleneck network as shown in Figure 3-1 with 10 source subnets and 10 destination subnets. Each subnet in turn conceives of 200 source nodes. We have evaluated two types of networks which are detailed in the following. This is only overview of the general network topology. The bottleneck router shown in Figure 3-1 may be connected with either wired or WiMAX subnets. The RTPD (Round Trip Time Delay) of each source subnet is varied.

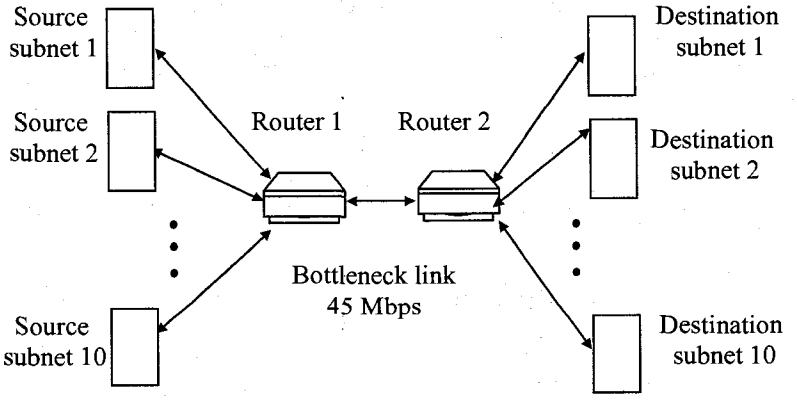


Figure 3-1: Bottleneck Network

3.1.1 General Network

In our study, we set a T3 (45M) link for the fixed wire. A multiplexer would aggregate the inputs. Figure 3-1 shows a simple example of a wired bottleneck network with 10 source subnets and 10 destination subnets. Each source/destination subnet has 20 DCCP source/destination nodes (there are 200 source nodes and 200 corresponding destination nodes in total). In a wired network, there is a fixed wire connecting a node to its router as shown in Figure 3-1. The DCCP sources in one subnet have the same propagation delay. The propagation delay for the DCCP sources in one subnet is different from that of each other subnet. The round trip time delay is composed of the RTPD (Round Trip Propagation Delay) and the queuing delay, which is varied for each subnet. A bottleneck congestion situation can occur in Router 1 if the packet arrival rate exceeds the packet service capacity.

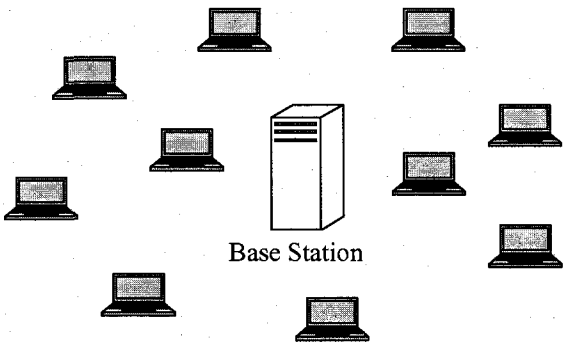


Figure 3-2: WiMAX Source Subnet

Figure 3-2 shows a simple example of a WiMAX source subnet with totally 10 mobile WiMAX source nodes. All WiMAX source nodes communicate with WiMAX base station. The base station sends the data packets received from the WiMAX nodes, to the bottleneck router. After extracting the destination IP address from the header of the data packets, the router deliver it to the corresponding destination node.

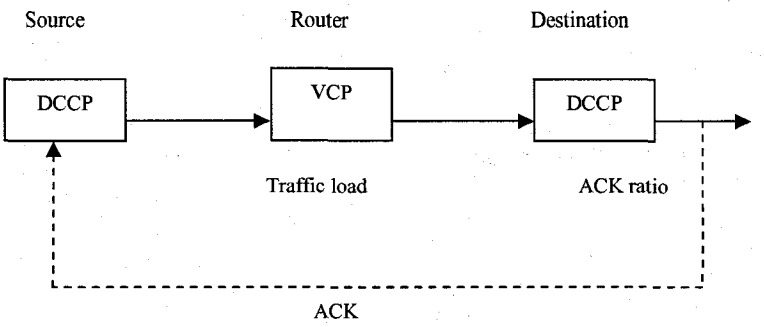


Figure 3-3: Block Diagram of DCCP/VCP Network Operation

3.1.2 Network Operation

Figure 3-3 depicts the functional blocks in our DCCP/VCP operation along a logical connection inside the network. A DCCP source node generates and sends its data packets including the source address and destination address in the IP header through a VCP router to its corresponding DCCP destination node. The DCCP-CCID2 sources start sending data packets with the initial slow start phase. The sender may send a data packet when *pipe* < *cwnd*, but must not send a data packet when *pipe* = *cwnd*. When the *cwnd* is smaller than the *ssthresh* (Slow Start Threshold), the sender is set at slow start phase. When the *cwnd* is greater or equal to the *ssthresh*, the sender is set as congestion avoidance phase. Although there is no retransmission of data packet, the *cwnd* is halved in case of a loss event.

The VCP controller appends control information to the data packets as they pass through in the router. The link between two adjacent functional block has a capacity of C = 45 M bits per second, and each packet emerging from the VCP router has a size of s bits. When a destination node is receiving packets, it dose not acknowledge each one. Instead the destination node acknowledges the received packets by sending one ACK back to the source node for every ζ data packets it received. We define the acknowledgement ratio ζ as the number of data packets sent for every acknowledgement received from the receiver. The acknowledgement ratio ensures that there is no ACK congestion. Once there is a loss event reported by the receiver, there is no retransmission effectively.

Since the bottleneck router is not so congested at the beginning of the initialization. The traffic load factor ρ is less than 80%. It encodes $(01)_2$ into the ECN bits in the header of the data packets and send them to the DCCP destination nodes. The DCCP source nodes

receive their ACKs from the DCCP destination nodes and change their congestion window according to the MI algorithm of VCP controller. Therefore, the congestion window size increases rapidly and the traffic load in the bottleneck router grows up fast to around 80%. Then, the traffic load factor ρ encodes $(10)_2$ into ECN bits and reports it to the DCCP destination nodes. After the DCCP source nodes receive the information that the bottleneck is going to be congested, they change their congestion window according to AI algorithm in VCP controller. This algorithm allows the congestion window to grow slower than that in the MI algorithm to carefully achieve high link utilization without buffer over flow. When the traffic load grows to a little bit larger than 1, the traffic load factor ρ encodes $(11)_2$ into the ECN bits and send it to the DCCP destination nodes. After the DCCP destination nodes report the congestion information to be DCCP source nodes by ACKs, the DCCP source nodes start to operate in the MD algorithm. Under this algorithm, the congestion window size of the source nodes decreases fast to avoid congestion happen in the bottleneck router. Therefore, the traffic load is reduced to be less than 1 but larger than 80%. The DCCP sources start to perform the AI algorithm again. Thus, AI-MD algorithm is alternatively governing the source's congestion window size. The high link utilization without congestion is achieved.

Notably that, the VCP controller has its limitation. When there is loss event, the source performs the same as TCP-Reno. That is, whenever there is a packet dropped, the congestion window size will be halved. In wired networks, there is almost no packet dropped, this limitation is not obvious. However, this limitation makes the source throughput performance worse in wireless networks. This is to be discussed in Chapter 6.

3.2 Traffic Control Mechanism

We use VCP algorithm in our research to control the DCCP-CCID2 traffic flows. We create a bottleneck router to study the network performance for the RED and VCP controller. The bottleneck router control mechanism for our implemented VCP algorithm can be described by the following analytical model, which we present below to introduce the model, symbols and notations to be used later.

Figure 3-4 describes one connection traversing a single-bottleneck VCP router that we implemented in thesis. The router compares the arrival traffic rate with the link capacity,

and calculates the ratio between them. Called load factor, it is a normalized parameter. So, it is independent of bandwidth or round trip time delay. The load factor ρ is classified into three states: low-load, high-load, or overload. The three states can be encoded into two ECN bits as done in [XiSu05], carried back by ACK to the sender. Then, the sender matches its control algorithm with the network load. The detail algorithm of the VCP controller that we implemented in our bottleneck network to control the DCCP-CCID2 traffic is as follows:

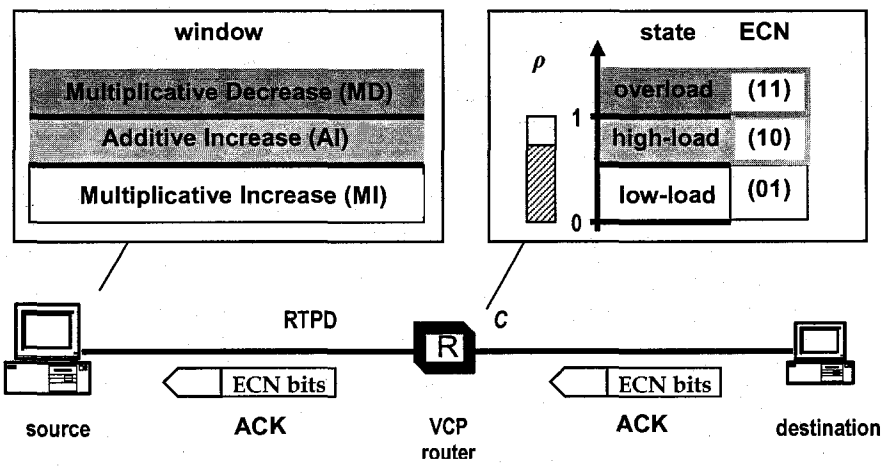


Figure 3-4: Analytical Model of a Bottleneck VCP Router

Since VCP uses the traffic load factor to report the traffic load, the traffic load factor ρ is calculated in the router by Equation (3-1),

$$\rho = {\lambda_l + k_q \cdot q_l} / {\gamma \cdot C \cdot t_p}. \tag{3-1}$$

where λ_l is the incoming traffic size during each t_p period. The parameter t_p is measured by a packet counter and is reinitialized to 0 when the time fires at t_p. The parameter k_q represents how fast to drain the link steady queue. The parameter q_l is the persistent queue size, which is measured by Equation (3-2) per t_q period. The parameter a is the exponential moving average weight. The link target utilization γ means that we try to control the link utilization under around γ. It ensures that there is almost no packet loss. The link capacity is C.

$$q_l = a \cdot q_l + (1-a) \cdot q(t). \tag{3-2}$$

When the load factor ρ ∈ [0, 0.8), where 0.8 is the constraint value identified in the VCP algorithm, the traffic load is in the low region. The router requests source to perform the MI (Multiplicative Decrease) algorithm, in which the source congestion window size is increased very fast to reach relatively high link utilization. When ρ ∈ [0.8, 1), the network is in high load. The source is set in the AI (Additive Increase) algorithm to achieve the source throughput steady fairness. When ρ ∈ [1, 8), the network is overloaded. The source performs the MD (Multiplicative Increase) algorithm to slow down the source sending rate. In MI, the flow throughput increases very fast to reach a bottleneck link utilization of 80%.

$$\text{MI:} \quad cwnd = cwnd + (1+\mu)^{\min(R/t_p, \sigma_{MI})} - 1. \tag{3-3}$$

$$\text{AI:} \quad cwnd = cwnd + \alpha \cdot \min((\frac{R}{t_p})^2, \sigma_{AI}) / cwnd. \tag{3-4}$$

$$\text{MD:} \quad cwnd = \max(1, \beta \cdot cwnd). \tag{3-5}$$

The *cwnd* is changed according to Equation (3-3), where μ is the MI parameter. The round trip time R is estimated as done in TCP. The parameter σ_{MI} is the MI scaling limit to bound the burst traffic introduced due to the MI scaling. After the link utilization reaches 80%, the load factor reports the source to perform AI and MD algorithms. The *cwnd* changes in AI based on Equation (3-4), σ_{AI} is the AI scaling limiter to bound the burst traffic introduced by AI scaling.

Example

We choose t_p=200ms, k_q=0.5, t_q=10ms, a=0.875, γ=0.98, μ=0.625, a=1, β=0.875, s_{AI}=10, and s_{MI}=2.5. This is an example for the source node to execute MD algorithm. After the AI algorithm, the traffic load is becoming higher, and getting close to the maximum value 100%. If the traffic load is over 100%, it means that the congestion of the flows is too high. The queue size of the buffer will overflow, which causes packet loss. To avoid this situation, once the traffic load reaches 100%, the router reports the source to go to MD algorithm described by Equation (3-5). MD forces the window size to rapidly decrease to a comparatively low value. To avoid the repetitious decreasing of *cwnd*, after doing Equation (3-5), the *cwnd*

calculation will not follow MD for one t_p period, then follows AI algorithm for one R.

Notably that VCP algorithm uses the target link utilization γ to avoid incoming data rate growing larger than the link capacity. This parameter ensures two things in our model. First, at the moment when the traffic load factor ρ reaches 1, the link utilization is actually slightly lower than 1. It ensures that there is no buffer overflow. Second, when the router reports the load factor to the source, it has to first encode the load factor into the two ECN bits before sending the packet out of router to the receiver. The receiver has to take the ECN bits out and encapsulate it into the acknowledgement. Then, it sends the acknowledgement back to the sender. Finally, the sender executes the MD algorithm. All of these procedures needs time, so the target utilization ensures no heavy traffic congestion and limits the packet loss rate. The target link utilization γ is set at 0.98 in our algorithm.

3.2.1 CCID2 Model

As mentioned in Section 2.2.1, DCCP-CCID2 algorithm is a source-based traffic and congestion control algorithm which controls the source's congestion window size according to the network congestion information in ACKs. In this section, we will provide the details about the DCCP-CCID2 algorithm.

The CCID2 is the DCCP source model we want to interconnect with VCP algorithm. A DCCP-CCID2 source sends packets in two phases as described in Table 3-1: the slow start phase and the congestion avoidance phase. A parameter called "*pipe*" is used to control the *cwnd*. It is defined as the sender's estimate of the number of data packets outstanding in the network.

Table 3-1: Different Phase in DCCP-CCID2 Source

Slow start phase (*cwnd* < *ssthresh*)	Congestion avoidance phase (*cwnd* = *ssthresh*)
■ the *cwnd* is increased by one for every received packet ■ *pipe* is decreased also by one for that received packet, it means that the sending rate is doubled per received packet	■ the *cwnd* is increased by one when there is no loss in the last *cwnd* packets ■ the sending rate is increased linearly

The ACK ratio ζ should always meet the following constraints in our model: (1) it is an integer. (2) it dose not exceed *cwnd*/2, when rounded up to an integer, except that the default value of ACK Ratio equals to 2 is always acceptable. (3) it is two or more for a congestion window of four or more packets. These constraints as ascertain that the minimum allowed value of *cwnd* are 2.

The sender changes the ACK Ratio ζ based on the following rules: 1) for each congestion window of data with the loss event, ζ is doubled (i.e. the sending rate of ACK is halved). 2) for each $cwnd \cdot \left(\zeta^2 - \zeta\right)^{-1}$ consecutive congestion windows of data without loss event, ζ is decreased by 1 (i.e. the sending rate of ACK is increased by $\left(\zeta^2 - \zeta\right)^{-1}$).

According to the above discussion, ACK ratio changes can be grouped as follow:

$$\zeta = \begin{cases} 2 & (\zeta \le 2) \\ 2\zeta & (\zeta > 2 \text{ and a loss per cwnd}) \\ \zeta - 1 & (\zeta > 2 \text{ and no loss per cwnd} \cdot (\zeta^2 - \zeta)^{-1}) \\ \zeta_{old} & (\zeta > 2 \text{ and } \zeta > cwnd/2) \end{cases}$$

Note that, *cwnd* usually varies over time, and the dynamics will be rather complex, but it is roughly TCP friendly since it dose not affect much on the existing TCP traffics. In our model, the DCCP-CCID2 sender uses the most recent value of *cwnd* to determine whether to decrease ACK Ratio ζ by 1.

Based on the above rules, the DCCP-CCID2 receiver in our model always sends at least one ACK per window of data when *cwnd* = 1, and at least two acknowledgements per window of data otherwise. Thus, the receiver could be sending two ACKs per window of data even in the face of very heavy congestion on the reverse path. We would note that under heavy congestion, all the ACKs are dropped. Then the sender falls back on an exponentially backed-off timeout. If congestion is sufficiently heavy on the reverse path, the DCCP-CCID2 sender reduces its sending rate on the forward path (i.e. data packet path), which reduces the sending rate on the reverse path (i.e. ACK path) as well.

3.2.2 CCID3 Model

This thesis also implements the DCCP-CCID3 algorithm in our bottleneck network model for comparison purpose. DCCP-CCID3 uses a throughput equation for the allowed sending rate

as a function of the loss event rate and round trip time delay. It is also a source-based control mechanism like DCCP-CCID2. However, CCID3 avoids network congestion by adjusting the source node's sending rate according to the network status that reported by ACK. The algorithm of CCID3 follows the TFRC mechanism standardized by the IETF RFC3448 [HaFl03]. More details can be found in Appendix D and the related references.

3.3 Traffic Flow Models

There are two types of traffic flows that we used in this thesis: the long-lived DCCP flow and the short-lived traffic flow. Both types of the flow sources have a transmission rate limited by the traffic control algorithm. For the VCP controller we implemented, the transmission rate of a long-lived DCCP source is controlled by the CCID2 algorithm. The details of the long-lived and short-lived DCCP-CCID2 flows are provided in Section 3.2.1.

3.3.1 Long-lived DCCP Flow Model

The long-lived DCCP source always tries to send as many data packets as possible. A good example is video conference, which transfers data for a long time without disconnection. This thesis focuses on the analysis and performance evaluation of using VCP algorithm to control the window-based source control mechanism DCCP-CCID2 (i.e. DCCP/VCP).

3.3.2 Short-lived Source Flow Model

The short-lived source works as the background noise and interfere to the whole network. This mechanism can evaluate the performance under the diversity of real world internet environment. Unlike the long-lived DCCP source, the short-lived source traffic flow is characterized by the uncertainty of the time of its entering/reentering the network and the time length of its being active. HTTP (Hypertext Transfer Protocol) source is a typical example of the short-lived traffic source, which enters the network after an undetermined think time, sends a file of an undetermined length, and then waits for another think time period.

Much study has been done by researchers on the traffic pattern of the shorted-lived TCP flows, e.g. [CrBe97], and various probability functions were proposed to describe the characteristics of the think time and file length distributions, such as the self-similarity model

[LeTa94] and the wide area traffic model [PaFl95]. For our work, we have adopted the model from [OtLa99]. The think time characteristic is depicted by the following probability distribution

$$P\{T > t\} = p_h\left[1+(\frac{t}{\pi_h})^{\eta_h}\right]^{-1} + p_l\left[1+(\frac{t}{\pi_l})^{\eta_l}\right]^{-1}. \quad (3\text{-}7)$$

where p_h, p_l, p_k, p_l, η_h and η_l are constant parameters, and $p_h + p_l = 1$. For the think time distribution, we use $p_h = 0.4953916$, $p_l = 0.5046084$, $p_l = 1.0$, $p_l = 0.0245032$, $\eta_h = 1.243437$, and $\eta_l = 3.252665$ in our model.

The file length (in bytes) probability distribution follows:

$$P\{F > f\} = \left[1+(\frac{f}{\pi})^{\alpha}\right]^{-1}. \quad (3\text{-}8)$$

in which we select $\alpha = 1.15066$ as its positive constant with a median value $\pi = 2190$ in our model. The distribution of Equation (3-8) has a finite mean but an indefinite variance. The distribution of Equation (3-7) is a mixture of two distributions of Equation (3-8). It has been shown that the traffic pattern of short-lived traffic flows following these two distributions is realistic [OtLa99].

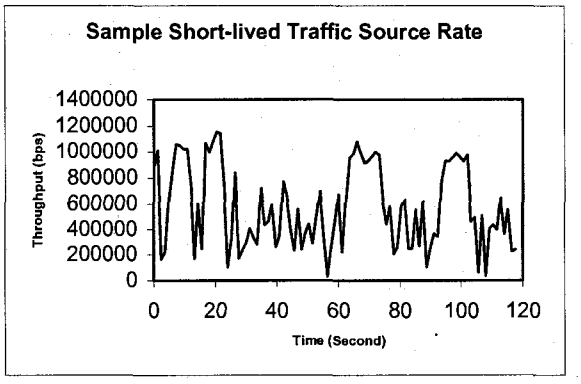


Figure 3-5: Transmission Rate of Sample Short-lived Traffic Flow [OtLa99]

The transmission rate of the short-lived traffic source is limited by the real-time bandwidth available on the network. Figure 3-5 shows the transmission pattern for a typical short-lived traffic flow, which we have verified to be commensurate to real traffic from the Internet.

3.4 Performance Measures

The network performance is evaluated by means of simulations and specific performance metrics. This work focuses on improving DCCP throughput, stabilizing the congested router buffer queue, reducing the packet drop rate. Thus, the network performance is largely improved.

There are four main performance measures in this thesis. They are source throughput, buffer queue size, packet drop rate, and round trip delay. Source throughput is defined to be the data rate in bits per second sent out by a DCCP source node. This is also referred to the source sending rate. Buffer queue size is defined to be the number of packets in the buffer that are waiting to be serviced by the router. Packet drop rate is defined to be the ratio between the number of packets dropped by the router and the number of packets received by the router in a 10ms time interval. Round trip delay is defined to be the time interval from the instant when a packet is sent out from the sender until the instant when its ACK is received by the sender.

Notably that, according to [Stev94, Stev97, AlPa99], fairness is one of the most important properties of TCP. That is, the average value of each node's source throughput is almost the same. This can be seen from the performance evaluation parts in this thesis.

To allow a fairness comparison, we use the same number of sources (we enable 200 TCP/DCCP sources), same scenarios (we change the number of flows at the same time), and same parameters (we set the same service rate in the router) among the compared strategies in this thesis.

3.5 OPNET Models

OPNET [OpNe05] is a simulation language capable of simulating large communication networks with detailed protocol modeling and performance analysis. Features include the graphical specification of models, the dynamic and event-scheduled simulation kernel, the integrated data analysis tools and the hierarchical and object based modeling.

Models built with OPNET are hierarchical. More detailed source codes are written in the OPNET Process Model which is the lowest level. The Process Model domain is structured as a FSM (Finite State Machine). The FSM can be structured with the help of a graphical editor that allows the user to specify the relationship between the states. The states

can then be programmed with a C-like language called Proto-C and OPNET Kernel function codes. The second level is the node level. Processes that were specified in the process domain, different source and destination modules offered by OPNET, as well as data generators and queues can then be grouped into nodes in the node domain. Nodes can then be connected to build up different network architects in the network domain. The DCCP packet is built under the OPNET Packet Format tool. The simulated time in our simulation can be set at OPNET Modeler Simulation Sequence. All the simulation results are collected and shown by the OPNET Tools: Probe Model and Analysis Configuration.

