

Using Bandwidth Estimation to Optimize Buffer and Rate  
Selection for Streaming Multimedia over  
IEEE 802.11 Wireless Networks

by  
Mingzhe Li

A Dissertation  
Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment  
of the Requirements for the  
Degree of Doctor of Philosophy

in  
Computer Science

by

---

December 2006

APPROVED:

---

Professor Mark Claypool  
Advisor

---

Professor Robert Kinicki  
Co-advisor

---

Professor Emmanuel Agu  
Committee Member

---

Professor Constantinos Dovrolis  
External Committee Member  
College of Computing  
Georgia Institute of Technology

---

Professor Michael Gennert  
Head of Department



## Abstract

As streaming techniques and wireless access networks become more widely deployed, a streaming multimedia connection with the “last mile” being a wireless network is becoming increasingly common. However, since current streaming techniques are primarily designed for wired networks, streaming multimedia applications can perform poorly in wireless networks. Recent research has shown that the wireless network conditions, such as the wireless link layer rate adaptation, contending traffic, and interference can significantly degrade the performance of streaming media applications. This performance degradation includes increased multimedia frame losses and lower image quality caused by packet loss, and multiple rebuffering events that stop the media playout. This dissertation presents the model, design, implementation and evaluation of an application layer solution for improving streaming multimedia application performance in IEEE 802.11 wireless networks by using enhanced bandwidth estimation techniques. The solution includes two parts: 1) a new *Wireless Bandwidth estimation tool (WBest)* designed for fast, non-intrusive, accurate estimation of available bandwidth in IEEE 802.11 networks, which can be used by streaming multimedia applications to improve the performance in wireless networks; 2) a *Buffer and Rate Optimization for Streaming (BROS)* algorithm using WBest to guide the streaming rate selection and initial buffer optimization. WBest and BROS are implemented and incorporated into an emulated streaming client-server system, *Emulated Streaming (EmuS)*, in Linux and evaluated under a variety of wireless conditions. The evaluations show that with WBest and BROS, the performance of streaming multimedia applications in wireless networks can be significantly improved in terms of multimedia frame loss, rebuffer events and buffer delay.



## Acknowledgments

First of all, I would like to express my deep gratitude to my advisors, Prof. Mark Claypool and Prof. Robert Kinicki, for their advise and encouragement throughout my graduate studies. They inspired my interests in multimedia and network research and guided me to the right direction. Their supports are essential to the completion of this dissertation.

My deep thanks also go to the members of my Ph.D. committee, Prof. Emmanuel Agu and Prof. Constantinos Dovrolis from Georgia Institute of Technology, who provided valuable feedback and suggestions to my comprehensive exam, my dissertation proposal and dissertation drafts.

I would also like to thank all my friends from CC and PEDS research group, Jae Chung, Choong-Soo Lee, Feng Li, Rui Lu, Chunling Ma, Huahui Wu and Hao Shang. They provide many supports and inspiring discussions related to this research.

Thanks to my parents and sisters, for their love and encouragement during my graduate studies in WPI.

Finally, I would like to thank my wife, Dr. Wei Han, for her love, encouragement, patience and understanding throughout the dissertation research. Without her supports, the dissertation could never be completed.



# Contents

<b>Contents</b>	<b>i</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Algorithms</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 The Dissertation . . . . .	4
1.3 Contributions . . . . .	7
1.4 Roadmap . . . . .	10
<b>2 Background</b>	<b>11</b>
2.1 Streaming Multimedia . . . . .	11
2.1.1 Streaming Multimedia Applications . . . . .	12
2.1.2 Media Scaling . . . . .	16
2.1.3 Performance Metrics of Streaming Multimedia . . . . .	20
2.2 Wireless Network . . . . .	22
2.2.1 Overview of Wireless Networks . . . . .	22
2.2.2 IEEE 802.11 Wireless Local Area Networks (WLANs) . . . . .	25

<b>3</b>	<b>Related Work</b>	<b>31</b>
3.1	Streaming Multimedia . . . . .	31
3.1.1	Streaming Buffer . . . . .	31
3.1.2	Characterization of Streaming over Wireless Networks . . . . .	35
3.1.3	Streaming Performance Improvement over Wireless Networks . . . . .	37
3.2	Bandwidth Estimation . . . . .	40
3.2.1	Bandwidth Related Metrics . . . . .	41
3.2.2	Bandwidth Estimation Techniques . . . . .	45
3.2.3	Evaluation of Bandwidth Estimation Techniques . . . . .	52
3.3	Wireless Network Performance Study . . . . .	56
3.3.1	Analytical Modeling . . . . .	56
3.3.2	Network Simulations . . . . .	60
3.3.3	Performance Measurements . . . . .	61
<b>4</b>	<b>Packet Dispersion in IEEE 802.11 Wireless Networks</b>	<b>65</b>
4.1	Overview . . . . .	66
4.2	Packet Dispersion Issues in Wireless Networks . . . . .	67
4.2.1	Rate Adaptation Simulation . . . . .	67
4.2.2	The Impact of Wireless Networks . . . . .	69
4.3	Wireless Network Packet Dispersion Model . . . . .	72
4.3.1	Packet Dispersion Model . . . . .	72
4.3.2	Model Validation . . . . .	77
4.4	Analysis . . . . .	81
4.4.1	Packet Dispersion in 802.11 . . . . .	81
4.4.2	Analysis of the Estimation Results . . . . .	84
4.4.3	Analysis on the Variance of the Bandwidth Estimation . . . . .	87
4.5	Summary . . . . .	91

---

<b>5</b>	<b>WBest: Wireless Bandwidth Estimation Tool</b>	<b>93</b>
5.1	Overview . . . . .	93
5.2	Requirements of Wireless Streaming Applications . . . . .	96
5.3	WBest algorithm . . . . .	100
5.3.1	Assumptions . . . . .	100
5.3