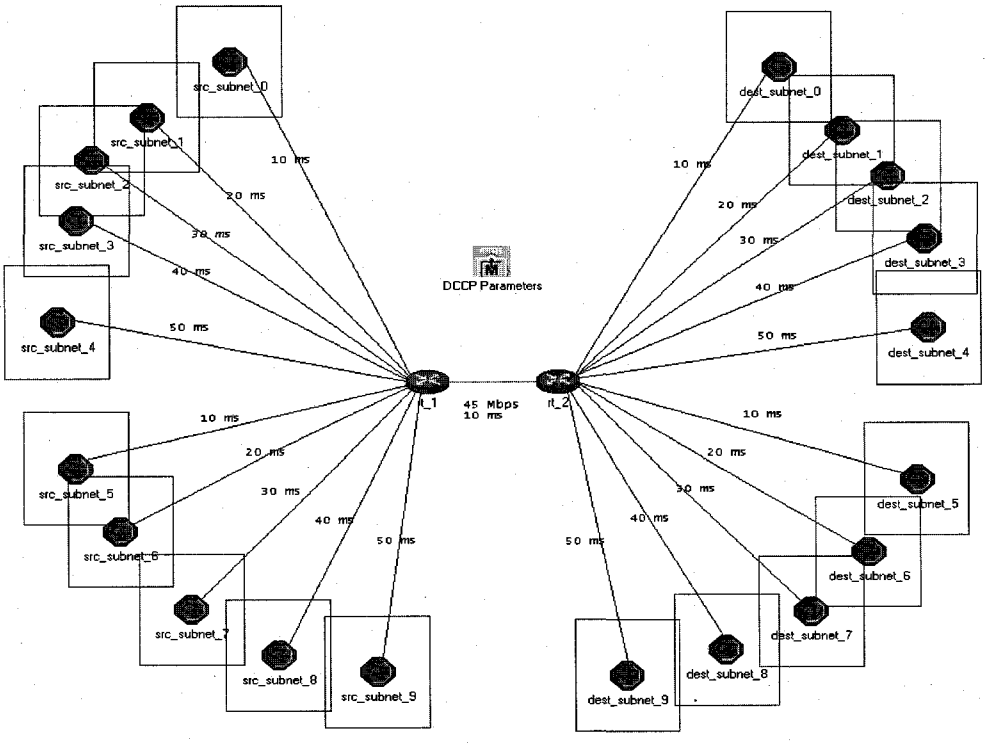


Figure 3-6: OPNET Network Model - Bottleneck

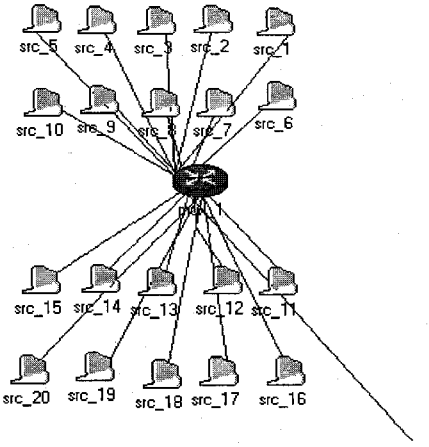


Figure 3-7: OPNET Network Model – Wired Source Subnet

Figure 3-6 depicts the network topology for OPNET simulation. There are 10 source subnets and 10 corresponding destination subnets labeled from 0 to 9 respectively as shown in Figure 3-6. Totally there are N=100 DCCP flows across the bottleneck router. Each source/destination subnet has 20 DCCP senders/receivers as shown in Figure 3-7. The bottleneck link capacity is T3 (i.e. C=45Mbps). The RTPD (Round Trip Propagation Delay) between sources and destinations varies from 30ms to 110ms. The average queuing delay can be approximated as 20ms. The packet size is s=1024 bytes. The two types of sources – long-lived DCCP source (e.g., video conference), which always tries to send as many data packets as possible, and the short-lived source (e.g., http), which enters the network after a random think time, sends a undermined length file, and then waits for another think period, can be set on our demand from the node "DCCP Parameters" which stores a set of preset simulation parameters.

Each source only sends packets to its corresponding destination, e.g., packets received by Receiver 1 in the Destination Subnet 1 came from Sender 1 in the Source Subnet 1. In Figure 3-7, the node named Mux_1 is a multiplexer which congregates the data packets sent by the 20 DCCP sources in the subnet and send out to the router. The buffer size of the multiplexer is assumed to be infinite in order to avoid the packet loss caused by the multiplexer buffer overflow. Unlike the multiplexer, the buffer size of the router is limited and set to be 700 packets in our simulation.

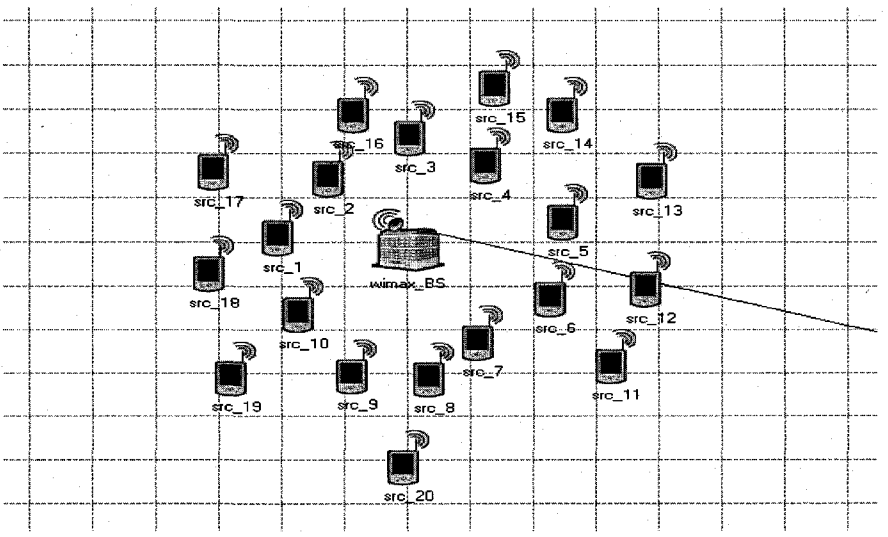


Figure 3-8: OPNET Network Model – WiMAX Source Subnet

WiMAX

We also implemented our simulation in WiMAX environment. Figure 3-8 depicts a source subnet with 20 wireless nodes. All wireless nodes are connected with the WiMAX base station over a WiMAX channel. They transmit the data packets to the router via the base station. Like the wired environment, the number of long-lived and short-lived sources can be set on our demand.

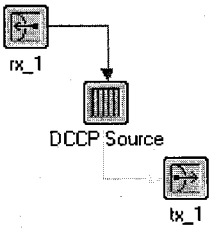


Figure 3-9: OPNET Node Model – Wired DCCP Source

Figure 3-9 shows the OPNET node model for the DCCP source. We can see that there are three different objects in the DCCP node model. The DCCP source generates packets according to some particular DCCP-CCID2 mechanisms and sends them to the transmitter (i.e. tx_1), which is an OPNET built-in object. The transmitter forwards the packets outside to the network link. The receiver (i.e. rx_1), also an OPNET built-in object, receives ACKs,

which are transmitted to the node through the network link, and forwards them to the DCCP source. The DCCP source then processes the ACKs and responds accordingly.

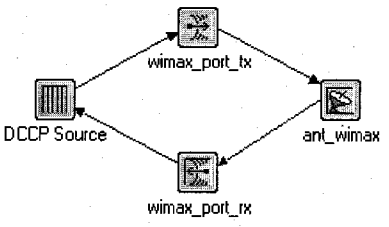


Figure 3-10: OPNET Node Model – WiMAX DCCP Source

The OPNET node model of the WiMAX DCCP source is depicted in Figure 3-10. The DCCP packets are also generated by DCCP source and then sent to the OPNET built-in object radio transmitter (i.e. wimax_port_tx). The transmission parameters of the WiMAX physical layer such as channel bandwidth, minimum frequency, and transmission power can be adjusted in this radio transmitter. The DCCP packets are then sent to the base station by an isotropic antenna (i.e. ant_wimax), which also receives ACKs from the base station for forwarding to the WiMAX receiver (i.e. wimax_port_rx). The ACK occupies the downlink and uplink slots as well. Unlike wired networks, the ACKs received by the WiMAX antenna may come from other WiMAX sources. The receiver has to check whether the ACKs match the channel of its own WiMAX source node. Then, it selectively sends the ACKs back to the DCCP source.

Since we use TDD for WiMAX node to support its physical layer, the uplink and downlink transmission occur at different times and usually share the same frequency. A TDD frame has a fixed duration and contains one downlink and one uplink subframe. The TDD framing is adaptive in that the bandwidth allocated to the downlink versus the uplink can vary. The downlink subframe transmits data packets based on TDM. However, in the uplink subframe, the data packets are transmitted based on TDMA to deal with bursty. The subchannels in the frame share the same frequency but are separated by time. The uplink and downlink operates alternatively.

For the details of the process models that we use in our simulation, please refer Appendix D. In addition to the DCCP source process model and the router process model,

there are still some other OPNET process models we used for the studies in this thesis. They are the DCCP destination process model, the source subnet multiplexer process model, the source subnet demultiplexer process model, the router ACK process model, the destination subnet multiplexer process model, the destination subnet demultiplexer process model, and the global parameter process model.

The OPNET models we created also include the link model, the packet format model, the probe model and the simulation sequence. They are necessary parts of the OPNET network models. All these models together become the whole set of simulation models we used to do simulations in the thesis.

3.6 Assumptions

The following assumptions are used throughout this thesis:

1. The destination does not send any ACK packets actively to the source receiver, unless it has received a packet from the source. This allows us to simplify the simulation and limit the effect of any factors other than those in the congestion control algorithms used in the performance comparison;
2. The buffer queue size of Router 2 (Node rt_2 in Figure 3-6) is big enough so that no packet loss will occur there. This assumption allows us to focus on the performance comparison in a one-bottleneck-router network;
3. All the packets have the same size;
4. The receiver window size of the DCCP destination is large enough that it does not affect the window size of the DCCP source. This would allow us to focus on the queuing performance evaluation of the congestion control algorithms residing in the router only.
5. Each source and its counterpart destination are one to one mapping and do not have cross traffic. This would allow us to simplify the simulation and evaluation of the queuing performance in the bottleneck router;
6. Each subnet has the same ratio of long-lived sources to short-lived sources for the simplicity of running simulation;
7. The buffer queue size of the WiMAX base station is large enough so that there is no other packet loss event (e.g. congestion loss) in the WiMAX base station than the

wireless link corruption. This assumption allows us to focus on one single bottleneck performance comparison.

3.7 Concluding Remarks

In this chapter, we detailed the main network operation as well as the general parameters and configuration that are used in this thesis. The control mechanism of VCP controller and operation of DCCP traffic sources that we implemented are also given. The OPNET simulation models that we developed and verified are provided. The assumptions considered for the performance comparison of the congestion control mechanisms are described.

Chapter 4
Implementation and Mathematical Analysis of DCCP/VCP Controller

In this chapter, we will present the mathematical analysis of the DCCP/VCP controller to be implemented. First, we will introduce the network traffic and congestion control mechanism specific to the DCCP/VCP controller. Then, we will provide the implementation method for the VCP router-based control algorithm. Finally, a mathematical analysis of the DCCP/VCP controller's equilibrium properties in steady state will be presented.

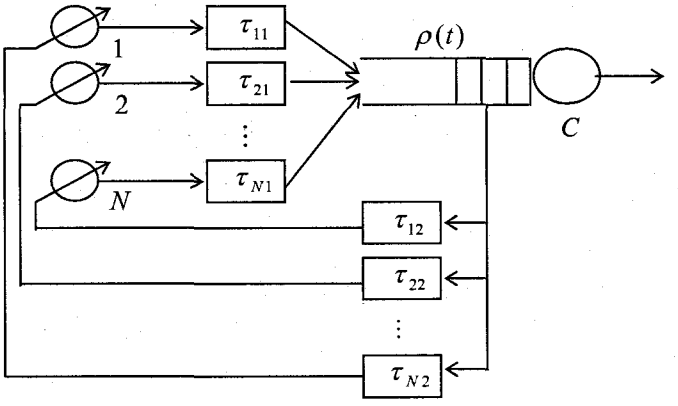


Figure 4-1: Analytical Model of a Bottleneck Network

4.1 Implementation

Our DCCP/VCP controller is implemented in both the DCCP source node and the VCP router. Figure 4-1 above depicts the analytical model of a bottleneck VCP router shown by the server and its buffer where N DCCP flows intersect. Divide the round trip time delay from the source to the congested router into two parts: (1) t_{i1} is the forward path delay from the ith source to the bottleneck router, and (2) t_{i2} is the backward delay from the bottleneck router back to the ith source through the destination. The service rate of the congested VCP router is C bits per second. The instantaneous traffic load factor of the VCP router is defined

to be $\rho(t)$. The buffer queue size of the bottleneck router is monitored and regulated continuously in a continuous-time fluid regulation way, a high piecewise stabilization of the queue size and a low packet drop rate are expected to be achieved and maintained.

```
Send Packets:
DCCP packets are sent out by the governing of cwnd, the slow start and
the congestion avoidance phases.
Timeout estimation:
if timeout then cwnd = cwnd/2;
Receive ACKs:
get the ECN bits from the header.
calculate the ACK ratio:
      if lost in the last cwnd
            ζ = min(cwnd/2, ζ*2);
            ζ = max(ζ, 2);
      if no lost per cwnd/(ζ^2-ζ)
            ζ = max(2, --ζ);
change the cwnd:
      if ECN bits are 00 or 01                  //MI
            inc = pow((1.0 + μ), min(rtt/t_p, s_MI))-1.0;
            cwnd = cwnd + inc;
      if ECN bits are 10                        //AI
            inc = a*min(sqrt(rtt/t_p), s_AI);
            cwnd = cwnd + inc/cwnd;
      if ECN bits are 11                        //MD
            cwnd = max(1, ß*cwnd);
      if packet drop
            cwnd = cwnd/2;
No data packet retransmission whenever there is a loss event or timeout.
```

Figure 4-2: DCCP/VCP Algorithm in the DCCP source
(See Table of Notation for explanation of Notations)

The details of our DCCP/VCP control mechanism can be found in Section 3.2. Our implementations are all based on the above discussion. There are multiple mechanisms in the source nodes and the router, such as initializing the values for each parameter, sending packets, receiving ACKs, timeout estimation, changing congestion window size, calculating

ACK ratio, calculating traffic load factor, and etc. In this section, we will provide the most important parts of implementation algorithms for the DCCP source node and the VCP router.

Figure 4-2 provides the detailed operation of our DCCP/VCP source nodes. The implementation of the algorithms in the DCCP source node consists 3 basic functions, i.e. sending packets, measuring timeout, and receiving ACK packets. First, sending data packets is the very basic function of a source node. As a window-based control mechanism, the sending rate of a DCCP-CCID2 node is governed by the congestion window size. DCCP-CCID2 source node operates exactly the same as that in TCP when estimating a timeout. However, in DCCP-CCID2 a timeout is seen to be a loss event when the congestion window size is halved without retransmission of data packets. When receiving an ACK from the destination node, the DCCP-CCID2 node first gets the ECN bits from the header of the ACK. Then, it calculates the acknowledgement ratio according to DCCP-CCID2's ACK ratio algorithm. Finally, the DCCP-CCID2 source node changes its congestion window size according to the MI, AI, and MD algorithms of VCP. The congestion window size is then halved whenever there is a loss event. There is no data packet retransmission always.

Figure 4-3 gives the basic implementation of our DCCP/VCP controller algorithms in the VCP router. The main functions of the VCP router are enqueuing the incoming data packets, calculating the traffic load factor, encoding the ECN bits, and sending the data packets to their destination nodes. For each incoming data packet, the VCP router first checks if the buffer queue is full. If there is a space for the packet, the router inserts the incoming data packet into the buffer queue. For each t_p=200ms, the router counts down the incoming traffic size. For every t_q=10ms, the VCP router updates its persistent buffer queue size like RED router does. That is, using $a=0.875$ as an exponential moving average as the RED algorithm to calculate the persistent queue size. After collecting the above information, the traffic load factor ρ is calculated in periods of t_p seconds. Then, the VCP router compares the value of ρ, and decides which ECN bits to encode and encapsulate into the header of the data packets. If the value of ρ is less than 0.8, VCP encapsulates 01 into the ECN header. If the value of ρ is between 0.8 and 1, 10 will be encapsulated into the ECN header. If the value of ρ is greater than 1, then we encapsulate 11 to be the ECN bits.

Finally, a data packet carrying the traffic load information is sent out to its DCCP destination node.

Enqueue the packets:

for each packet arrival

if q >= buffer size //q is the current queue size

discard the packet;

else

enqueue the packet;

per every t_p, update $\lambda_l = q_{in} * s$;

per every t_q, update $q_l = a*q_l+(1-a)*q(t)$; //a=0.875

Calculate the traffic load factor:

$$\rho = \frac{\lambda_l + k_q \cdot q_l}{\gamma \cdot C \cdot t_p}.$$

Encode ECN bits:

if ρ < 0.8

ECN = 01; //MI

if 0.8 = ρ < 1

ECN = 10; //AI

if ρ = 1

ECN = 11; //MD

Encapsulate the ECN bits into the header of the data packets.

Send the data packets to the destination node.

Figure 4-3: DCCP/VCP Algorithm in the VCP router
(See Table of Notation for explanation of Notations)

4.2 Mathematical Analysis

In this section, we provide some mathematical analysis in order to provide more insight into our proposed network operation shown in Figure 3-1. In networks with high bandwidth-delay product, a DCCP source performs the MD algorithm on each loss event. That is, the *cwnd* is halved once when a loss event happens. This can cause the *cwnd* suddenly decreases a lot when more than one loss event happen. Therefore, the AIMD algorithm produces more fluctuations in throughput and has a slow convergence under the high utilization. Our simple fluid model of the DCCP/VCP control strategy would help us to conduct the analysis of VCP in solving these problems.

4.2.1 A Fluid Model

In order to conduct our mathematical analysis on the DCCP/VCP net work model, we first consider a simplified continuous-time fluid model. We consider a single bottleneck router with an infinite buffer size shared by N DCCP flows. The sources are indexed by $i = 1,\ldots, N$, and we assume all the flows have the same round trip time.

As mentioned in Section 3.3, the following equations describe the activities of a DCCP source node's congestion window changing policy:

MI: $$cwnd = cwnd + (1+\mu)^{\min(R/t_p, \sigma_{MI})} - 1 \,. \tag{4-1}$$

AI: $$cwnd = cwnd + \alpha \cdot \min((\frac{R}{t_p})^2, \sigma_{AI}) \cdot cwnd^{-1} \,. \tag{4-2}$$

MD: $$cwnd = \max(1, \beta \cdot cwnd) \,. \tag{4-3}$$

With the calculation of the traffic load factor ρ in the VCP router:

$$\rho = (\lambda_l + k_q \cdot q_l) \cdot (\gamma \cdot C \cdot t_p)^{-1}. \tag{4-4}$$

We let

$$F(x) = h \cdot \frac{1-x}{x} \qquad (h>0), \tag{4-5}$$

where h is the coefficient of the MI parameter. Equation (4-5) describes the effect of the MI parameter. Integrating (4-5) with Equation (4-1), (4-2), (4-3), we obtain

$$\dot{w}_i(t) = \frac{1}{T} \cdot [w_i(t) \cdot F(\rho(t)) + \alpha] \,. \tag{4-6}$$

in which $w_i(t)$ is the instantaneous congestion window size of a DCCP flow i.

Equation (4-6) describes the whole MIAIMD model in our DCCP/VCP mathematical analysis. This model makes three simplifications comparing to the VCP protocol. First, it uses the exact load factor value $\rho(t)$, while VCP uses a quantized value of the load factor. Second, in the MI and AI phases, VCP uses either the multiplicative factor or the additive factor term, but not both as the MIAIMD model does. Third, in the overload state, VCP applies a constant MD parameter β instead of $F(\rho(t))$.

4.2.2 The Effect of Acknowledgement Ratio

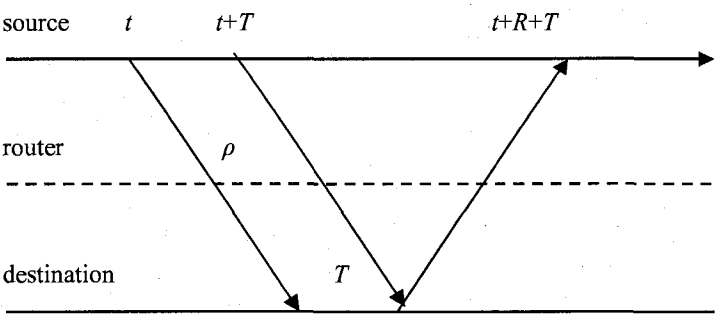


Figure 4-4: Effect of ACK Ratio in DCCP/VCP

In our DCCP/VCP network model, the acknowledgement ratio can influence on VCP controller to change the DCCP source *cwnd* since the congestion information is carried back to the DCCP source by the ACKs. According to the definition of ACK ratio ζ, acknowledgement will be sent back to the source every ζ received packets. Hence there is a delay T in the DCCP destination node to send an ACK as can be seen from Figure 4-4. This delay is obviously a function based on ACK ratio ζ.

To conduct our mathematical analysis, we have to find out T_{avg} of T (i.e. the average time period from the time instant when the first bit of a packet is sent out until the instant when the first bit of the ACK is returned) as follows:

Because of the queuing delay and the service delay of a packet, the packets arrive at the destination node every $\frac{s}{C}$ seconds, where s is the packet size in bits. Let T_j be the T for the jth packet. Then, we can obtain

$$T_1 = \frac{(\zeta - 1)\cdot s}{C},$$

$$T_2 = \frac{(\zeta - 2)\cdot s}{C},$$

$$T_3 = \frac{(\zeta - 3)\cdot s}{C},$$

$\cdots,$

$$T_{\zeta-1} = \frac{1\cdot s}{C},$$

$$T_\zeta = \frac{0 \cdot s}{C}.$$

Thus, the average time cycle that a packet is acknowledged is

$$\begin{aligned} T_{avg} &= \frac{T_1 + T_2 + T_3 + \ldots + T_{\zeta-1} + T_\zeta}{\zeta} \\ &= \frac{\frac{\zeta}{2} \cdot (\zeta - 1) \cdot \frac{s}{C}}{\zeta} \\ &= \frac{(\zeta - 1) \cdot s}{2C}. \end{aligned} \tag{4-7}$$

We introduce two important parameters which are round trip propagation delay R_p and queuing delay R_q. The round trip propagation delay is defined as the time a packet travels in the wire (but not in the router buffer) of the packet is sent out from the sender until it is acknowledged. Queuing delay is defined to be the time duration from the time of arrival of a packet until the time it is sent out into network. Thus $RTT = R_p + R_q$. In addition to the round trip propagation delay R_p and queuing delay R_q, the average time cycle that the *cwnd* is changed by the DCCP source is

$$\begin{aligned} R' &= R_p + R_q + \frac{s}{C} + T_{avg} \\ &= R_p + R_q + \frac{s}{C} + \frac{(\zeta - 1) \cdot s}{2C} \\ &= R_p + R_q + \frac{(\zeta + 1) \cdot s}{2C}. \end{aligned} \tag{4-8}$$

4.2.3 Steady State Analysis

With the above MIAIMD fluid model, accounting for the effect of the ACK ratio ζ, we are now ready to conduct the mathematical analysis. Our mathematical analysis focuses on window size and flow sending rate properties at equilibrium.

4.2.3.1 Windows Size

Window size is defined to be the number of packets that the receiver is currently willing to receive. We first conduct our mathematical analysis on the congestion window size of the DCCP source node. According to the above sections, we first substitute Equation (4-7) into Equation (4-6) and obtain

$$\dot{w}_i(t) = \frac{1}{R'} \cdot [w_i(t) \cdot F(\rho(t)) + \alpha] = \frac{2C \cdot [w_i(t) \cdot F(\rho(t)) + \alpha]}{2C \cdot (R_p + R_q) + (\zeta + 1) \cdot s} \quad (4\text{-}9)$$

when the traffic load factor $\rho_{(t)}$ can be obtained from Equation (4-4). Summing over all N flows in the router, we obtain

$$w(t) = \sum_{i=1}^{N} w_{i(t)} = \frac{1}{R} \cdot \left[F(\rho(t)) \cdot \sum_{i=1}^{N} w_i(t) + N \cdot \alpha\right] = \frac{2C \cdot [F(\rho(t)) \cdot w(t) + N \cdot \alpha]}{2C \cdot (R_p + R_q) + (\zeta + 1) \cdot s} \quad (4\text{-}10)$$

Based on Figure 4-4, the normalized traffic load ρ(t) received by the source at the moment $t+R+T$ can be computed by dividing the incoming sender's rate r at t by the maximum service rate (i.e. γC):

$$\rho(t) = \frac{r}{\gamma \cdot C} = \frac{\sum_{i=1}^{N} w_i(t) / R'}{\gamma \cdot C} = \frac{w(t) / R'}{\gamma \cdot C} = \frac{2 \cdot w(t)}{\gamma \cdot [2C \cdot (R_p + R_q) + (\zeta + 1) \cdot s]} \quad (4\text{-}11)$$

Now, substituting (4-7), (4-8), and (4-11) into Equation (4-10), we obtain

$$w(t) = \frac{2C}{2C \cdot (R_p + R_q) + (\zeta + 1) \cdot s} \cdot \left[h \cdot w(t) \cdot \left(\frac{2 \cdot \lambda \cdot C \cdot (R_p + R_q) + (\zeta + 1) \cdot s \cdot \gamma}{2 \cdot w(t)} - 1\right) + N \cdot \alpha\right] \quad (4\text{-}12)$$

Please refer Appendix B for the details of the proof. According to [XiSu05] and Appendix B, we are able to prove that the delayed differential Equation (4-12) is globally asymptotically stable and has a unique steady state equilibrium

$$w^* = \gamma \cdot \frac{2C \cdot (R_p + R_q) + (\zeta + 1) \cdot s}{2} + N \cdot \frac{\alpha}{h} \quad . \quad (4\text{-}13)$$

Let Equation (4-10) be zero at steady state, then the *cwnd* of each flow at steady state is simply Equation (4-13) divided by N, i.e.,

$$w_i^* = \frac{2C \cdot \gamma \cdot (R_p + R_q) + (\zeta + 1) \cdot s \cdot \gamma}{2N} + \frac{\alpha}{h}, \quad i \in [1, N] \quad . \quad (4\text{-}14)$$

4.2.3.2 Flow Rate

Having the steady state congestion window size of a single DCCP flow i, the flow sending rate is obviously

$$r_i^* = \frac{w_i^*}{R} = \frac{C \cdot \gamma}{N} + \frac{(\zeta - 1) \cdot s \cdot \gamma}{2N \cdot R} + \frac{\alpha}{h \cdot R} \quad . \tag{4-15}$$

where where $R=R_p+R_q+s/C$.

From the above mathematical analysis on our DCCP/VCP network model, we can obtain the following information. Equation (4-15) shows that the equilibrium sending rate depends only on the ACK ratio, the packet size, the link capacity, the target link utilization, the round trip time delay and the number of DCCP flows. We also can see that if the number of flows or the round trip time delay increase, the steady flow rate will decrease. If the link capacity, target link utilization, ACK ratio, or packet size increase, the equilibrium rate will increase. During the MI and AI phases, there is almost no packet loss, the ACK ratio of the DCCP-CCID2 will decrease to 2 and stay at this value until the first loss event happen.

4.3 Concluding Remarks

The mathematical analysis focusing equilibrium properties has been provided in this chapter. The fluid model that we used in here is a simplified fluid model. The three states (MI, AI, and MD) according to the traffic load in the VCP router is considered into our model. The effect of the ACK ratio property of DCCP-CCID2 traffic flows is also included. Equation 4-15 is the most important result we get from our analysis. It can be seen that the DCCP flow sending rate in a single bottleneck DCCP/VCP network can be approximated by our analysis. Our simulation performance evaluation in Chapter 5 will help to illustrate these.

Chapter 5
Performance Evaluation in Wired Environment

This chapter uses simulation to evaluate the performance of DCCP/VCP under the wired network environment. The main goal of this evaluation is to verify the correctness of the DCCP/VCP congestion control algorithm. Different performance measures and relevant parameters are given. Comparisons are made to other related congestion control algorithm to highlight the metrics of our control strategy.

5.1 Simulation Environment

The network simulator OPNET Modeler [Opne05] Version 10.5 is used to conduct all the experiments and simulations in this thesis. OPNET modeler is a hierarchical simulation tool. The simulation is conducted hierarchically, starting with the Network Model, and downward to the Node Model, the Process Model, and finally the Packet Format Model. Details can be found in Section 3.5 and Appendix D.

The simulation is running in the Simulation Sequence Model. All the simulation parameters can be promoted up to this level and reset in the beginning of every simulation run. The file name can be set in here to extract the simulation result. After finishing the simulation run, the simulation result can be viewed in the Analysis Configuration Model. This model collects every simulation result ".ov" file under the file menu, and then it can output and draw the data into figures. The data can be also extracted to a spreadsheet tool such as MS Excel.

The simulation measures in our performance evaluation are defined and measured in the Probe Model. The performance measures in this thesis are the source throughput, the buffer queue size, packet drop rate. These definitions can be found in Section 3.4.

Since the DCCP/VCP congestion control strategy is aimed at improve the flow throughput smoothness, the measure source throughput is important in this thesis to verify the DCCP/VCP network performance. Another metrics of DCCP/VCP is to stabilize the

buffer queue size as well as providing smooth source throughput. Thus, we make the measure of buffer queue size to be the second important measure to see the congestion avoidance capability of the algorithm. Finally, we measure the packet drop rate to verify the packet loss situation in the single bottleneck network.

The simulated time is 120 seconds, which has been determined to be adequate to establish reasonable steady-state results and achieve the full use of the bottleneck link utilization. Our simulation is running on a Pentium IV 1.5 GHz desktop computer with 512M memory computer with Windows XP operating system. A typical simulation run takes 800 seconds. In steady-state performance analysis, each data point in a figure is the average of 4 different simulation runs. This enabled us to obtain 95% confidence interval for the average queue size deviation measure. We have also obtained the 95% confidence interval for our simulated results. Some sample values are presented in Appendix E. One can see the values of these intervals are usually quite small compared with the mean.

Table 5-1: Propagation Delays in Wired Network

Subnet	Round Trip Propagation Delay
1	30 ms
2	50 ms
3	70 ms
4	90 ms
5	110 ms
6	30 ms
7	50 ms
8	70 ms
9	90 ms
10	110 ms

Table 5-1 presents the round trip propagation delay of each subnet ranging from 30 ms to 110 ms. This setting is used to simulate the different round trip propagation delay in the real world in order to account for different network situations between the connections. For example, there can be more gateway routers in the Subnet 10 connection than that in the Subnet 1 connection.

The parameters used in the think time probability distribution function of Equation (3-7) are set as $p_h = 0.4953916$, $p_l = 0.5046084$, $\pi_h = 1.0$, $\pi_l = 0.0245032$, $\eta_h = 1.243437$, $\eta_l = 3.252665$, and the parameters of the file length probability distribution function of Equation (3-8) are $\pi = 2190$ and $\alpha = 1.15066$. The buffer size of Router 1 in the wired network is 800 packets. All the packets have the same size of 1024 bytes.

Table 5-2: Simulation parameters in Wired Network

Parameter	Value
link capacity	45 Mbps
RED	
minimum buffer threshold	150
maximum buffer threshold	700
maximum packet drop probability	0.1
queue averaging filtering gain	0.002
VCP	
link load factor measurement interval	200 ms
link queue sampling interval	10 ms
link target utilization	98%
how fast to drain the link steady queue	0.5
how fast to probe the available bandwidth (MI)	0.25
AI parameter a	1.0
MI parameter ξ	0.0625
MD parameter β	0.875
MI scaling limiter	2.5
AI scaling limiter	10

Table 5-2 provides all the simulation configuration parameters in the wired network. To allow a fair comparison, we simulate the network performance for the control strategies under the same network configurations for the other common parameters (e.g. round trip time delay). The simulation scenario is also the same as each other.

We verified the performance of our congestion control strategy in two scenarios. The first scenario has no step change. There are always 200 long-lived sources sending packets to the router. The second scenario has a step change from 100 long-lived sources to 200 sources

during the simulation. As for the traffic control mechanisms of the congestion controller RED to be compared with VCP controller in this thesis, the detailed descriptions can be found in Appendix A1.

We make comparisons among the following few related congestion control strategies especially with our proposed DCCP/VCP under the wired network environment:

(1) TCP/RED: This is the most popular and widely adopted internet congestion control strategy nowadays.

(2) DCCP/RED: This is the new proposed DCCP-CCID2 source-based control mechanism that cooperates with AQM algorithm RED.

(3) TCP/VCP: This is the new proposed VCP router-based control mechanism. It is the most popular source-based control TCP.

(4) CCID3/RED: This is a rate-based source control cooperating with the widely adopted AQM algorithm RED. The rate-based TFRC algorithm is a research topic whose result is not as good as people expected. We only use it for comparison in this thesis.

We did not compare the control strategy CCID3/VCP. This is because CCID3 is a rate-based control algorithm. There is no such thing as congestion window in CCID3. Therefore, VCP cannot work together with CCID3.

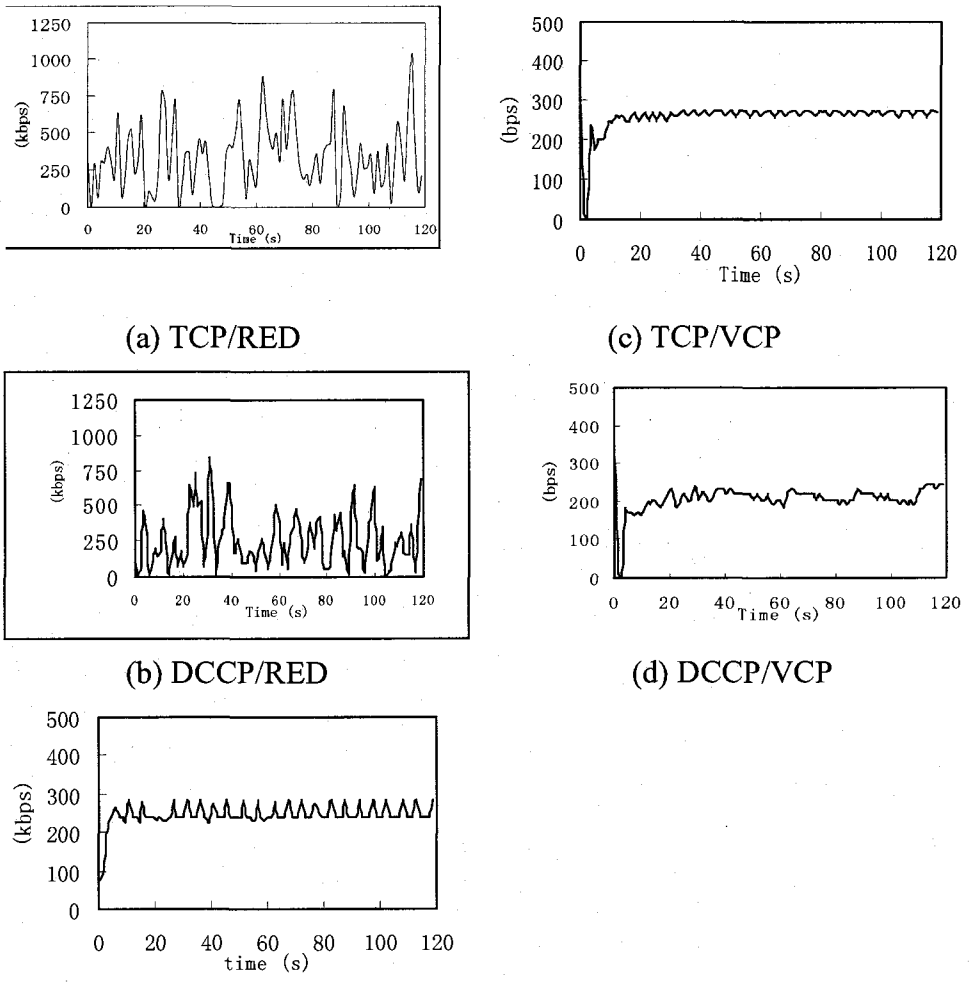
5.2 Scenario 1: No Step Change

This scenario is setup to evaluate the long term performance of the congestion strategies. We enable 200 source nodes in our bottleneck network from the beginning of the simulation to the end. All of the 200 source nodes try to send as much data as possible throughout the simulating time.

5.2.1 Source Throughput

Figure 5-1 shows that the flow throughputs of the source node from the subnet with RTPD of 30ms under TCP/RED, DCCP/RED, TCP/VCP, DCCP/VCP, and CCID3/RED respectively. The average source flow throughput is around 250 kilobits per second. This value shows our results are correct since the internet is fair to each flow [FaFl96]. There are 200 flows in total in the network. According to the fairness transmission rate property of TCP and DCCP [Stev94, Stev97, AlPa99], the average source throughput should be around the value of the

link capacity C divided by the number of flows. That is, 45,000kbps/200 = 225 kbps.



(a) TCP/RED

(c) TCP/VCP

(b) DCCP/RED

(d) DCCP/VCP

(e) CCID3/RED

Figure 5-1: Source Throughputs under Different Control Strategy (Wired)

From the results, we can see that the flow throughputs of TCP/RED (Figure 5-1a) and DCCP/RED (Figure 5-1b) are fluctuating with a very large amplitude. This is because the RED router randomly drops packets, which result in the source congestion window size is halved randomly. The fluctuation of TCP/VCP (Figure 5-1c) and DCCP/VCP (Figure 5-1d) are much less except for the beginning of the simulation. This is because all the sources send packets aggressively in the beginning of simulation, which causes many packets to be lost in the bottleneck router buffer. The TCP and DCCP sources *cwnd* halves per loss packet so that the throughput decreases immediately to a very low level. After a short period when there is

no loss event, the VCP controller starts to affect the *cwnd*. The *cwnd* increases with the MI algorithm in VCP to a comparatively high value. Then the AI and MD algorithms of VCP take the control of the source *cwnd* and operate alternatively. There are 200 short-lived sources sending packets randomly into the network. The disturbance from the short-lived sources brings the small fluctuations in Figure 5-1c and Figure 5-1d.

Using C=45, γ=0.98, N=200, ζ=2, s=1024, R=0.15, a=1, h=0.5 in Equation (4-15), we obtain a flow rate of 225 kbps. Therefore our simulation result around this value for the DCCP/VCP ascertains our previous theoretical analysis.

Figure 5-1e shows that the source flow throughput has more oscillation than that in TCP/VCP and DCCP/VCP. However, the source throughput is much smoothed by the DCCP-CCID3's TFRC rate-based control mechanism. Since the RED router randomly drops packets to achieve congestion avoidance, the packet drop rate information is fed back by the ACKs to the source nodes. The source sending rate is then adjusted by TFRC algorithm according to the packet drop rate. Thus, the oscillation in the source throughput is affected by packet loss.

5.2.2 Buffer Queue Size

Figure 5-2 shows the average buffer queue size of the bottleneck router under the four strategies. As we can see from Figure 5-2a, the buffer queue size of RED router with N = 200 TCP sources is very large. The average queue size is around 180 packets. The RED router queue size with N = 200 DCCP sources is shown in Figure 5-2b. It can be seen that the DCCP/RED buffer queue size fluctuates and its average value is around 125 packets, lower than that of TCP/RED queue. It is because of the *pipe* limitation [FlKo06b] of *cwnd* and the acknowledgement ratio mechanism in DCCP-CCID2. The cooperation of DCCP-CCID2's above new properties and the traditional RED algorithm shortens the buffer queue size a little bit.

The VCP controller further gives the 200 TCP flows a shorter buffer queue size, whose average value is around 105 packets (Figure 5-2c). However, there are more fluctuations in the buffer queue size. Figure 5-2d shows that the buffer queue size of VCP router with 200 DCCP sources is farther reduced. The buffer queue size is oscillating and the average value is the shortest with around 80 packets. The larger fluctuations in Figure 5-2c

and Figure 5-2d are caused by the interaction of the traffic load factor ρ and the short-lived sources. As a whole, VCP controller can maintain a small average queue size.

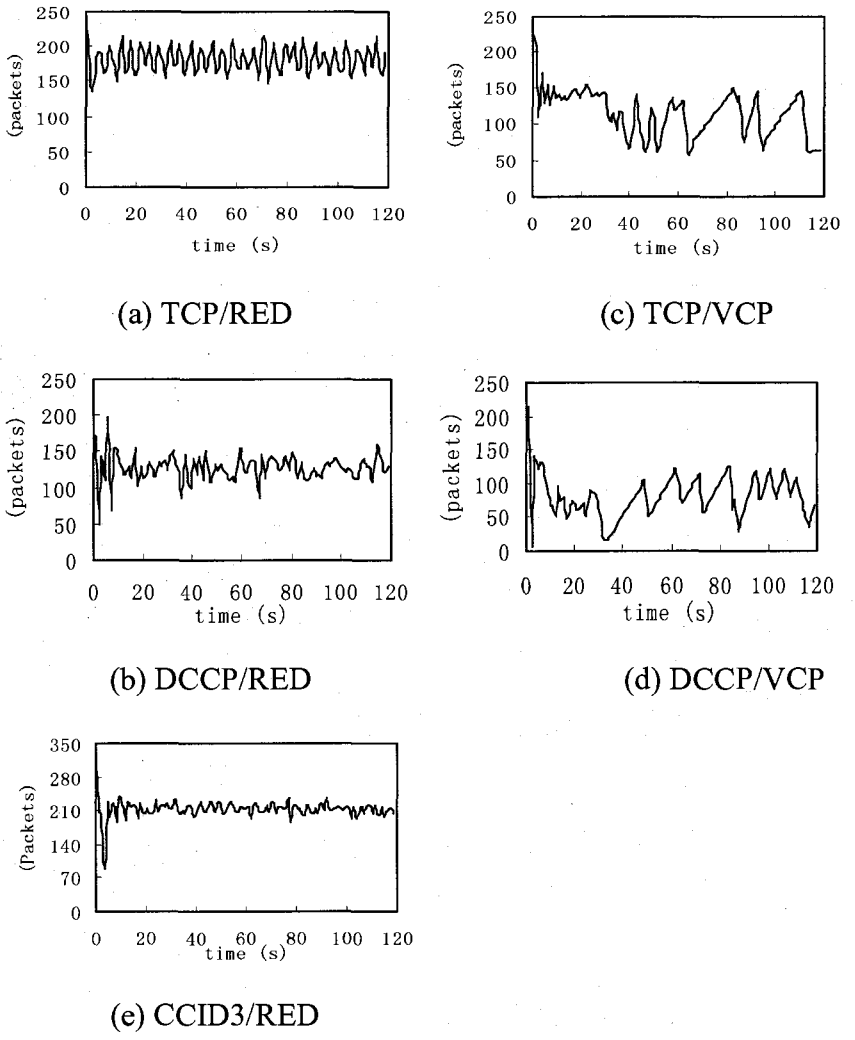


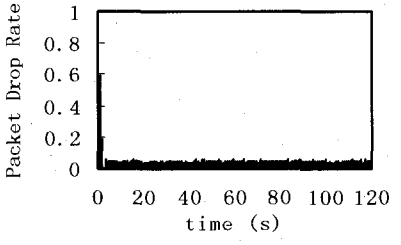
Figure 5-2: Buffer Queue Size under Different Control Strategy (Wired)

From Figure 5-2e, CCID3/RED congestion control strategy has the largest queue, which is around 210 packets. Since CCID3 is a rate-based control mechanism, its goal is to provide smooth source sending rate and friendly competing flows with TCP. However, its TFRC algorithm does not focus on stabilizing queue size. This is one of the biggest shortcomings of the TFRC algorithm, which we have discussed in Chapter 1. Obviously, our simulation results show that the buffer queue size cannot be stabilized by CCID3's TFRC algorithm. As a whole, VCP controller can maintain a small average queue size.

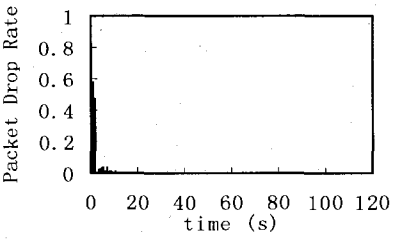
The short-lived sources influence the buffer queue size performance by randomly sending data packets. The sudden changes throughout the simulation can cause the big fluctuation of the buffer queue size. This can be seen from Figure 5-2.

5.2.3 Packet Drop Rate

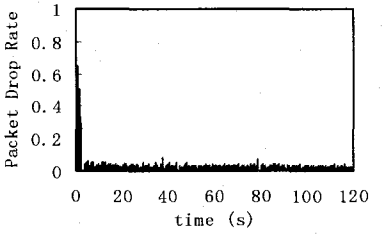
Figures 5-3 shows the packet drop rate of the bottleneck router in our network. Packet drop rate is caused by buffer overflow. It can be seen from Figure 5-3a that the RED router with 200 TCP sources keeps on dropping packets to maintain the traffic un-congested. About 7% of the packets are dropped during the simulated time period. The highest packet drop rate happens at the beginning (0^{th} second) when all the TCP sources starts to send packets to the router, after the senders realize the link is super congested, the router has already dropped nearly 70% of the TCP packets that cannot be processed.



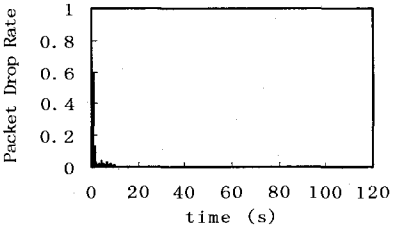
(a) TCP/RED



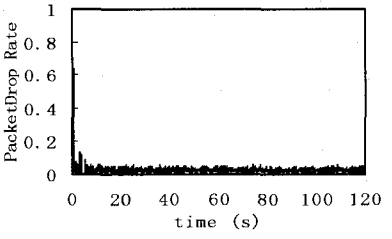
(c) TCP/VCP



(b) DCCP/RED



(d) DCCP/VCP



(e) CCID3/RED

Figure 5-3: Packet Drop Rate under Different Control Strategy (Wired)

Figure 5-3b shows the packet drop rate of RED router with 200 DCCP sources. As the control strategy, the RED router always discards 50% of the DCCP packets sent into the router in the beginning of simulation. The average packet drop rate is around 3% which is a little bit lower than that of TCP/RED. DCCP has no data retransmission when a drop event happens and CCID2's TCP-like congestion control algorithm cooperate with the ACK ratio mechanism to limit the number of dropped packets.

From Figure 5-3c and Figure 5-3d, it can be seen that the packet drop rate of VCP router with $N = 200$ TCP or $N = 200$ DCCP sources is only at a similarly high level (around 55%) with that of DCCP/RED in the beginning of simulation. Then the router reports the senders to reduce their packet sending rate. The packet drop rate starts to decrease. At the moment of the 30^{th} second (Figure 5-3c) and 10^{th} second (Figure 5-3d), the packet drop rate falls to 0, and remain there until the end of simulation, which proves that under the VCP controller there is almost no packet dropped by the bottleneck router.

Figure 5-3e shows the packet drop rate under CCID3/RED strategy with $N = 200$ DCCP-CCID3 sources. Since CCID3's TFRC algorithm is a rate-based source control mechanism, Section 5.2.2 has already shown that the rate-based control mechanism easily causes buffer overflow. The unstable buffer queue results in a very large packet drop rate (60%) at the beginning of the simulation. As time goes by, the packet drop rate decreases to 5%.

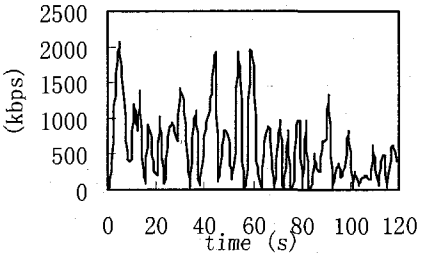
In summary, VCP algorithm is more capable in reducing packet drop rate than RED router. This is because the VCP algorithm is based on traffic load, unlike RED which is based on random dropping packets. Furthermore, the VCP algorithm divides the traffic situation into three load regions (as discussed in Section 3.3). The source congestion window size is changed according to the three link traffic load regions. The target link utilization γ also helps the bottleneck router avoid congestion to happen.

5.3 Scenario 2: With a Step Change

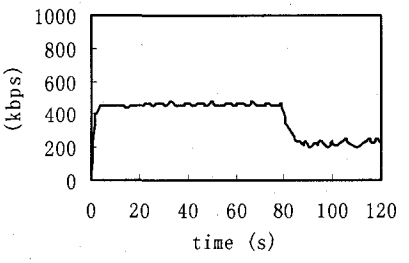
In scenario 2, we only enable 100 source nodes in our bottleneck network at the beginning of the simulation. The other 100 source nodes will enter the network from the 80^{th} second of the simulating time. There are 200 active traffic flows from the 80^{th} second to the 120^{th} second.

5.3.1 Source Throughput

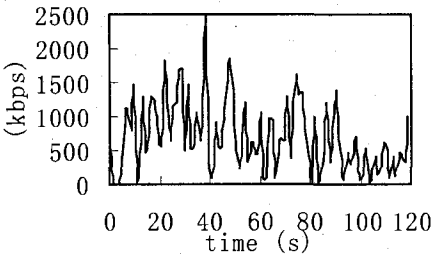
Figure 5-4 shows the source throughput under TCP/RED, DCCP/RED, TCP/VCP, DCCP/VCP, and CCID3/RED, respectively. There are 100 TCP (Figure 5-4a) and 100 DCCP (Figure 5-4b) source nodes sending data packets before the 80th second. The source throughputs of TCP and DCCP sources under RED are fluctuating around 480 kilobits per second from the 0th second to the 80th second. At the 80th second, the source throughput decreases to around 240 kilobits per second when there are other 100 source nodes sending data packets.



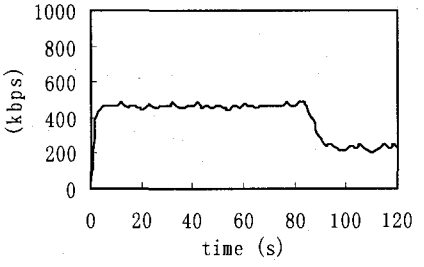
(a) TCP/RED



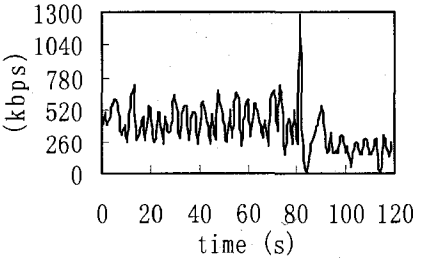
(c) TCP/VCP



(b) DCCP/RED

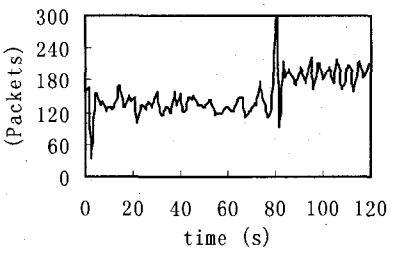


(d) DCCP/VCP

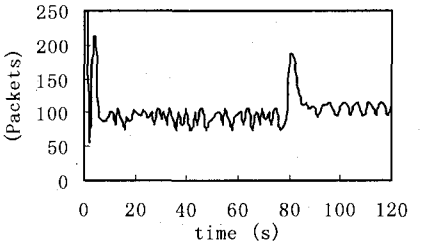


(e) CCID3/RED

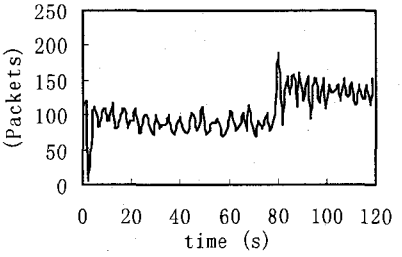
Figure 5-4: Source Throughput under Different Control Strategy (Wired-S2)



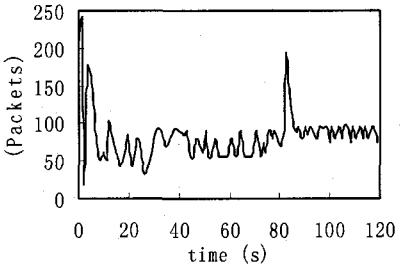
(a) TCP/RED



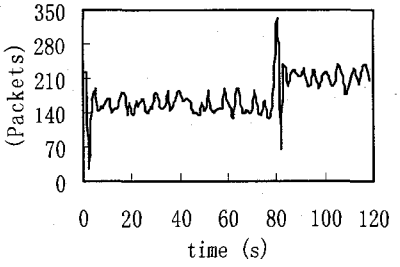
(c) TCP/VCP



(b) DCCP/RED



(d) DCCP/VCP



(e) CCID3/RED

Figure 5-5: Buffer Queue Size under Different Control Strategy (Wired-S2)

The source throughputs under VCP algorithm are much smoother than that under RED algorithm. As can be seen from Figure 5-4c and Figure 5-4d, the source throughputs are around 450 kilobits per second between the 0^{th} second and the 80^{th} second. When the other 100 sources start to send packet at the 80^{th} second, the flow throughput decreases to around 225 kilobits per second at about the 88^{th} second. The flow throughputs are still smooth till the end of simulation.

As the previous 4 control strategies, the source throughput under CCID3/RED are around 450 kilobits per second before the other 100 sources enter the network. After the 80^{th}

second, the source throughput reduces to about 225 kilobits per second. The source throughput under CCID3/RED is fluctuating less than that under TCP/RED and DCCP/RED. However, the amplitude is larger than that under VCP. This is caused by the packet loss in the RED router.

5.3.2 Buffer Queue Size

Figure 5-5 shows the buffer queue size under different control strategies in scenario 2. As we can see in the 5 figures, the buffer queue sizes before the 80th second are smaller than buffer queue sizes after. When the additional 100 sources enter that network at the instant of the 80th second, the buffer queue sizes of all the 5 control strategies suddenly increase a lot. After a period of 3 seconds, all the queue sizes (Figure 5-5a, b, c, d, and e) reduce to a larger size than the queue size before the 100 sources enter the network. One thing to notice is that, the queue size under VCP is still smaller than that under RED algorithm. This is because the VCP algorithm uses the traffic load to adjust the source throughput to maintain a short buffer queue size.

5.3.3 Packet Drop Rate

The packet drop rate under the 5 control strategies are shown in Figure 5-6. As can be seen in Figures 5-6a, 5-6b, and 5-6e, the packet drop rate in the RED router is around 4% before the 80th second. At the moment when the 100 new sources enter the network, the packet drop rate suddenly increases to a high level (about 13%), then it decreases to around 7%, which is still higher than before.

The packet drop rate is high (30-35% in Figure 5-6c and 28% in Figure 5-6d) at the beginning of the simulation. It decreases to 0 very quickly. However, when the 100 new source nodes start to send data packets to the network at the 80th second, the packet drop rate increases to around 30% (40% in Figure 5-6d) again, then it decreases back to 0 after 5 seconds. There is almost no packet dropped by the VCP router.

In summary, DCCP/VCP provides quick response to adjust the source throughput when there is a step change in the network. DCCP/VCP also maintains a comparatively short buffer queue size almost without packet loss.

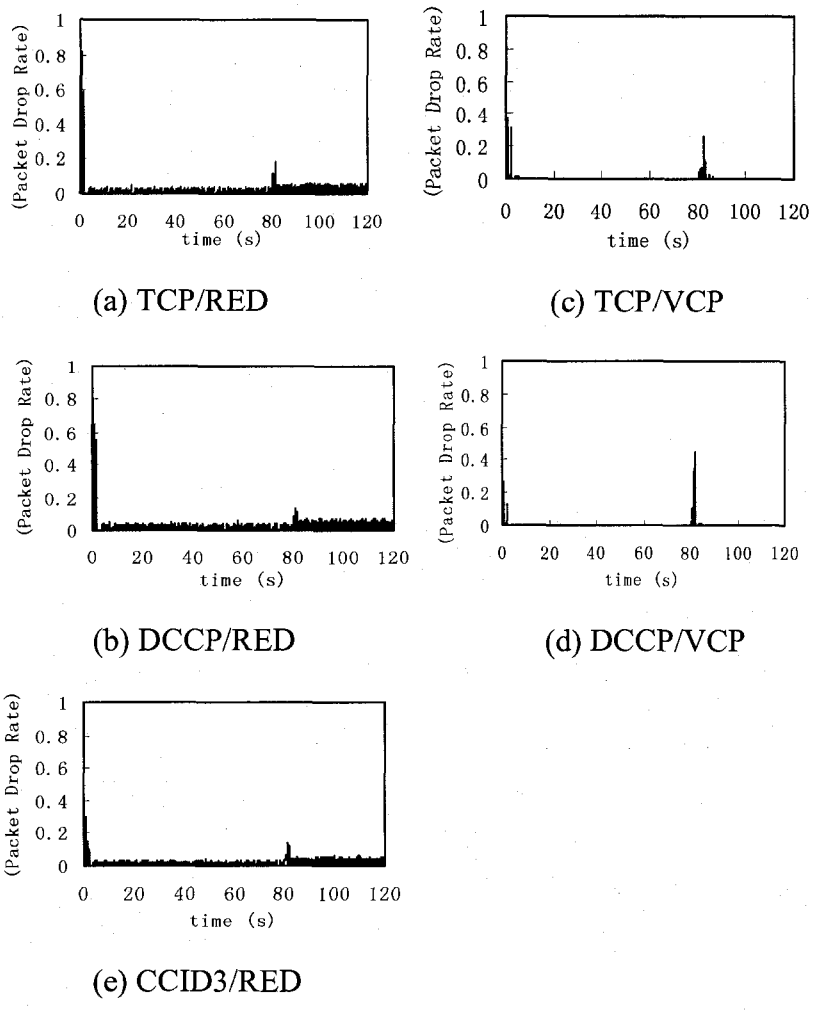


Figure 5-6: Packet Drop Rate under Different Control Strategy (Wired-S2)

5.4 Effect of Round Trip Time Delay

In this section, we are interested in investigating the effect of varying round trip time on the performance of the 5 control strategies in wired network. The bottleneck link capacity is 45Mbps, and there are 200 long-lived wired traffic flows throughout the simulation. The average round trip time delay of the traffic flows traversing the bottleneck router is varying from 0.1 second to 0.5 seconds by changing propagation delay. This is to simulate the variation of the propagation delay caused by the various network conditions in the real world. The change of *RTT* can be caused by a sudden change of the environment (e.g. the change of temperature can cause the propagation delay of the cable changed) such as using OSPF (Only Shortest Path First) routing protocol. A packet will be routed to the possible shortest path through a series of AQM routers. However, if one router is crashed, the previous router will

try to find a new router to route the packets. Some of the related work can be found from [HoYa06]. Such as the average buffer queue size and packet drop rate are measured in this section.

Figure 5-7 shows the effect of varying round trip time delay on the average buffer queue size of the bottleneck router in wired environment. As we can see, when the round trip time delay increases, the average buffer queue size decreases under all of the 5 control strategies. However, CCID3/RED's TFRC algorithm has the largest buffer queue size among all the control strategies. The buffer queue sizes under TCP/RED and DCCP/RED are slightly lower. VCP algorithm has the shortest buffer queue size. This can be explained by the fact that as the round trip time delay increases, the ACKs take more time to report the congestion feedback to the source nodes. The congestion window size is increased more slowly. The source throughput increases slowly, the AQM algorithm has already shorten the buffer queue size when the source nodes receive the control information. There are many packets still traveling in the cable but not staying in buffer queue in the router. At this point, the buffer queue size of the bottleneck router becomes lower.

The effect of varying round trip time delay on the packet drop rate under 5 control strategies in wired network is shown in Figure 5-8. From the result in Figure 5-8, we can see that there is almost no packet loss in VCP router in either TCP or DCCP traffic flows due to the different average RTT values. We say average RTT here is because the RTT value for each subnet is different. When the average RTT increases, fewer packets are dropped by the RED router. This result matches the buffer queue size result since the probability of dropping packet is greater when the buffer queue size is larger.

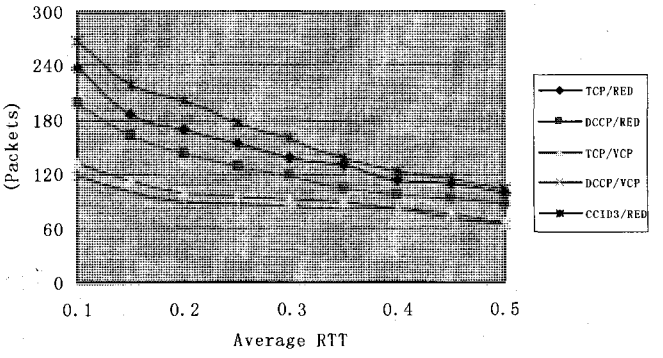


Figure 5-7: Effect of RTT on Average Buffer Queue Size (Wired)

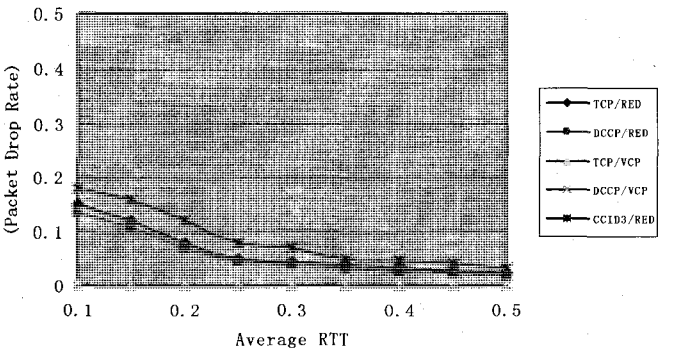


Figure 5-8: Effect of RTT on Packet Drop Rate (Wired)

DCCP/VCP gives the shortest buffer queue size when varying the round trip time delay. As the round trip time delay is changing, there is still no packet loss in VCP router.

5.5 Concluding Remarks

This chapter provides the simulation configuration and the simulation results of three measures. Allowing a fairness comparison with the same simulation configurations, same RED and VCP parameters, and same network scenarios, we made performance evaluation and comparison under different congestion control strategies. From the simulation results in the wired environment, DCCP/VCP has strong capability in handling step changes as well as varying round trip time. The simulation results of DCCP/VCP source throughput verify that our mathematical analysis result in Equation (4-15) is correct.

Chapter 6
Performance Evaluation in WiMAX Environment

As mentioned in Chapter 1, WiMAX with its broad bandwidth and high mobility support is becoming one of the most promising wireless applications nowadays. In this chapter, we will study the performance of our proposed DCCP/VCP congestion control strategy, as well as performance comparison with the other control strategies under WiMAX environment.

6.1 Simulation Environment

We conduct our performance evaluation via simulations using the OPNET simulator. The details of the OPNET models have been discussed in Section 3.5. The Packet Format Model, Simulation Sequence Model, Analysis Configuration Model, and Probe Model are the same as those in the wired environment in Chapter 5. The differences are in the WiMAX Network Model, Process Model, Pipeline Stage Model and Antenna Pattern Model. The process and functions of the WiMAX physical layer and MAC layer are developed strictly according to the WiMAX standard [IeWi05] using C++ code in the Pipeline Stage Model. The Antenna Pattern Model is used to develop an isotropic WiMAX antenna in our simulation. The isotropic antenna sends data packets and receives ACKs to (or from) the WiMAX base station with a preset antenna gain.

As an initial wireless simulation, we consider a static wireless network, i.e. no nodes are moving around. For the round trip propagation delay of each subnet, we use the same configurations shown in Table 5-1 to simulate the different round trip propagation delay in the real world. The round trip propagation delays of the subnets range from 30 ms to 110 ms. However, longer propagation delay between the WiMAX source and basestation can incur more packet loss by corruption which can cause worse performance of WiMAX source throughput. Since the maximum distance between WiMAX node and base station is less than 10 km, the maximum propagation delay from a WiMAX node to a base station is less than 10 km/$3 \cdot 10^8$ m/s = 0.03 ms, which can be ignored in here. For the short-lived sources, the

same think time distribution and the file length distribution with the same value of parameters are also used in the WiMAX environment. As shown in Table 5-2, the same RED and VCP algorithm parameters are used in WiMAX as well.

Table 6-1: WiMAX Physical Layer Configurations

Parameters	Settings
Channel Access Method	OFDMA
Modulation Scheme	QPSK
Channel Bandwidth	25 MHz
Minimum Frequency	8 GHz
Sending Power	2500 W
Maximum Bitrate	75 Mbps
Frame Length	50 ms
Slot Size	2 ms

Table 6-1 provides the WiMAX physical layer configurations in our simulation. In our simulation, we use the OFDMA channel access method which can support broader bandwidth than SC. Since the frequency bands of WiMAX can be ranged from 2GHz to 11 GHz, OPNET simulator only allow a preset fixed frequency band. The minimum frequency is selected to be 8 GHz in our simulation to ensure the correct implementation of WiMAX application. Since OPNET simulation dose not support randomly changing modulation schemes, we select to use QPSK scheme in this thesis. We select the middle value of WiMAX channel bandwidth which is 25 MHz in our simulation. For the transmission power, we choose the value of 2500 W which is within the range of WiMAX transmission power. The maximum bitrate is a parameter in OPNET wireless model that constrains the source sending rate. We use the maximum value of WiMAX sending rate 75 Mbps to be the maximum bitrate in our simulation. Notably, this value can be only achieved in the ideal state and is not expected to be achieved in our simulation.

The difference between WiMAX and mixed Wired/WiMAX networks are in the composition of the networks as shown in Table 6-2 above. We set all source nodes in the subnets to be WiMAX nodes in the WiMAX network. However, in the deployment of real

WiMAX network, the WiMAX traffic and the wired traffic can share the bandwidth of a bottleneck router. To simulate this scenario, we setup the mixed network which has 5 WiMAX subnets and 5 wired subnets.

Table 6-2: Subnet Difference Between WiMAX and Mixed Networks

	WiMAX Network	Mixed Network
Subnet 1	WiMAX	WiMAX
Subnet 2	WiMAX	WiMAX
Subnet 3	WiMAX	WiMAX
Subnet 4	WiMAX	WiMAX
Subnet 5	WiMAX	WiMAX
Subnet 6	WiMAX	Wired
Subnet 7	WiMAX	Wired
Subnet 8	WiMAX	Wired
Subnet 9	WiMAX	Wired
Subnet 10	WiMAX	Wired

Similar to the evaluation done in wired network to allow a fair comparison, we simulate the network performance for different control strategies under the same network configurations for the other sharing parameters.

To validate the performance of the DCCP/VCP controller, we compare its OPNET simulation results in both WiMAX and Mixed (Wired/WiMAX) networks with those of TCP/RED, DCCP/RED, TCP/VCP, and CCID3/RED. In evaluating the controllers in the WiMAX and the mixed Wired/WiMAX networks, we use the same three performance measures (source throughput, buffer queue size, and packet drop rate) that are used to evaluate the DCCP/VCP performance in the wired network. The simulation measures source throughput is taken from the WiMAX source node, while the buffer queue size and packet drop rate are measured in the bottleneck router (Router 1).

All network systems are simulated for a time period of 120 seconds to achieve the steady-state and the full use of the link utilization. Our simulation is running on a Pentium IV CPU computer with Windows XP operating system. A typical simulation run takes 980

seconds. To verify our code is correct, we have double checked our code against the operation of the MAC layer and physical layer in WiMAX standard [IeWi05]. From the cooperation of each of the transition states in the process model, we know that our code is running correct based on the WiMAX standard. The WiMAX throughput is actually not smooth. The length of downlink and uplink subframes in a TDD frame is adaptively changing all the time. This also causes the fluctuation of the WiMAX source throughput.

The following sections provide the simulation results in both WiMAX and Mixed (Wired/WiMAX) networks and the discussions regarding to the network performance. We made performance comparisons between different related congestion control strategies: TCP/RED, DCCP/RED, TCP/VCP, DCCP/VCP, and CCID3/RED. The rationale of using these combinations can be found in Section 5.1.

For the traffic control mechanisms of the congestion controller RED to be compared with VCP controller one can find the detailed descriptions in Appendix A1.

6.2 WiMAX Networks

We first evaluate the performance of the network with all WiMAX subnets. As we have mentioned in Section 6.1, there are 400 WiMAX source flows in total traversing a bottleneck router. Half of the WiMAX flows are long-lived flows, while the other half are short-lived flows (to be the disturbance in the real world). The performance evaluation is made between the same congestion control strategies that are used in the wired network.

6.2.1 Scenario 1: No Step Change

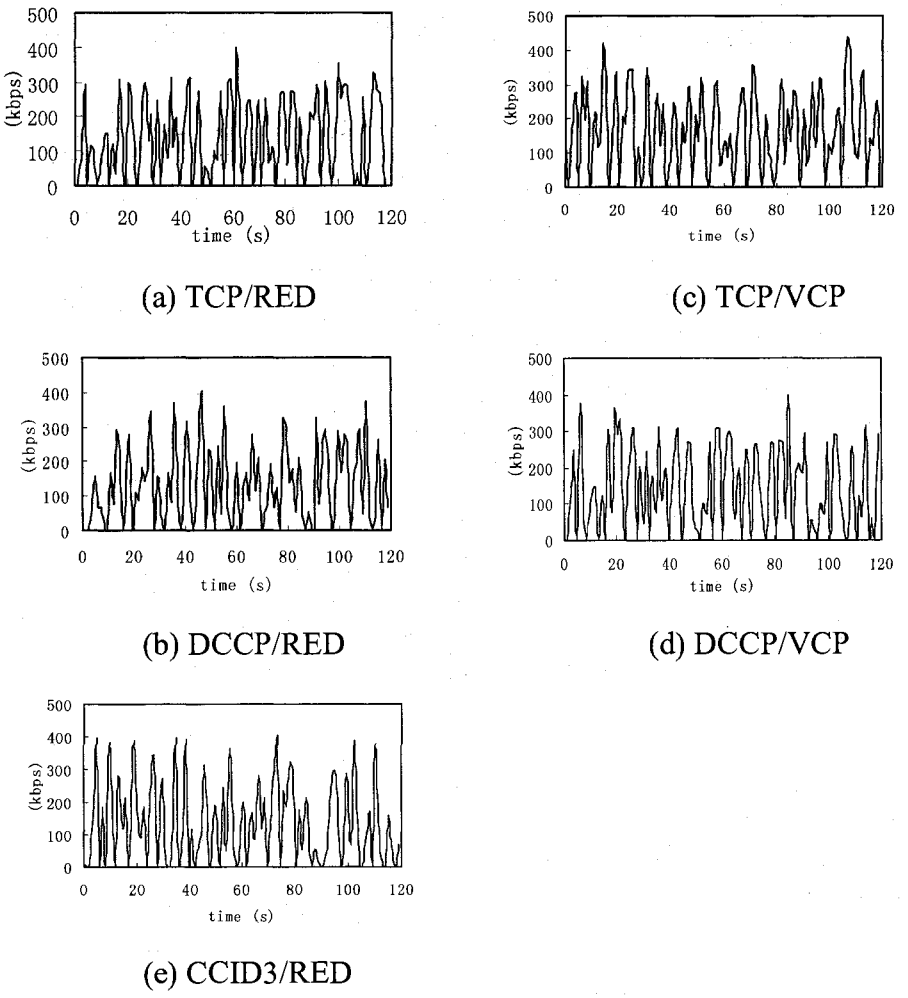
Like the wired environment, in scenario 1, we enable all the 200 source nodes in our bottleneck network from the beginning of the simulation to the end. All of the 200 source nodes try to send as much data as possible throughout the simulating time.

6.2.1.1 Source Throughput

Figures 6-1a to 6-1e show the WiMAX source throughputs under TCP/RED, DCCP/RED, TCP/VCP, DCCP/VCP, and CCID3/RED congestion control strategies respectively. In order to allow a fair comparison and investigate the source throughput performance of WiMAX

applications, all the source throughputs under the five control strategies are measured from the same WiMAX source node (Node 1), whose RTPD is around 30ms.

We can see from Figures 6-1a and 6-1b that the RED algorithm results in quick fluctuation in the WiMAX source throughputs. The average throughputs under TCP/RED and DCCP/RED are around 200,000 bits per second. The RED algorithm randomly drops data packets to avoid congestion, which brings the source throughput big fluctuations. In addition, the packet loss caused by the wireless channel corruption even makes the source throughput worse. We can see that the source throughput sometimes can drop to about 0 bits per second.



(a) TCP/RED

(c) TCP/VCP

(b) DCCP/RED

(d) DCCP/VCP

(e) CCID3/RED

Figure 6-1: Source Throughput under Different Control Strategy (WiMAX)

When we change the algorithm to VCP, as can be seen from Figure 6-1c and 6-1d, the WiMAX source throughput still fluctuates with a large amplitude. It can range from 0 bits

per second to 400,000 bits per second. This is caused by the unexpected packet drop due to wireless channel fading. In VCP algorithm, although the router avoids congestion according to the traffic load, the data packets can be dropped due to the corruption such as wireless multipath fading or weather reasons. When there is either a congestion loss or corruption loss, the router will acknowledge the WiMAX source. The *cwnd* is halved by the source node per one packet loss. Thus, the wireless source throughput is hardly smoothed by the VCP algorithm under WiMAX environment.

Figure 6-1e shows the WiMAX source throughput under CCID3/RED congestion control strategy. The source throughput is fluctuating like the other congestion control strategies.

6.2.1.2 Buffer Queue Size

Figure 6-2 shows the buffer queue size of the bottleneck router under the five congestion control strategies. We are interested in investigating the ability of the five strategies of stabilizing the buffer queue.

The TCP/RED (Figure 6-2a) strategy gives a very big average buffer queue size (330 packets), while the average queue size under DCCP/RED (Figure 6-2b) strategy is about 310 packets. Like the wired network, the advantage of DCCP/RED is taken from the DCCP-CCID2 algorithm's *pipe* and ACK Ratio properties cooperation. However, RED algorithm still cannot stabilize the bottleneck router's buffer queue size very well.

From Figure 6-2c and Figure 6-2d, we can see that the VCP algorithm performs well and largely reduces the average buffer queue size to 85 packets for TCP/VCP (Figure 6-2c) and 75 packets for DCCP/VCP (Figure 6-2d). This is because the VCP algorithm achieves congestion avoidance according to the traffic load in the bottleneck router. Unlike RED, which is based on randomly drop packets, the target link utilization γ avoids the bottleneck link utilization from growing up to 1. Thus, the buffer queue size of the bottleneck router can be maintained to be stable. The cooperation of DCCP-CCID2's ACK Ratio and VCP algorithm also makes the buffer queue size shorter than that under TCP/VCP. Buffer overflow is well avoided by DCCP/VCP in WiMAX environment.

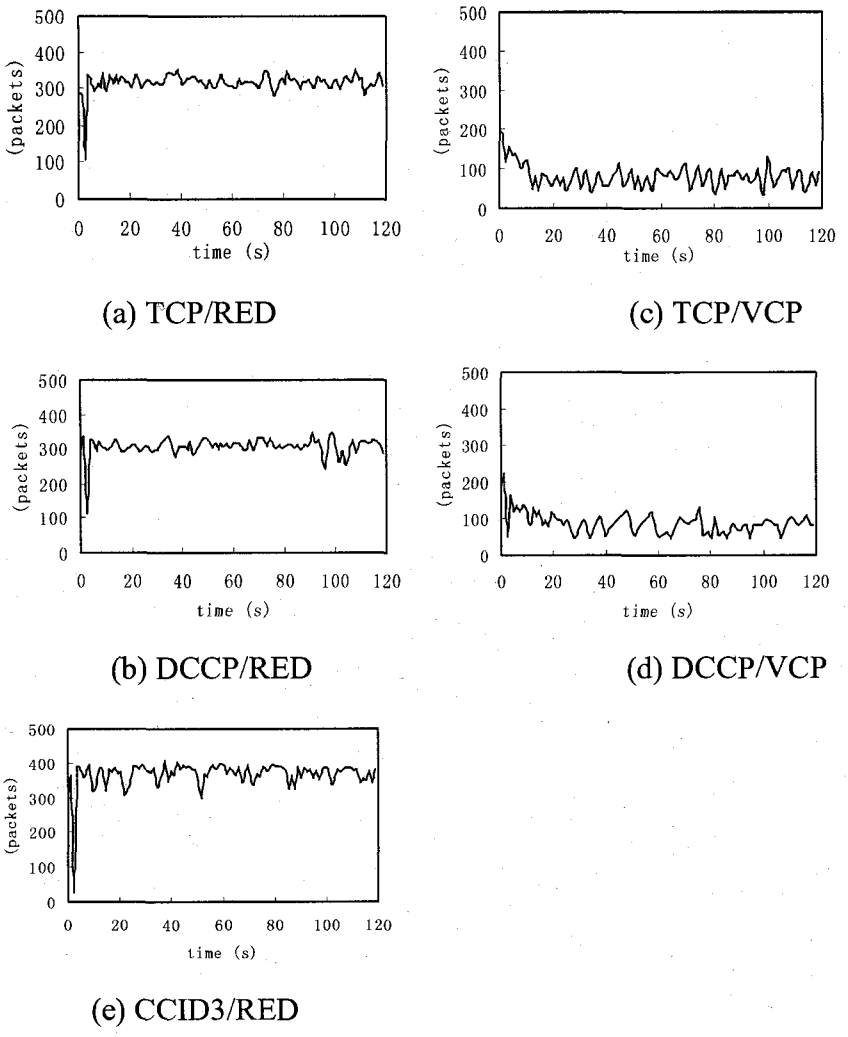


Figure 6-2: Buffer Queue Size under Different Control Strategy (WiMAX)

Figure 6-2e shows the buffer queue size under CCID3/RED congestion control strategy. The buffer queue size is the largest around all of the five control strategies, which is about 370 packets. This result shows that CCID3/RED gives poor performance on maintaining a short buffer queue size.

Notably that the fluctuation of the buffer queue size in WiMAX network is more than that in wired network. This is because there are much more packet loss in wireless channel than that in the wired network. Packet loss may cause the sudden change of the *cwnd* in source node, which may cause the bad performance of the queue.

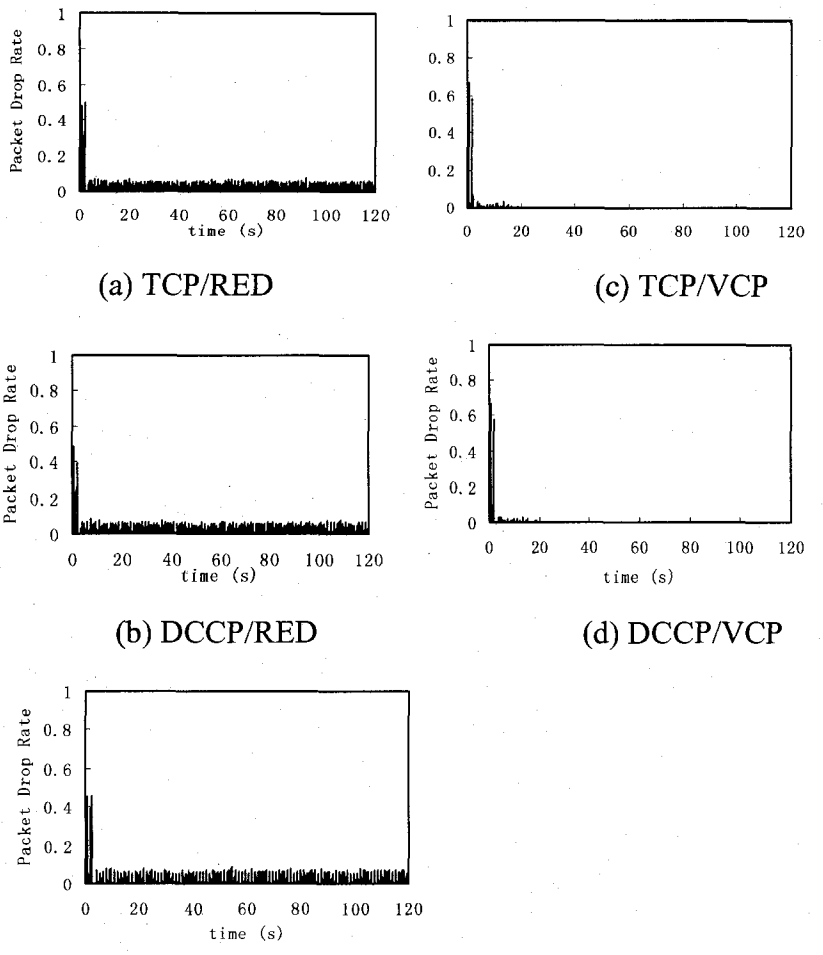


Figure 6-3: Packet Drop Rate under Different Control Strategy (WiMAX)

6.2.1.3 Packet Drop Rate

Figure 6-3 shows the packet drop rate of the bottleneck router under the five congestion control strategies under WiMAX environment. There are two types of packet loss in wireless networks, congestion packet loss and wireless channel corruption packet loss. One thing to note is that the packet drop rate here is only caused by the congestion, not by the wireless channel degradation.

As can be seen from Figure 6-3a and Figure 6-3b, the packet drop rate is very high at the beginning of the simulation. After about 3 seconds, the packet drop rate is reduced and remained at around 7% by TCP/RED (Figure 6-3a) strategy. The DCCP/RED (Figure 6-3b) strategy slightly reduces the packet drop rate to 5%. As shown in Figure 6-2, the buffer queue size under DCCP/RED is shorter than that under TCP/RED. There are fewer packets dropped

by a shorter buffer queue, which is correct.

From Figure 6-3c and Figure 6-3d, we can see that, there is almost no packet dropped due to congestion in TCP/VCP (Figure 6-3c) and DCCP/VCP (Figure 6-3d). At the moment of 18^{th} second (10^{th} second for DCCP/VCP), the packet drop rate reduced to 0 under both strategies. The packet drop rate performance also reflects the correctness of the buffer queue size performance.

Figure 6-3e shows the packet drop rate under CCID3/RED congestion control strategy. Most data packets (around 50%) are dropped by the RED router in the beginning of the simulation. However, it reduces fast to about 8% and stays at this level from t=5s. Thus the rate-based control mechanism CCID3 is not good at stabilizing buffer queue size and avoiding packet loss.

The results of Figure 6-3 show that although the wireless environment may cause unexpected packet loss, which may bring arbitrary halving on *cwnd*, VCP still works well on congestion control and ensures no packet dropped by the congested bottleneck router.

Notably that the packet drop rate is more than that in wired network. The unexpected packet loss in the wireless channel causes the bad performance of the RED algorithm. However, the VCP algorithm which uses the traffic load to avoid congestion performs as well as that in wired network.

From the simulation results in WiMAX application, although the changing wireless link capacity and the packet dropped by corruption make the DCCP source throughput is not as smooth as it is under the wired VCP controller, DCCP/VCP still works well on avoiding buffer overflow, thus protecting packets from being dropped by the congested router.

6.2.2 Scenario 2: With a Step Change

Like the wired environment, in scenario 2, we only enable 100 source nodes in our bottleneck network at the beginning of the simulation. The other 100 source nodes will enter the network from the 80^{th} second of the simulating time. There are 200 active traffic flows from the 80^{th} second to the 120^{th} second.

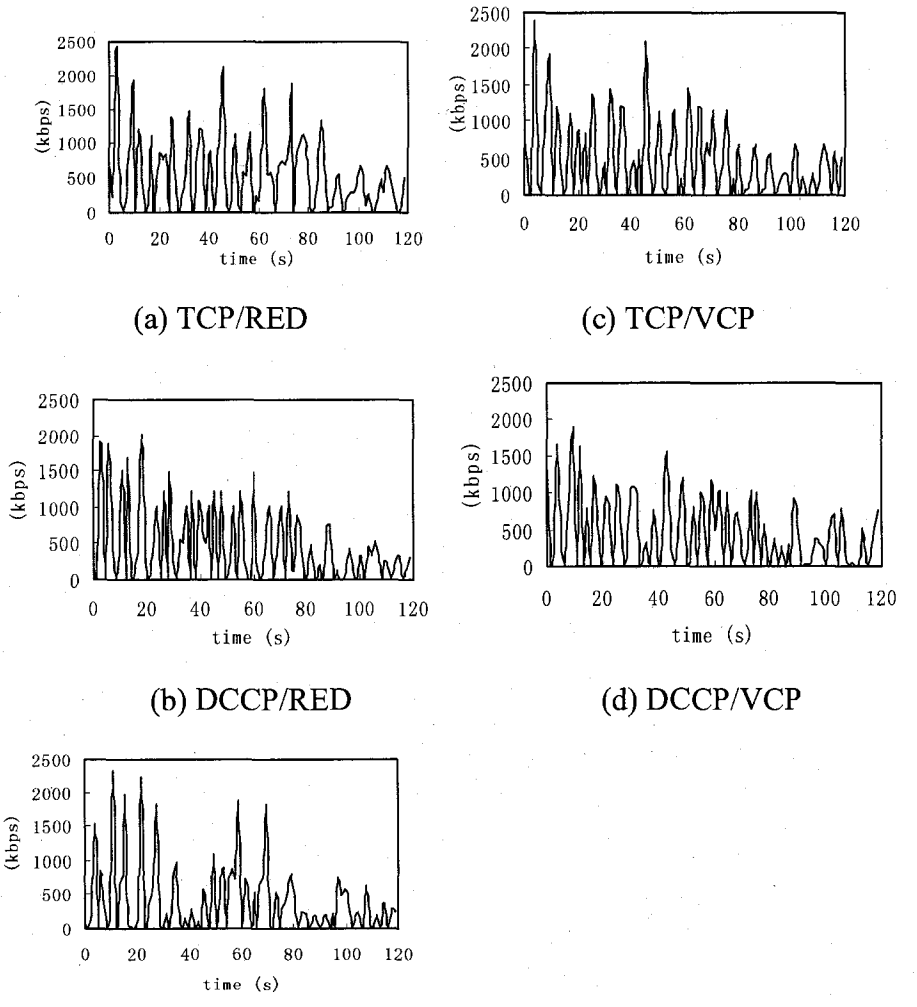
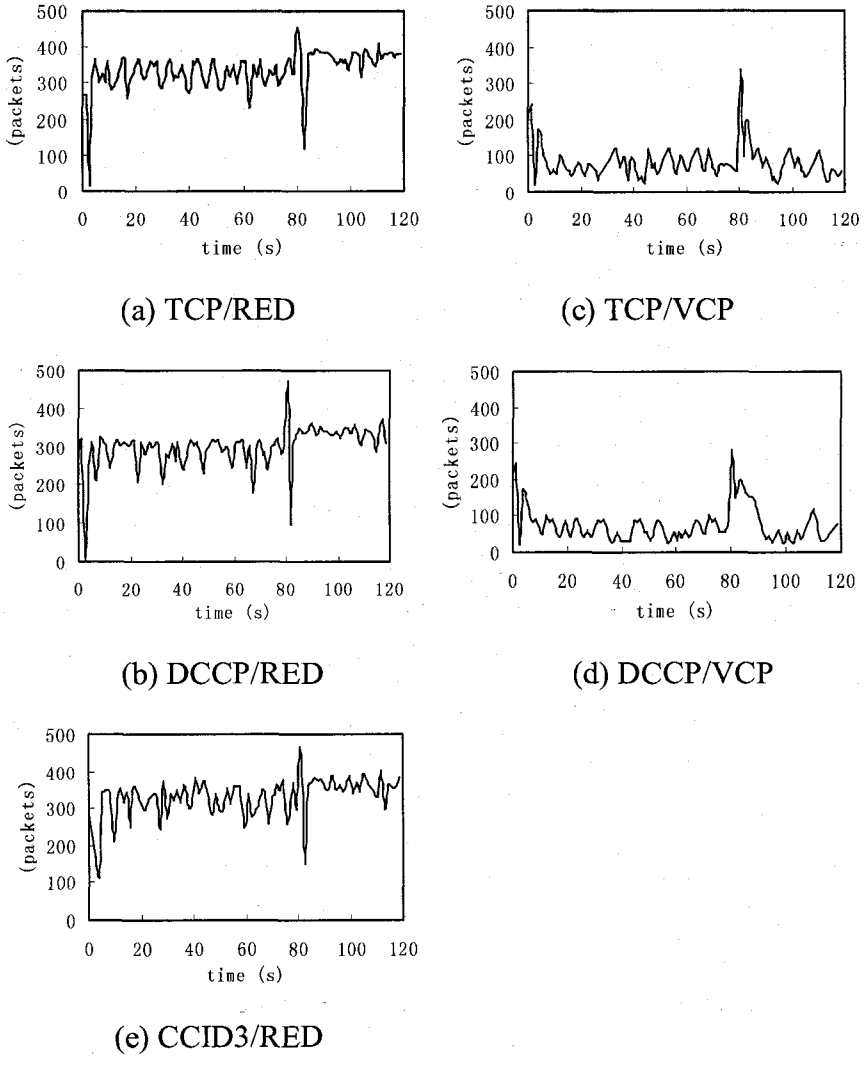


Figure 6-4: Source Throughput under Different Control Strategy (WiMAX-S2)

6.2.2.1 Source Throughput

The source throughputs under TCP/RED, DCCP/RED, TCP/VCP, DCCP/VCP, and CCID3/RED are shown in Figure 6-4. We can see the throughputs are fluctuating under all of the control strategies. As in Scenario 1, VCP algorithm cannot provide a smooth source throughput in WiMAX environment. From the start of the simulation to the 80th second, the source throughputs are around 450 kilobits per second. This is reasonable. Since there are originally 100 source nodes sending packets to the network. The link capacity of the bottleneck is 45Mbps. The network is fair to all the flows traversing the bottleneck router. The flow rate of each flow should be around 45Mbps/100, which is 450 kilobits per second.

After the 100 new flows enter the network, there are 200 flows in total sending data packets to the bottleneck router. The average flow rate of each flow should be around 45Mbps/200, which is 225 kilobits per second. This also can be seen from Figure 6-4 after the 80th second.



(a) TCP/RED (c) TCP/VCP

(b) DCCP/RED (d) DCCP/VCP

(e) CCID3/RED

Figure 6-5: Buffer Queue Size under Different Control Strategy (WiMAX-S2)

6.2.2.2 Buffer Queue Size

Figure 6-5 shows the buffer queue sizes under the 5 control strategies in WiMAX environment scenario 2. Comparing with the 5 figures, VCP algorithm provides the TCP (Figure 6-5c) and DCCP (Figure 6-5d) traffic flow the shortest buffer queue size. The largest buffer queue size appears in the CCID3/RED strategy. As in the wired environment scenario

2, the buffer queue size has increased at the 80th second. Then it reduces to a higher level than that before the 80th second.

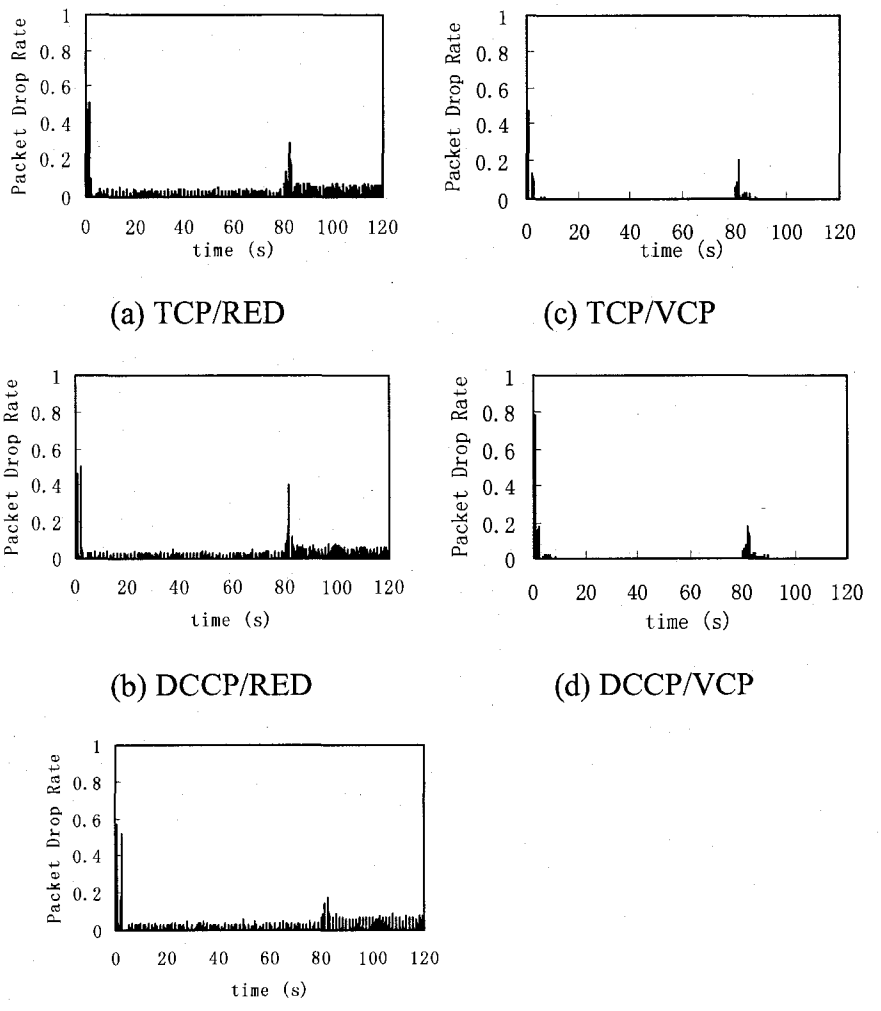


Figure 6-6: Packet Drop Rate under Different Control Strategy (WiMAX-S2)

6.2.2.3 Packet Drop Rate

The packet drop rate under the 5 control strategies is shown in Figure 6-6. As we can see from Figure 6-6a, b, and e, RED router always tries to drop data packets throughout the simulation. About 5% data packets are dropped by the RED router before the 100 new flows enter the network. After the 80th second, there are 200 active traffic flows in the network. The packet drop rate is higher than before.

However, the VCP avoids dropping data packet. Even though the 100 new flows cause a sudden packet drop at the 80th second, the VCP router reduces the packet drop rate to 0 again at the 90th second. There is almost no packet loss in VCP router.

6.2.3 Effect of Round Trip Time Delay

In this section, we are interested in investigating the effect of varying round trip time delay on the performance of the 5 control strategies for a WiMAX network. The basic setting of our bottleneck is a 45Mbps link capacity with 200 long-lived WiMAX traffic flows in the simulation. As in the wired network, the average round trip time delay of the traffic flows traversing the bottleneck router is varying from 0.1 second to 0.5 seconds. The average buffer queue size and packet drop rate are measured in this section.

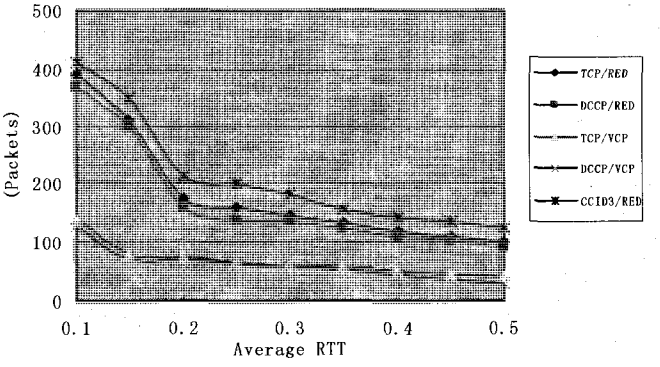


Figure 6-7: Effect of RTT on Average Buffer Queue Size in WiMAX Network

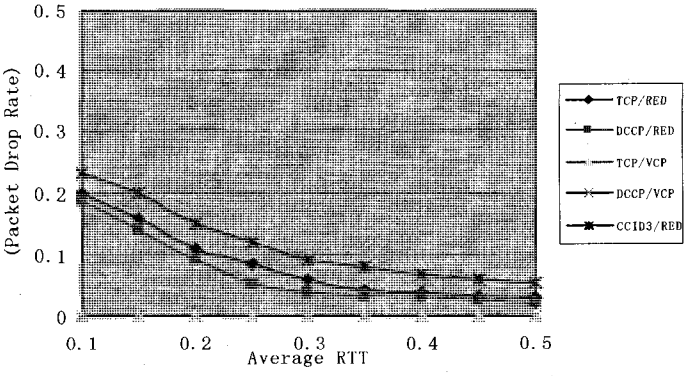


Figure 6-8: Effect of RTT on Packet Drop Rate in WiMAX Network

Figure 6-7 shows the effect of round trip time delay on the average buffer queue size under the 5 control strategies in WiMAX network. As a result of varying RTT from 0.1

second to 0.5 seconds, the buffer queue size of the bottleneck router becomes lower. We can see that the TCP/VCP and DCCP/VCP control strategies give the lowest buffer queue size. It can be ranging from about 150 packets to 50 packets when changing the RTT. CCID3/RED gives the longest buffer queue size which can be ranging from 420 packets to 130 packets when the RTT is changed. The buffer queue sizes under TCP/RED and DCCP/RED can be ranging from around 400 (380 for DCCP/RED) packets to 100 (95 for DCCP/RED) packets when the RTT changes. Another observation is, when the RTT is small, the deviation of buffer queue size between RED and VCP router is large. When the RTT is large, the difference of RED and VCP is small. VCP router can provide the shortest buffer queue size than RED router.

Figure 6-8 shows the effect of RTT on packet drop rate under the 5 control strategies in WiMAX network. We can see that the packet drop rate under VCP algorithm is always zero when the RTT changes. However, RED algorithm gives a relatively big buffer queue size when the RTT is small. As the RTT becomes smaller, the packet drop rate decreases.

6.3 Mixed Networks

In this section, we evaluate the performance of the network with five WiMAX subnets and five wired subnets. Like the WiMAX network, half of the Wired/WiMAX source flows are long-lived flows, while the other half are short-lived flows to simulate the real world network disturbance. The performance evaluation is made between the same congestion control strategies that are used in the wired and WiMAX network. Again, in order to study the performance of WiMAX application, the source throughput is taken from the same WiMAX source nodes (Node 1) in the source subnet whose RTPD is around 30ms.

6.3.1 Scenario 1: No Step Change

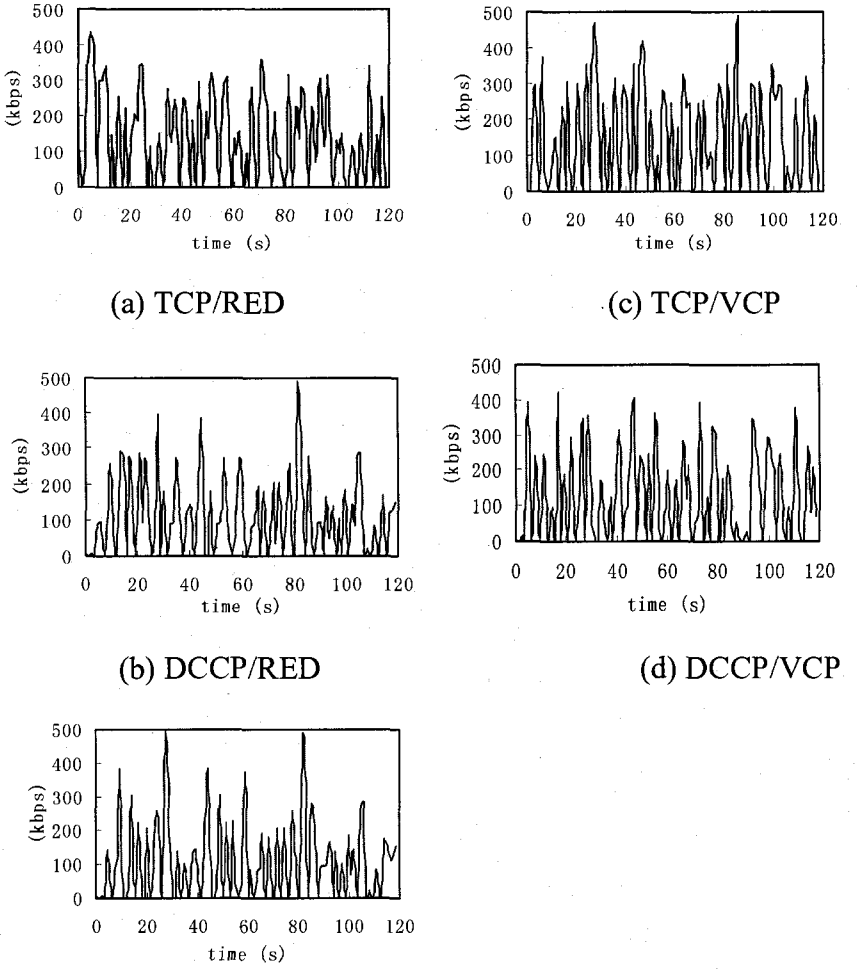
Like the wired and WiMAX environment, in scenario 1, we enable all the 200 source nodes in our bottleneck network from the beginning of the simulation to the end. All of the 200 source nodes try to send as much data as possible throughout the simulating time.

6.3.1.1 Source Throughput

Figure 6-9a to Figure 6-9e show the WiMAX node source throughputs under TCP/RED,

DCCP/RED, TCP/VCP, DCCP/VCP, and CCID3/RED congestion control strategies in a WiMAX/Wired mixed bottleneck network.

We can see from Figures 6-9a and 6-9b that the RED algorithm results in quick fluctuation and large amplitude in the WiMAX source throughputs. The average throughputs of the $N = 200$ TCP and DCCP sources under TCP/RED and DCCP/RED are around 200,000 bits per second. The large amplitude ranging from 0 to 400,000 bits per second is caused by the RED algorithm and the corruption of wireless link such as multipath fading.



(a) TCP/RED

(c) TCP/VCP

(b) DCCP/RED

(d) DCCP/VCP

(e) CCID3/RED

Figure 6-9: Source Throughput under Different Control Strategy (Mixed)

As can be seen from Figure 6-9c and 6-9d, the VCP algorithm does not smooth the WiMAX source throughput as well as it is in the wired environment. Like the source throughput performance in WiMAX network, the WiMAX source throughput fluctuates with

large amplitude which ranges from 0 to 400,000 bits per second. This is caused by the unexpected packet drop due to wireless channel fading. In VCP algorithm, although there is almost no packet dropped by congestion reason, the data packets can be dropped due to the corruption such as wireless multipath fading or weather reasons. The *cwnd* is halved by the router when there is either a congestion loss or corruption loss. Thus, the wireless source throughput is hardly smoothed by the VCP algorithm under WiMAX environment.

Figure 6-9e shows the WiMAX source throughput under CCID3/RED congestion control strategy. The rate-based CCID3 congestion control mechanism provides the WiMAX node a comparatively smooth source throughput. However, it has more fluctuations than it is in the wired network which is caused by the unexpected packet loss in the wireless channel.

From the above results, we can find that the source throughput performance in the Mixed (Wired/WiMAX) network is more or less similar to it is in the WiMAX environment. This is acceptable since the source throughputs are all measured from the same WiMAX source node.

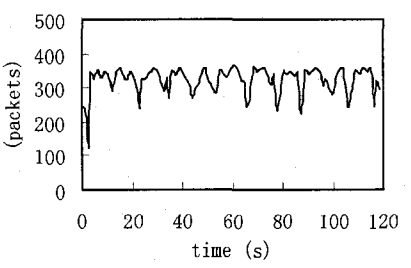
6.3.1.2 Buffer Queue Size

Figure 6-10 shows the buffer queue size of the bottleneck router under different related congestion control strategies.

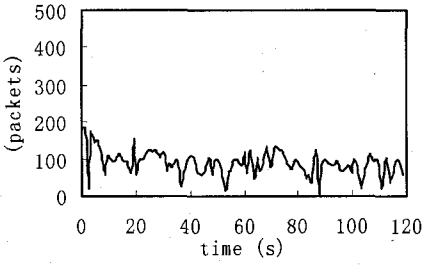
Figure 6-10a shows the buffer queue size under TCP/RED strategy. Like in the wired and WiMAX networks, TCP/RED congestion control strategy maintain a comparative large buffer queue size whose average value is around 350 packets. As can be seen from Figure 6-10b, the average queue size under DCCP/RED congestion control strategy is a little shortened to 330 packets by the cooperation of the DCCP-CCID2 algorithm's *pipe* and ACK Ratio functions. However, RED algorithm still cannot stabilize the bottleneck router's buffer queue size very well.

From Figure 6-10c and Figure 6-10d, we can see that the VCP algorithm performs well and greatly reduces the average buffer queue size to 80 packets for TCP/VCP (Figure 6-10c) and 75 packets for DCCP/VCP (Figure 6-10d) respectively. However, there are more fluctuations in the buffer queue size. This is because the VCP controller tries to maintain a small buffer queue size according to the traffic load factor ρ which is feedback to the source node by ACKs. All these take time. When the MD algorithm kicks in, the buffer queue size is

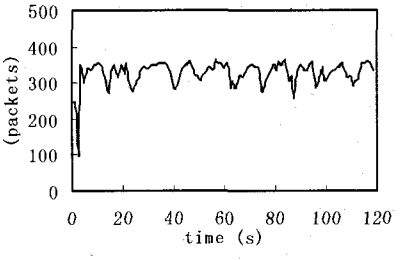
greatly shortened. However, the AI algorithm that follows increases the incoming traffic load significantly. The source nodes have to wait until the next feedback and execute the MD algorithm to reduce the congestion window size. However, fluctuation reflects the repetition of these actions.



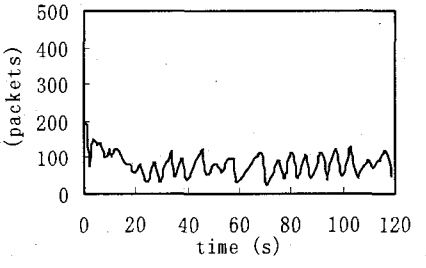
(a) TCP/RED



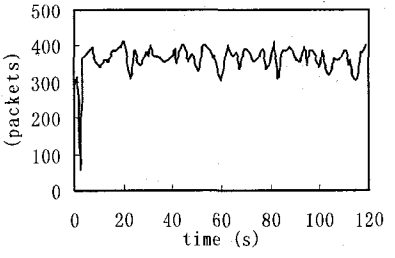
(c) TCP/VCP



(b) DCCP/RED



(d) DCCP/VCP



(e) CCID3/RED

Figure 6-10: Buffer Queue Size under Different Control Strategy (Mixed)

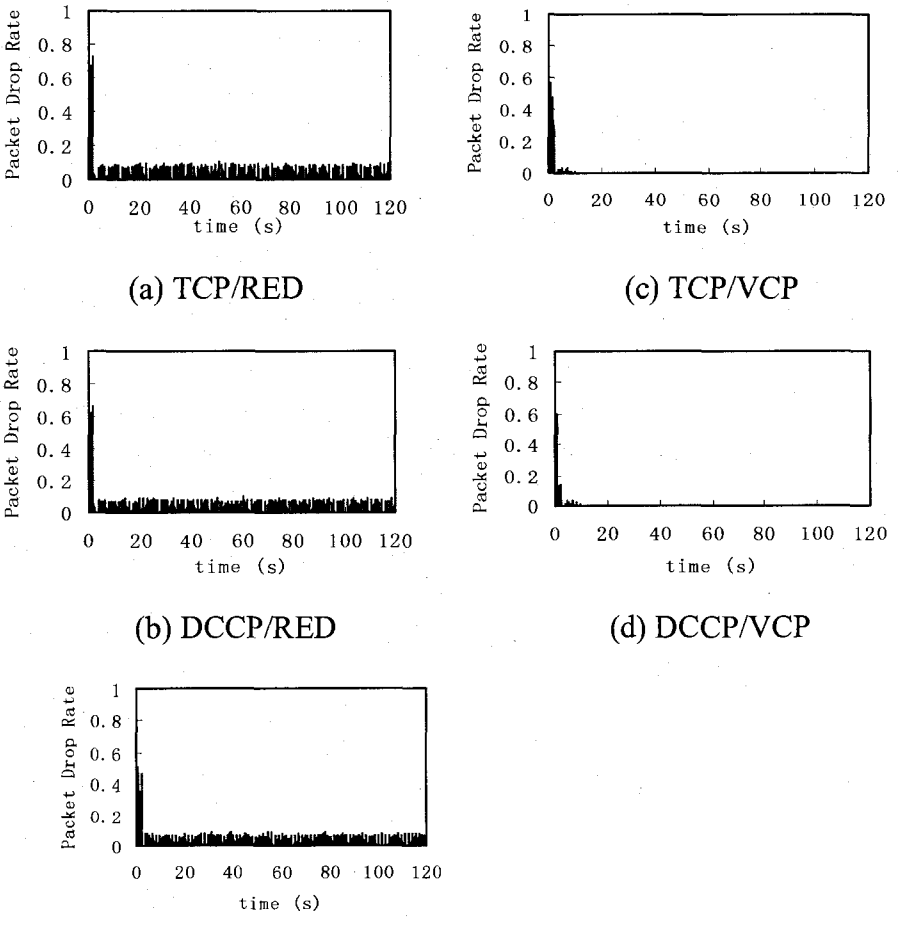
Figure 6-10e shows the buffer queue size under CCID3/RED congestion control strategy. The buffer queue size is the largest around all of the five control strategies, which is about 380 packets. This result also shows that CCID3/RED gives poor performance in stabilizing buffer queue size as it is in the pure WiMAX network.

6.3.1.3 Packet Drop Rate

Figures 6-11 shows the packet drop rate of the bottleneck router in our network model.

Packet drop rate is caused by buffer overflow. It can be seen from Figure 6-11a that the RED router with 200 TCP sources keeps dropping packets to maintain a healthy traffic. The packet drop rate is very high in the beginning of the simulation. This is because e all 200 TCP sources try to send as many packets as it can. However, after the 3rd second, about 8% of the packets are dropped during the simulated time period.

Figure 6-11b shows the packet drop rate of RED router with 200 DCCP sources. As the control strategy, the RED router always discards 65% of the DCCP packets sent into the router in the beginning of simulation. The average packet drop rate is around 6% which is a little bit lower than that of TCP/RED. DCCP has no data retransmission when a drop event happens and CCID2's TCP-like congestion control algorithm cooperate with the ACK ratio mechanism to limit the number of dropped packets.



(a) TCP/RED

(c) TCP/VCP

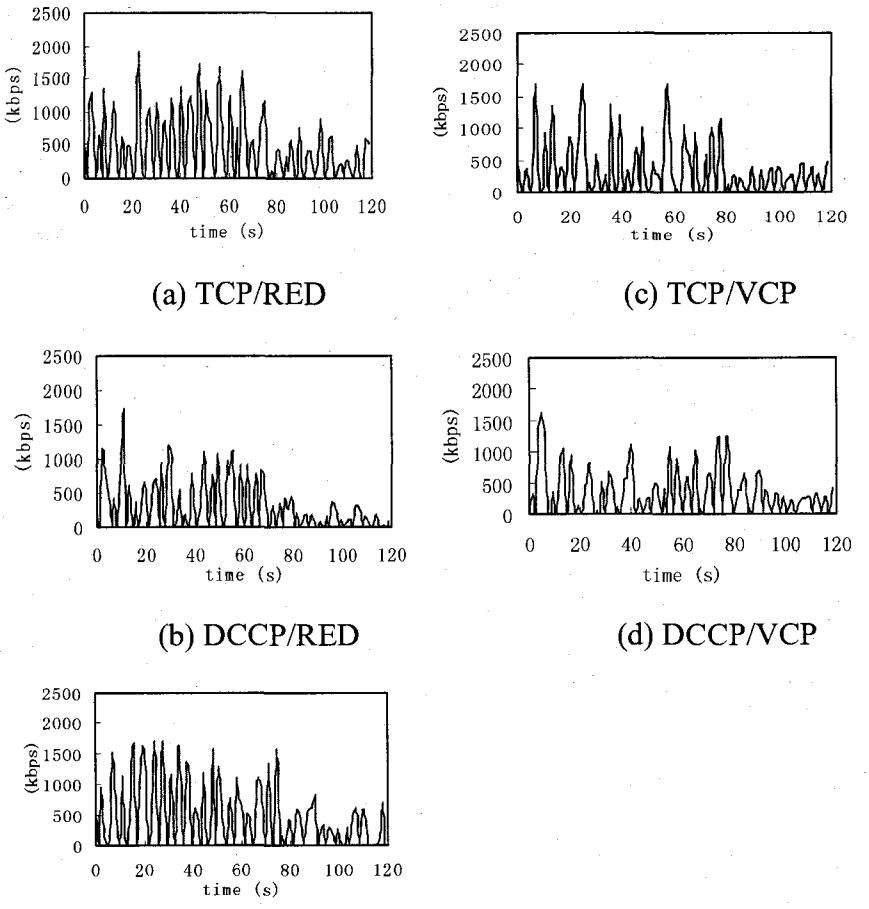
(b) DCCP/RED

(d) DCCP/VCP

(e) CCID3/RED

Figure 6-11: Packet Drop Rate under Different Control Strategy (Mixed)

Figure 6-11e shows the packet drop rate under CCID3/RED strategy with N = 200 DCCP-CCID3 sources. Like the other congestion control strategies, the packet drop rate is very high at the 1st second. About 56% incoming packets are dropped by the RED router. As time goes by, the widely adopted AQM RED algorithm tries to avoid the buffer queue from overflow. Fewer packets are dropped after the 5th second. However, the packet drop rate under CCID3/RED congestion control strategy is still the largest (about 10%). This also verifies the correctness of Figure 6-11e.



(a) TCP/RED (c) TCP/VCP

(b) DCCP/RED (d) DCCP/VCP

(e) CCID3/RED

Figure 6-12: Source Throughput under Different Control Strategy (Mixed-S2)

In this section, we find that VCP algorithm can protect the data packets from being dropped. The performance of packet drop rate in the traffic load based VCP router is much less than that in the RED router. However, like the pure WiMAX network, the WiMAX source throughput in the mixed network is still not as smooth as it is in the wired

environment. The performance of buffer queue size and packet drop rate also verify the correctness of each other.

6.3.2 Scenario 2: With a Step Change

Like the wired and WiMAX environment, in scenario 2, we only enable 100 source nodes in our bottleneck network at the beginning of the simulation. The other 100 source nodes will enter the network from the 80th second of the simulating time. There are 200 active traffic flows from the 80th second to the 120th second.

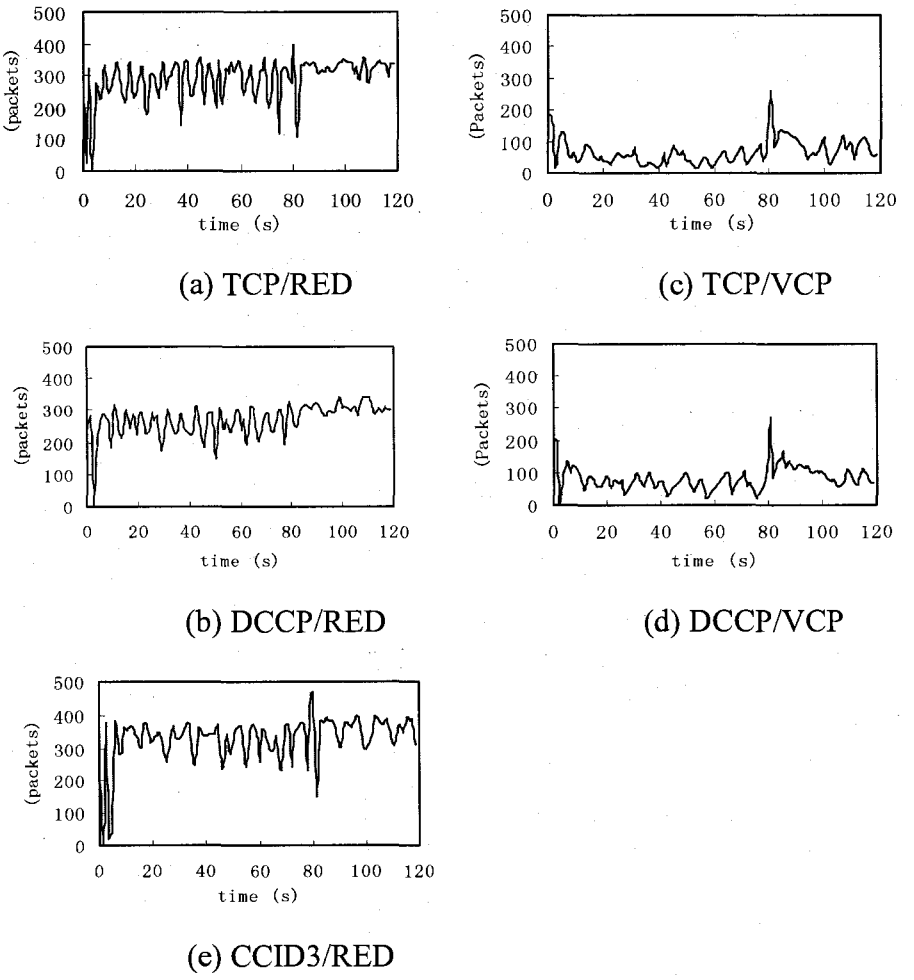


Figure 6-13: Buffer Queue Size under Different Control Strategy (Mixed-S2)

6.3.2.1 Source Throughput

Figure 6-12 shows the source throughputs under the 5 control strategies. As we know from the previous sections, VCP algorithm cannot provide the WiMAX source flow a smooth

source throughput. However, the WiMAX source throughputs in mixed environment in scenario 2 are fluctuating according to the situation of the network reasonably. Before the 100 new source flows enter the bottleneck network, the source throughputs are around 45 kilobits per second, which shows the fairness of the 5 control strategies. After the 100 new flows enter the network at the 80th second, the source throughputs reduce to around 225 kilobits per second, which also shows the fairness of the network.

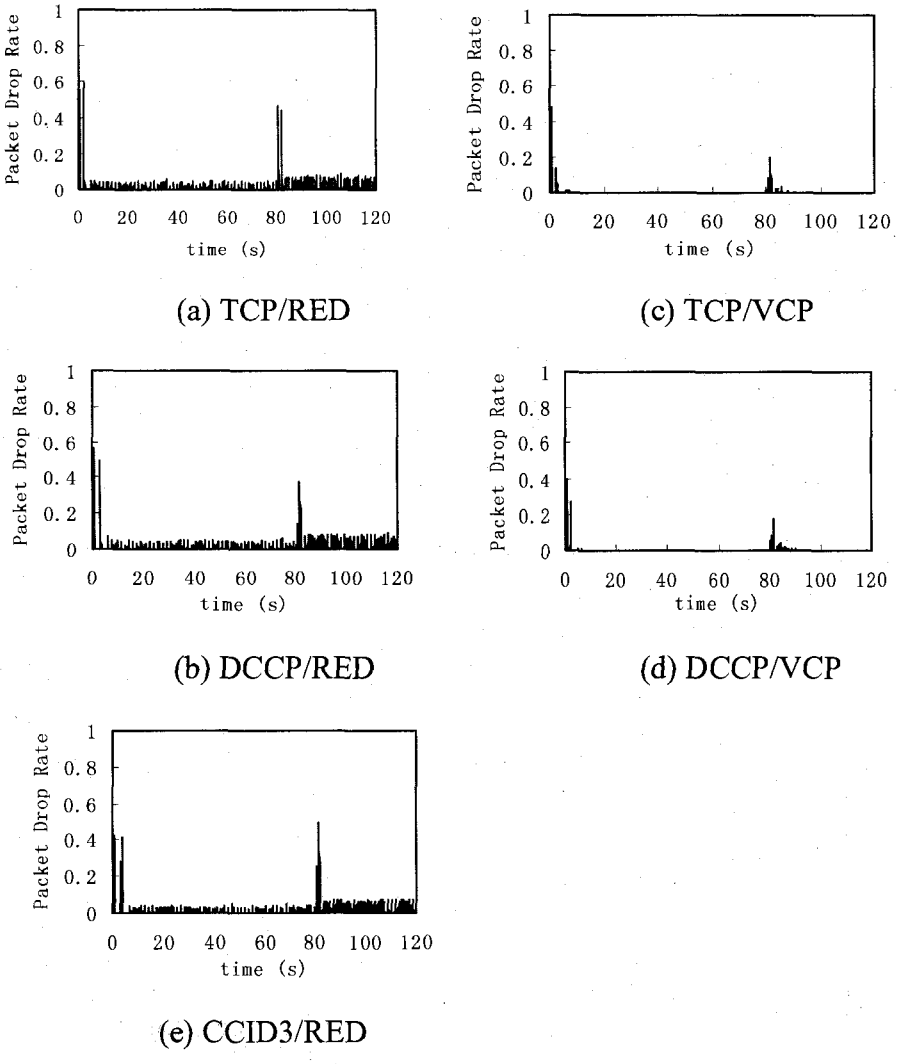


Figure 6-14: Packet Drop Rate under Different Control Strategy (Mixed-S2)

6.3.2.2 Buffer Queue Size

The buffer queue sizes in the mixed network under the 5 control strategies are shown in Figure 6-13. As our experience in the previous sections of the thesis, the buffer queue sizes are more or less similar to the performance in the pure WiMAX network. The shortest buffer

queue sizes are provided by the VCP router. The largest buffer queue size is caused by CCID3/RED control strategy. At the 80th second when the number of traffic flows traversing the bottleneck router changes from 100 to 200, the buffer queue sizes increases with a big overshoot to a higher level.

6.3.2.3 Packet Drop Rate

Figure 6-14 shows the packet drop rate under TCP/RED, DCCP/RED, TCP/VCP, DCCP/VCP, and CCID3/RED in the Wired/WiMAX mixed network in scenario 2. As can be seen from Figure 6-14a, b, and e, there are 5% received data packets dropped by the RED router before the 100 new traffic flows enter the network. When there are 200 active traffic flows in the network, the packet drop rate in RED router increases to about 10%.

From Figure 6-14c and Figure 6-14d, even though the sudden change of the number of flows traversing the bottleneck router causes the packet loss, there is still almost no packet loss in the VCP router.

6.3.3 Effect of Round Trip Time Delay

The effect of varying round trip delay on the performance of the 5 control strategies in mixed network is investigated in this section. In the real world, the WiMAX traffic will always share the network resource together with the wired network. To investigate the effect of *RTT*, we take the mixed network as the model to investigate the effect of *RTT*. The basic setting is a 45Mbps bottleneck link capacity with 100 long-lived WiMAX traffic flows and 100 long-lived wired traffic flows in the simulation. As the wired and WiMAX network, the average round trip time delay of the traffic flows traversing the bottleneck router is varying from 0.1 second to 0.5 seconds. The average buffer queue size and packet drop rate are measured.

Figure 6-15 shows the effect of different round trip time delay on the average buffer queue size under 5 control strategies in a mixed network. As we can see, the average buffer queue size of TCP/VCP is about 140 for *RTT*=0.1s. The average buffer queue size under DCCP/VCP is similar to that under TCP/VCP. However, DCCP/VCP has a relatively smaller average buffer queue size. As the *RTT* increases, the buffer queue size decreases slightly and levels off at about 80 packets. The buffer queue sizes of TCP/RED are at 420 for the *RTT*=0.1s. The average buffer queue size performance under DCCP/RED is similar except

the values are relatively smaller than those under TCP/RED. The CCID3/RED strategy has the largest buffer queue size among the other four. Its buffer queue size is at about 460 packets when *RTT*=0.1s.

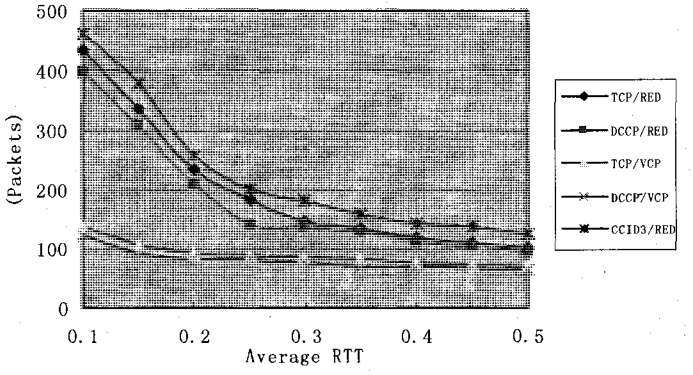


Figure 6-15: Effect of RTT on Average Buffer Queue Size in Mixed Network

We note that when the round trip time delay is large, the difference in the buffer queue sizes among the 5 congestion control strategies is small. In the mixed network, VCP algorithm also provides the TCP and DCCP flows much shorter buffer queue size than RED algorithm. CCID3/RED's buffer queue size is the largest since CCID3's TFRC algorithm is incapable of stabilizing buffer queue size as known in Section 6.3.1.2.

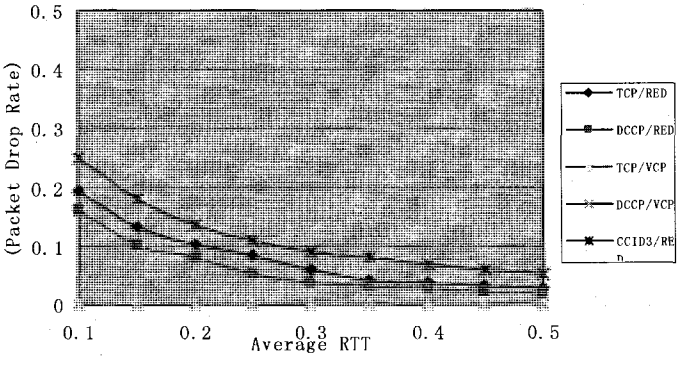


Figure 6-16: Effect of RTT on Packet Drop Rate in Mixed Network

The effect of varying round trip time delay on the packet drop rate under 5 congestion control strategies in a mixed network is shown in Figure 6-16. As we can see, no mater the *RTT* change, the VCP algorithm provides the TCP and DCCP traffic zero packet drop rate in the mixed network. The RED algorithm provides CCID3 the largest packet drop rate among the five strategies. The largest value is about 25% when *RTT*=0.1s. As the *RTT* increases, the

packet drop rate decreases to about 5% when *RTT*= 0.5s. The packet drop rate under TCP/RED is at about 20% when *RTT*=0.1s. It decreases to 4% when *RTT*=0.5s. The packet drop rate under DCCP/RED is similar as that under TCP/RED. However, it is a slightly smaller than that in TCP/RED.

We can see that the varying round trip time delay does not influence the packet drop rate in the VCP router. There is still almost no packet loss in VCP router in the mixed network. However, the RED router always tries to drop data packets. As the RTT is increasing, fewer packets are dropped by the RED router. Since the CCID3/RED strategy gives the largest buffer queue size among all the control strategies, the packet drop rate under CCID3/RED is the largest.

6.4 Comparison

Having evaluated the performance of the wired network, the WiMAX network and the mixed network, we are now in a position to compare these networks in terms of their throughput, buffer queue length, packet drop rates and the effect of RTT on the performances.

First, we note that the source throughputs of WiMAX nodes do not perform as well as it is in the wired environment. The window-based congestion control strategy halves the congestion window *cwnd* per packet loss. As a rate-based source control mechanism, CCID3's TFRC algorithm is supposed to provide a smooth flow throughput over wired networks. However, the packet loss caused by the wireless channel influences the rate-based CCID3 algorithm. The source throughput of WiMAX CCID3 source node fluctuates a lot. The wired network provides the best source throughput which is much smoother than that of the WiMAX and mixed networks. The WiMAX and mixed source throughput is similar and fluctuating all the time. This is caused by the expected packet loss in the wireless channel. Overall, there is almost no difference between the WiMAX source throughput and the Mixed source throughput.

The buffer queue size of the congested router in WiMAX network is larger than that in the wired network. This is caused by the unexpected packet loss in wireless channel. The packet loss in wireless channel may cause the loss of the important congestion information. This can cause buffer over flow. The buffer queue size under the mixed network is similar to that of the WiMAX network which is larger than that in the wired network. Another thing we

note is that the average buffer queue sizes of DCCP/AQM strategies are always smaller than that of TCP/AQM strategies. This shows that the mean queuing delay of DCCP traffic is shorter than that of TCP traffic. The short delay of DCCP can provide much benefit to real-time multimedia transmissions. The queuing delay of the wired network is the shortest over all of the three networks. The queuing delay of the WiMAX and mixed networks is similar. Overall, the WiMAX network provides a slightly shorter buffer queue size and shorter queuing delay than the mixed network.

The packet drop rate of the congested RED router in WiMAX network is greater than that in the wired network. This is reasonable because the larger buffer queue size caused by the wireless link can cause more packet loss. The packet drop rate under the mixed network is similar to that under the WiMAX network. Overall, the packet drop rate under the WiMAX network is slightly less than that under the mixed network. However, the VCP router almost does not drop packet in all the three networks.

When the *RTT* increases in the wired network, the buffer queue size and the packet drop rate decreases. The buffer queue size and the packet drop rate in WiMAX network change almost the same as it is in wired network. However, when the *RTT* is less than 0.2 second, the buffer queue size and the packet drop rate change faster than it is in wired network. This shows that when the *RTT* is small, the bad performance caused by the wireless channel is more obvious.

6.5 Concluding Remarks

This chapter provides three performance measures (WiMAX source throughput, buffer queue size, and packet drop rate) in two different network scenarios (WiMAX and Mixed networks). The results show that although the source throughput under the control strategy of DCCP/VCP is not as smooth as it is in the wired environment. DCCP/VCP still performs well on maintaining a small buffer queue size and avoiding packet loss due to congestion. We also note shorter round trip time delay gives a better queuing performance in terms of the average of buffer queue size, packet drop rate, and their variations.

In summary, we make the observation that the VCP can perform better than all the other algorithms in all types of networks except for the source throughput (with comparable results). By itself, VCP performs the best in a wired network as expected.

Chapter 7
Conclusion

This thesis attempts to solve the problems of multimedia transmission that is under the widely adopted TCP/RED. We have proposed to use DCCP/VCP congestion control strategy to support multimedia transmissions. With the TCP-like DCCP-CCID2 window-based congestion control mechanism which dose not require the data packets arrives at the destination node in order, the random delay problem [KoHa06] arising from waiting for the missing packets in the absolute reliable TCP mechanism has been solved. In addition, VCP algorithm can be used as the router feedback congestion control mechanism. Instead of randomly dropping packets in RED, VCP uses the traffic load factor to divide the incoming data traffic to the gateway router into low-load, high-load, and over-load, three regions. Then it leverages only two ECN bits to feedback the traffic load information to the DCCP source nodes. The DCCP source nodes adjust the congestion window size according to the three traffic load regions. This mechanism insures that there is no congestion in the gateway router.

We have made mathematical analysis and performance evaluation to verify our proposed DCCP/VCP controller. A continuous-time fluid model is used. The effect of ACK Ratio in DCCP-CCID2 is taken account into the VCP operation. According to the fluid model and the MIAIMD model of VCP, we obtain the steady state congestion window size and the source flow rate at equilibrium.

In evaluating the performance of DCCP/VCP control strategy, we have implemented a single bottleneck VCP router in a large scale DCCP-CCID2 flows using OPNET simulator. We have also conducted a simulation in the wired, the WiMAX, and the mixed networks. The performance measures of source throughput, buffer queue size, and packet drop rate are used in our performance evaluation. We run OPNET simulations under the related congestion control strategies, TCP/RED, DCCP/RED, TCP/VCP, DCCP/VCP, and CCID3/RED to conduct the performance comparison.

In the wired environment, the DCCP-CCID2 source throughput under DCCP/VCP strategy is as smooth as CCID3/RED. In addition, the buffer overflow problem of RED algorithm is well solved by VCP algorithm. There is almost no packet dropped by the

bottleneck VCP router. However, in the WiMAX and mixed networks, the WiMAX source throughput is hardly smoothed by VCP algorithm. Unexpected packet loss in the wireless channel degrades the source throughput performance of DCCP/VCP control strategy. On the other hand, the buffer queue size and packet drop rate is still very small under DCCP/VCP strategy.

In summary, VCP can provide a smooth source throughput, a small buffer queue size, and almost zero packet drop rate in wired network. Thus, it can well support a resilient real-time DCCP communication over wired environment. The source throughput in a WiMAX network is not as smooth as that in the wired environment due to the unexpected corruption nature of wireless link. However, DCCP/VCP can provide a relatively smaller buffer queue size and almost no packet loss in the router. Thus DCCP/VCP can be recommended to transmit multimedia applications for WiMAX as well.

7.1 Design Experience

Simplicity and compatibility are two design issues of this thesis. By "simplicity" we mean an AQM-style approach where routers merely provide feedback on the level of network congestion, and end-hosts only need to perform congestion control actions based on this feedback. For compatibility, we need to allow the use of the existing IP header format. To address these challenges, we have built a simple TCP-like window-based control with the use of link load factor as a congestion signal to feedback the source nodes. We also restrict ourselves to using only two bits to encode the congestion information.

Our simulation results in Chapter 5 and Chapter 6 show that our algorithm provides short buffer queue size and zero packet drop rate. The source throughput shows the fairness of our algorithm. To achieve a real-time multimedia communication, DCCP is a good choice to avoid congestion as well as decreasing TCP's arbitrary delay caused by the sequence in order for received data packets. From our simulation results, when there is a sudden change in the network, such as suddenly enter 100 new traffic flows and varying round trip time delay, the traffic flow throughput changes to achieve fair bandwidth allocation. The buffer queue size and packet drop rate also change to avoid congestion and packet loss. Our simulation can also support more source nodes such as 500 or 1000 source nodes sending packets. The maximum number of source nodes can be supported by our simulation is around

2000. When the RTT is smaller than 30ms, DCCP/VCP is very sensitive to the sudden change. Since in the real network environment, the round trip time delay is larger than 0.3 seconds for nearly most of the situations, the RTT issue should not be a problem.

From the research work in this thesis, I learned that congestion control algorithms in transport layer is very important for a network to operates well and avoid packet loss. No matter whether it is window-based source control mechanism or rate-based source control mechanism, the router-based control is indispensable. The correct cooperation between source and router based control mechanisms is very important to achieve the efficiency and optimality. This can be illustrated by the programming mistakes when I implemented the VCP algorithm in the router. For example, when I need the value from a packet header, I just declared a static variable and get it in one process state (e.g. packet arrival state). Unlike the local variable that can be only seen or changed by a state or a function, the static variable can be seen or changed by all the states in a process model. Therefore, when we process this value in the other process state (e.g., service complete state), the value could have already been changed by another arrived packet (e.g., in the packet arrival state). This has caused a lot of network anomaly. The correct approach is to declare a local variable and always get the value from the packet before using it. In the way of resolving the difficulties such as modeling the DCCP source, implementing the VCP algorithm in the router, and building a correct WiMAX node, I really learned a lot from the research work in this thesis.

Notably that, the VCP controller has its limitation. When there is loss event, the source performs the same as TCP-Reno. That is, whenever there is a packet dropped, the congestion window size will be halved. In wired networks, there is almost no packet dropped, this limitation is not obvious. However, the limitation makes the source throughput performance worse in wireless networks. On the other hand, the limitation of our DCCP/VCP lies in that cannot improve the source throughput in the WiMAX environment. The WiMAX node sending rate is fluctuating all the time. This problem is caused by the unexpected packet loss by the wireless link.

7.2 Future Work

There are still some remaining issues to be investigated regarding the implementation of the DCCP/VCP in a multiple-router network. A multiple-router network can provide a multiple

bottleneck in a network which is much more complex than the single bottleneck network. However, it may better reflect the real world network environment than the single bottleneck network.

In addition, addressing the throughput oscillation problem in the WiMAX environment, we would like to develop a congestion control mechanism that can distinguish the congestion packet loss from the wireless channel packet loss. This is a challenge for all the researchers who want to execute congestion control in the wireless environment. Others might involve relaxing some of the assumptions such as the buffer size.

References

[AlPa99] M. Allman, V. Paxson and W. Stevens, "TCP Congestion Control", *RFC 2581*, April 1999. http://www.faqs.org/rfcs/rfc2581.html.

[BaPa97] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performances over Wireless Links", *IEEE/ACM Transactions on Networking,* vol. 5, pp.756-769, Dec. 1997.

[BaSe95] H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz, "Improving TCP/IP Performances over Wireless Networks", *Proceedings of IEEE Mobicom 95.* pp. 2-11, Berkeley, California, November 1995.

[BrOM94] L.S. Brakmo, S.W. O'Malley, and L.L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance", *Proceedings of ACM SIGCOMM 1994*, pages 24--35, London, UK, May 1994.

[CaGe00] C. Casetti, M. Gerla, S. S. Lee, S. Mascolo and M. Sanadidi, "TCP with Faster Recovery", *Proceedings of IEEE Milcom 2000*, pp. 320-324, Los Angeles, CA, October 2000.

[CaGe01] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", *Proceedings of IEEE Mibicom 2001.* pp 287-297, Rome, Italy, Jul. 2001.

[CrBe97] M. Crovella and A. Bestavros, "Self Similarity in World Wide Web Traffic: Evidence and Possible Causes", *IEEE/ACM Transactions on Networking,* vol. 5, pp.835-846, Dec. 1997.

[EwRi55] E. Wright, "A Non-linear Difference-Differential Equation", *J.Reine Angew. Math.*, 494:66-87, 1955.

[FaFl96] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", *Computer Communication Review*, V. 26, N. 3, pp. 5-21, July 1996.

[FeKa99] W. Feng, D. D. Kandlur, D. Saha and K.G. Shin, "A Self-configuring RED Gateway", *Proceedings of IEEE Infocom 1999.* pp. 1320-1328, New York, NY, March 1999.

[FiBo00] V. Firoiu and M. Borden, "A Study of Active Queue Management for Congestion Control", in *Proceedings of IEEE Infocom 2000.* pp. 1435--1444, Tel-Aviv, Israel, March 2000.

[FlHe99] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", *RFC 2582*, April 1999. http://www.faqs.org/rfcs/rfc2582.html.

[FlJa93] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Control", *IEEE ACM Transactions on Networking*, Vol. 1, No. 4, pp.397-413, 1993.

[FlKo06a] S. Floyd and E. Kohler, "TCP Friendly Rate Control (TFRC): the Small-Packet (SP) Variant", *Internet Draft, Mar.* 2006.

[FlKo06b] S. Floyd, E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control," *RFC4341, Mar.* 2006.

[FlKo06c] S. Floyd, E. Kohler, J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)," *RFC4342. Mar.* 2006.

[GeSa01] M. Gerla, M. Sanadidi, R. Wang and A. Zanella, "TCP Westwood: Congestion Window Control Using Bandwidth Estimation", *Proceedings of IEEE Globecom 2001.* San Antonio, Texas, USA, *Nov.* 2001.

[HaFl03] M. Handley, S. Floyd, J. Padhye, J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," *RFC3448, Jan 2003.*

[Hoe96] J. C. Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP", *Proceedings of ACM Sigcomm 1996.* pp. 270-280, Stanford, CA, August 1996.

[HoMi01a] C.V. Hollot, V. Misra, D. Towsley and W. Gong, "A Control Theoretic Analysis of RED", *Proceedings of IEEE Infocom 2001.* pp. 1510-1519, Anchorage, AK, Apr. 2001.

[HoMi01b] C.V. Hollot, V. Misra, D. Towsley and W. Gong, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows", *Proceedings of IEEE Infocom 2001.* pp. 1726-1734, Anchorage, Alaska, USA, April 22--26, 2001.

[HoYa06] Hong Yang, O. Yang, "Self-Tuning Utility-Based Controller for End-to-End Congestion in the Internet" *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference.* pp. 1-10.

[IeDc02] IETF DCCP working group. http://www.ietf.org/html.charters/dccp-charter.html. 2002.

[IeWi05] IEEE Standard for Local and metropolitan area networks 802.16e, 2005

[Jaco88] V. Jacobson, "Congestion Avoidance and Control", *Proceedings of ACM Sigcomm 1988.* pp. 314-329, Stanford, CA., August 1988.

[Jaco90] V. Jacobson, "Modified TCP Congestion Avoidance Algorithm", message to end2end-interest mailing list, Apr. 1990. ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt

[KaHa02] D. Katabi, M. Handley, C. Rohrs, "Congestion Control for High Bandwidth Delay Product Networks," ACM/*SIGCOMM Conference 2002, Proceedings, pp.89-102. August 19-23, 2002, Pittsburgh, USA.*

[KoHa01] E. Kohler, M. Handley, S. Floyd, and J. Padhye, "Datagram control protocol (DCP)". *IETF Internet-Draft, draft-kohlerdcp-00.txt,* July 2001.

[KoHa06] E. Kohler, M. Handley, S. Floyd, "Designing DCCP: Congestion Control without Reliability," ACM/*SIGCOMM Conference 2006, Proceedings, pp.27-38, Sep. 11-15, 2006, Pisa, Italy.*

[KuLa01] P. Kuusela, P. Lassila and J. Virtamo, "Stability of TCP-RED Congestion Control", *Proceedings of ITC-17,* pp. 655-666, Salvador da Bahia, Brazil, 2001.

[LeTa94] W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)", *IEEE ACM Transactions on Networking*, Vol. 2, No. 1, pp.1-15, 1994.

[LiMo97] D. Lin and R. Morris. "Dynamics of Random Early Detection". *Proceedings of Sigcomm 1997*. pp. 127-137, Cannes, France, *Sep.*1997.

[LoPa02] S. H. Low, F. Paganini, J. Wang, S. Adlakha and J. C. Doyle, "Dynamics of TCP/RED and a Scalable Control", *Proceedings of IEEE Infocom 2002*. New York, Jun. 2002.

[MaMa96a] M. Mathis and J. Mahdavi," Forward Acknowledgment: Refining TCP Congestion Control ", *Computer Communication Review*, Vol. 26, No. 4, pp. 281-291, Oct.1996.

[MaMa96b] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options", October 1996, *IETF RFC2018.* http://www.faqs.org/rfcs/rfc2018.html.

[Masc99] S. Mascolo, "Congestion Control in High-Speed Communication Networks Using the Smith Principle", *Automatica Journal, Special Issue on "Controls Methods for Communications Networks"*, vol.35, no.12, pp.1921-1935, Dec. 1999.

[MaSe97] M. Mathis, J. Semke, J. Mahdavi and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoid Algorithm", *ACM Computer Communication Review,* V. 27, N. 3, pp. 67-82, Jul. 1997.

[Morr97] R. Morris, "TCP Behavior with Many Flows", *IEEE International Conference on Network Protocol*, Oct. 1997.

[Morr00] R. Morris, "Scalable TCP Congestion Control", in *Proceedings of IEEE Infocom 2000*. pp. 205--211, October 1997, Atlanta, GE, USA, Oct. 1997.

[Opne05] Opnet Modeler, OPNET Version 10.5, Opnet Technologies, Inc, 2005

[OtLa99] F.J. Ott, T.V. Lakshman and L. Wong, "SRED: Stabilized RED", *Proceedings of IEEE Infocom 1999*. pp 1346-1355, Piscataway, N.J., 1999.

[PaFl95] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling", *IEEE/ACM Trans. on Networking*, vol. 3, No. 3, pp. 226--244, 1995.

[SaFe99] D. Saha, W. Feng, D. Kandlur and K. Shin, "BLUE: A New Class of Active Queue Management Algorithms", *Technical report CSE-TR-387-99, University of Michigan*, Apr. 1999.

[Sast75] A.R.K. Sastry, "Improving repeat-request (ARQ) performance on satellite channels under high error rate conditions", *IEEE Transactions on communications*, Vol. Com-23, No. 4, pp.436-439, Apr 1975.

[Stev94] W. Stevens, *TCP/IP* Illustrated, Volume 1, Addison-Wesley, Reading, MA, 1994.

[Stev97] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", *RFC 2001*, Jan. 1997. http://www.faqs.org/rfcs/rfc2001.html.

[TiMa01] P. Tinnakornsrisuphap and A. M. Makowski, "Queue Dynamics of RED Gateways under Large Number of TCP Flows", *Proceedings of IEEE Globecom 2001*. San Antonio, Texas, USA, November 2001.

[WiDe01] J. Widmer, R. Denda and M. Mauve, "A Survey on TCP-Friendly Congestion Control", *Special Issue of the IEEE Network Magazine "Control of Best Effort Traffic"*, vol. 15, no. 3, pp. 28--37, May/June 2001.

[XiSu05] Y. Xia, L. Subramanian, L. Stoica, S. Kalyanaraman, "One More Bit is Enough," ACM/*SIGCOMM Conference 2005, Proceedings. August 22-26, 2005.*

[YkUa93] Y. Kuang, "*Delay Di®erential Equations with Applications in Population Dynamics*".

[ZhYa03] Hongyi Zhang, Jun Cong and Oliver Yang, "Rate Control over RED with Data Loss and Varying Delays", *Proceedings of Globecom 2003*, San Francisco, Dec 2003, Session NG11-4.

Appendix A: RED

RED (Random Early Detection) was originally proposed [FlJa93] to overcome the severe performance problems with the earlier gateway queue management algorithm, e.g. the bias against bursty traffic and global synchronization [FlJa93]. The RED gateway uses a low-pass filter with an exponential weighted moving average to calculate the average queue size avg, which is then compared to two preset parameters, the minimum threshold $th_{\min}$ and the maximum threshold $th_{\max}$. If avg is less than $th_{\min}$, no packets will be dropped. If it is greater than $th_{\max}$, then every incoming packet will be dropped. If avg is between $th_{\min}$ and $th_{\max}$, the incoming packet will be dropped with a probability p_a, which is a function of the average queue size.

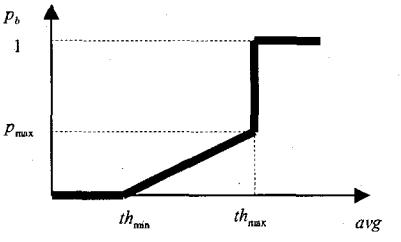


Figure A-1: Temporary Dropping Probability of RED

Eqn (A-1) shows how the temporary packet-dropping probability p_b is calculated, which is also indicated in Figure A-1. The final probability p_a with which an arriving packet is dropped is calculated in eqn (A-2), from which we can see that RED ensures that the gateway does not wait too long before dropping a packet.

$$p_b = p_{\max}(avg - th_{\min})/(th_{\max} - th_{\min}) \quad , \tag{A-1}$$

where $p_{\max}$ is the maximum allowed value for p_b.

$$p_a = p_b/(1 - count \cdot p_b) \quad , \tag{A-2}$$

where $count$ is the number of packets since last dropped packet.

Initialization:
    $avg := 0$
    $count := -1$
for each packet arrival
    calculate the new average queue size avg:
        if the queue is nonempty
            $avg := (1 - w_q)avg + w_q q$
        else
            $m := f(time - q\_time)$
            $avg := (1 - w_q)^m avg$
    if $th_{\min} \le avg < th_{\max}$
        $count++$
        calculate p_a
            $p_b := p_{\max}(avg - th_{\min})/(th_{\max} - th_{\min})$
            $p_a := p_b/(1 - count \cdot p_b)$
        drop the incoming packet with p_a
        $count := 0$
    else if $th_{\max} \le avg$
        drop the incoming packet
        $count := 0$
    else
        $count := -1$
when queue becomes empty
    $q\_time := time$

Saved Variables:

avg : average queue size
q_time : queue idle start time
$count$: number of packets since last \dropped packet

Parameters:

w_q : queue weight
$th_{\min}$: minimum threshold for queue
$th_{\max}$: maximum threshold for queue
$p_{\max}$: maximum allowed value for p_b

Other:

p_a : current packet-dropping probability
q : current queue size
$time$: current time
$f(t)$: ali near function of the time t

Figure A-2: RED Gateway Algorithms

Figure A-2 shows the RED algorithm, which consists of two parts. The first part is to determine the degree of burstiness allowed in the gateway queue through the computation of

the average queue size. The second determines how frequently the incoming packets should be dropped according the current congestion level.

Appendix B: Proof of Equilibrium for Equation (4-12)

To analyze the stability of Equation (4-12), let

$$y(t) = \frac{\gamma \cdot C \cdot T_{avg}}{w(t)} \quad . \tag{B-1}$$

After some manipulation we can obtain

$$\dot{y}(t) = h_1 \cdot y(t) \cdot [1 - y(t - T_{avg}) - h_2 y(t)] \quad . \tag{B-2}$$

where $h_1 = h/T_{avg} > 0$, and $h_2 = Na/(hCT\gamma) > 0$. For all $t > 0$, we also have $y(t) \geq 0$, since $w(t) > 0$, and the initial condition $y(t) > 0$ (since we assume $w(t) = N < 8$) for $t \in [-T; 0]$.

Since the trivial solution $y(t) = 0$ is not a stable equilibrium of this equation, any small perturbation will move the system further off the origin. We therefore assume $y(t) > 0$ and $t > 0$. This assumption implies that the congestion window $w(t)$ is not infinite, at any time in general. Applying the following two lemmas we can show that Equation (4-12) is stable:

***Lemma* 1** [XiSu05]. If $h \leq 1 - \frac{N\alpha}{CT\gamma}$, then Equation (B-2) has a unique equilibrium $y^* = \frac{1}{1+h_2}$, that is globally asymptotically stable.

***Lemma* 2** [XiSu05]. If $h < \frac{N\alpha}{CT\gamma}$, then Equation (B-2) has a unique equilibrium $y^* = \frac{1}{1+h_2}$, that is globally asymptotically stable.

To prove Lemma 1, we first substitute $x(t) = y(t) - y^*$ in Equation (B-2)

$$\dot{x}(t) = -h_1 \cdot [y^* + x(t)] \cdot [x(t - T) + h_2 \cdot x(t)], \tag{B-3}$$

with a solution

$$y^* + x(t) = [y^* + x(t_0)] \cdot e^{-k_1 \cdot \int_{t_0}^{t-T} [x(\tau) + h_2 \cdot x \cdot (\tau + T)] d\tau}, \tag{B-4}$$

where t_0 is a constant. To show $\lim_{t \to +\infty} x(t) = 0$, we treat the following two cases separately.

Case 1.

If $|x(t)| > 0$ when $t > t_1$, for some $t_1 > 0$. For $x(t) > 0$, when $t > t_1 + T$, we have $x(t) < 0$ due to Equation (B-2), which means $x(t)$ is strictly decreasing for all $t > t_1 + T$. Because $x(t) > 0$, there exists a constant c such that $\lim_{t\to+\infty} x(t) = c$, and thus $\lim_{t\to+\infty} \dot{x}(t) = c$. We must have $c = 0$. Otherwise, $c > 0$, then $\lim_{t\to+\infty} x(t) = -c \cdot h_1 \cdot (y^* + c) \cdot (1 + h_2) < 0$, according to Equation (B-2), which results in a contradiction. The same analysis applies to when $x(t) < 0$.

Case 2.

If $x(t)$ is oscillatory, *i.e.*, there is a sequence $t_l' > t_2$ for some $t_2 > 0$, $t_l' \to +\infty$ as $l \to +\infty$ such that $x(t_l') = 0$ for all $x(t_l')$. We first prove that $x(t)$ is bounded when $t > t_2$. Obviously $x(t)$ is lower bounded, since $x(t) = y(t) - y^* > -y^*$ as $y(t) > 0$. Now we show that $x(t)$ is also upper bounded. Let $t_3, t_4 \in (t_2, 8)$, $t_3 < t_4$ be any two consecutive zeros of $x(t)$ such that $x(t) > 0$ for $t_3 < t < t_4$. (If $x(t) < 0$ we get an upper bound 0.) Because $w(t)$ is continuous and differentiable, so does $x(t) = y(t) - y^* = \frac{\gamma CT}{w(t)} - y^*$. Since $x(t_3) = x(t_4) = 0$, $x(t)$ has a local maximum in the interval (t_3, t_4). Denote this interval as $x(t_m)$ where $t_3 < t_m < t_4$, we have $x(t) < x(t_m)$ for all $t \in (t_3, t_4)$, we have $x(t) \leqslant x(t_m)$ for all $t \in (t_3, t_4)$. We also have $x(t_m) = 0$, which is deduced to $x(t_m - T) + k_2 \cdot x(t_m) = 0$ from Equation (B-3). Substituting this result into Equation (B-4), we obtain $x(t) > -y^*$, and $y^* = \frac{1}{1 + h_2}$. Setting $t_0 = t_m$-T, $t = t_m$, we get

$$-h_1 \cdot \int_{t_0 - T}^{t-T} [x(\tau) + h_2 \cdot x \cdot (\tau + T)] d\tau < h_1 \cdot T, \tag{B-5}$$

and thus

$$x(t) \leq x(t_m) < \frac{y^* \cdot (e^{h_1 T} - 1)}{1 + h_2 \cdot e^{h_1 T}}, \tag{B-6}$$

for all $t \in [t_3, t_4]$. Repeating this process for all the consecutive pairs of zeros of $x(t)$ in $(t_2, 8)$, we can see that $x(t)$ is bounded for $t > t_2$. Since $x(t)$ is continuous and bounded, denote

$$u = \limsup_{t\to+\infty} x(t), \qquad v = -\liminf_{t\to+\infty} x(t). \tag{B-7}$$

Obviously we have

$$u \geq -v. \tag{B-8}$$

We now prove that $u = v = 0$ (therefore $\lim_{t\to+\infty} x(t) = 0$). Let $\varepsilon > 0$ be an arbitrarily small constant such that, let $t_5 = t_5(\varepsilon)$, for all $t > t_5 > 0$,

$$-v-\varepsilon < x(t) < u+\varepsilon . \tag{B-9}$$

According to the definition of u, we can always find a local maximum $x(t_n) > u - \varepsilon$ for $t_n > t_5 + T$. Applying the same technique used to derive (B-4) on t_n with (B-7), we get

$$v-\varepsilon < x(t_n) < \frac{y^* \cdot [e^{\phi\cdot(v+\varepsilon)} - 1]}{1+k_2 \cdot e^{\phi\cdot(v+\varepsilon)}}, \tag{B-10}$$

where $\varphi = h_1 \cdot T \cdot (1+h_2) = h + \frac{N\alpha}{\gamma CT} > 0$. Since Equation (B-8) holds for $\varepsilon > 0$, we conclude

$$u < \frac{y^* \cdot [e^{\varphi\cdot v} - 1]}{1+h_2 \cdot e^{\varphi\cdot v}}, \tag{B-11}$$

where $0 < y^* < \frac{1}{1+h_2} < 1$ and $h_2 > 0$. Following the similar steps in deriving (B-9) on a local minimum generates

$$v < \frac{y^* \cdot [1-e^{-\varphi\cdot u}]}{1+h_2 \cdot e^{-\varphi\cdot u}}. \tag{B-12}$$

From the three proved cases (i.e. eqn(B-10), eqn(B-6), and (B-9)) we have now, we can summarize our argument in the following:

i) According to (B-12), if $u < 0$, then $v < 0$, so $-v > 0 > u$, violating (B-6);

ii) According to (B-8), if $u = 0$, then $v \geqslant 0$. We have $v = 0$ as well due to (B-10). Therefore $v = 0$;

iii) According to (B-11), if $u > 0$, then $v > 0$. From (B-10) we have $v < y^* < 1$. If $f = 1$, we get

$$1+u < e^{\phi\cdot v} \leq e^{v} < e^{1-e^{-\phi u}} \leq e^{1-e^{-u}} . \tag{B-13}$$

However, for $u > 0$, we have

$$1+u-e^{1-e^{-u}} = \int_0^u \int_0^{\xi} (1-e^{-\eta}) \cdot e^{(1-e^{-\eta}-\eta)} d\eta d\xi > 0 , \tag{B-14}$$

which contradicts Equation (B-13). To sum up, we must have $u = v = 0$ if $f = 1$, i.e., $h \leq 1 - \frac{N\alpha}{\gamma CT}$, which deduces $\lim_{t\to+\infty} x(t) = 0$.

To prove Lemma 2, we consider the following function

$$V(t) = y(t) - y^* \cdot \log y(t) + \frac{h_1}{2} \cdot \int_{t-T}^{t} [y(\tau) - y^*]^2 d\tau \,. \tag{B-15}$$

Again let $x(t) = y(t) - y^*$. Due to Equation (4-14) and $k_1 > 0$ we have

$$\begin{aligned}
\frac{dV}{dt} &= \dot{y}(t) \cdot [1 - \frac{y^*}{y(t)}] + \frac{h_1}{2} \cdot [x^2(t) - x^2(t-T)] \\
&= \frac{\dot{y}(t) \cdot x(t)}{y(t)} + \frac{h_1}{2} \cdot [x^2(t) - x^2 \cdot (t-T)] \\
&= -h_1 \cdot x(t) \cdot [h_2 \cdot x(t) + x \cdot (t-T)] + \frac{h_1}{x} \cdot [x^2(t) - x^2(t-T)] \\
&= -\frac{h_1}{2} \cdot [(2h_2 - 1) \cdot x^2(t) + x^2(t-T) + 2x(t) \cdot x(t-T)] \\
&= -\frac{h_1}{2} \cdot \{2 \cdot (h_2 - 1) \cdot x^2(t) + [x(t) + x(t-T)]^2\} \\
&\le -h_1 \cdot (h_2 - 1) \cdot x^2(t).
\end{aligned} \tag{B-16}$$

Integrating this inequality over [0, t], we have

$$V(t) + h_1 \cdot (h_2 - 1) \cdot \int_0^t x^2(\tau) d\tau \le V(0)\,, \tag{B-17}$$

where $V(0) = y(0) - y^* \cdot \log y(0) + \frac{h_1}{2} \cdot \int_{-T}^{0} [y(\tau) - y^*]^2 d\tau$ is bounded since $y(t)$ is finite when $t \in$ [-T, 0].

Note we have proved in Lemma 1 that $x(t)$ is bounded for all $t > t_2$ for some $t_2 > 0$. So does $y(t)$ because $y(t) = x(t) + y^*$. Thus $V(t)$ is also bounded. If $k_2 > 1$ (i.e., $k < \frac{N\alpha}{\gamma CT}$), we must have $\lim_{t \to +\infty} x(t) = 0$ since otherwise we obtain $h_1 \cdot (h_2 - 1) \cdot \int_0^t x^2(\tau) d\tau \to +\infty$ as $t \to +\infty$ resulting in a conflict with Equation (B-17).

Appendix C: DCCP-CCID3

This appendix provides more information for use in Section 2.2.2. In order to compete fairly with TCP, CCID3 uses the throughput equation, which roughly describes source node's sending rate as a function of the loss event rate, round trip time delay, and packet size.

The source sending rate r of DCCP-CCID3 follows Equation (C-1):

$$r = \frac{s}{R \cdot \sqrt{\frac{2 \cdot b \cdot p}{3}} + 3 \cdot T_{RTO} \cdot \sqrt{\frac{3 \cdot b \cdot p}{8}} \cdot p \cdot (1 + 32p^2)}. \qquad \text{(C-1)}$$

where s is the packet size. The parameter R is the round trip time delay. The number of packets acknowledged by a single ACK is specified by b. The parameter p is the loss event rate, between 0 and 1, of the number of loss events as a fraction of the number of packets transmitted. The source retransmission timeout is T_{RTO}. In this thesis, we set T_{RTO} to be $4 \cdot R$ for simplify. Notably that, base on TFRC algorithm, the DCCP-CCID3 source sending rate is limited to be always larger than $2 \cdot receive\_rate$ (i.e. two times of the ACK rate).

Appendix D: Process Model

Below are some examples of the process model we used in our simulation:

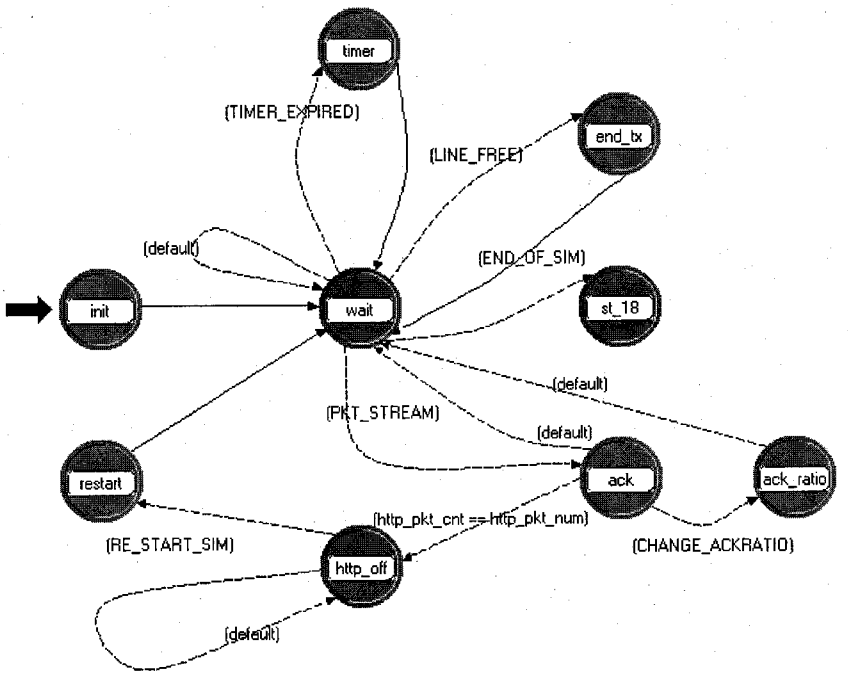


Figure D-11: OPNET Process Model – DCCP Source

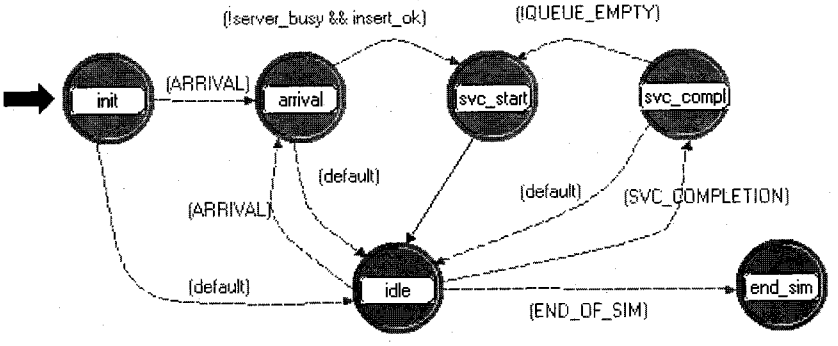


Figure D-12: OPNET Process Model – Router

Figure D-1 shows the OPNET process model of the DCCP source. The DCCP process model includes 3 unforced states (i.e. in red), six forced states (i.e. in green), at which the endpoint will unconditionally return to the unforced state immediately after having

performed the functions in the forced state, and a number of state transition conditions (as arrows). The process model includes all the possible states in which the DCCP source can be. For example, when the simulation begins, the system is in the init state first, and the simulation is initialized. Then the system goes into the wait state. If the DCCP source node receives an ACK, then it enters the ACK state to respond accordingly.

Figure D-2 shows the OPNET process model of the routers. Intuitively, we can see how the router works for incoming packets from its process model. After initializing the initial simulation attributes and receiving a packet, the router enters the arrival state, and then goes into the service start (i.e. svc_start) state if the server is not busy and the packet is successfully inserted into the router buffer queue. After the server finishes processing the packet and no more packets are waiting for service, it enters the idle state from the service complete (i.e. svc_compl) state.

Appendix E: Confidence Interval

This appendix provides more information on the 95% confidence intervals of our measurement. Confidence interval measures the likely difference observed between sample mean μ and the true population mean. One formula is given by:

$$\mu \pm Z_{\alpha/2} \cdot \sigma / \sqrt{n} \qquad \text{(E-1)}$$

The parameter s is the standard deviation. The parameter $Z_{a/2}$ is the distribution value for a given confidence level 1-a which can be obtained from any statistic handbook. Our simulations have provided n=4 samples for us to compute the mean and standard deviation. We shall use the normal distribution for $Z_{a/2}$ in Equation E-1 as our approximation from which we compute the 95% confidence intervals.

Referencing the Z table, we can find the value for $a/2$ is 0.05/2 = 0.025. The Z value is then -1.96. We compute the 95% confidence interval of every second point of the DCCP/VCP curve to cover the whole range.

The average RTT, mean value, standard deviation, and the 95% confidence intervals for the DCCP/VCP curve in Figure 5-7, Figure 6-7, and Figure 6-15, are listed in Figure E-1, Figure E-2, and Figure E-3, respectively.

Avg. RTT	Mean Value	Standard Deviation s	Confidence Interval	
0.1	118.3	0.8	117.5	119.1
0.2	89.3	1.0	88.3	90.3
0.3	85.8	1.0	84.8	86.8
0.4	81.5	0.8	80.7	82.3
0.5	63.8	0.9	62.9	64.7

Figure E-1: Confidence Interval in Figure 5-7

Avg. RTT	Mean Value	Standard Deviation s	Confidence Interval	
0.1	128.1	0.8	127.3	128.9
0.2	69.3	1.0	68.5	70.1
0.3	58.1	0.8	57.3	58.1
0.4	43.6	0.9	42.7	44.5
0.5	30.8	1.0	29.8	31.8

Figure E-2: Confidence Interval in Figure 6-7

Avg. RTT	Mean Value	Standard Deviation s	Confidence Interval	
0.1	121.3	0.8	120.5	122. 1
0.2	82.3	1.0	81. 3	83.3
0.3	76.4	0.8	75. 6	77.2
0.4	69.1	1.0	68. 1	70.1
0.5	65.3	0.9	64. 4	66.2

Figure E-3: Confidence Interval in Figure 6-15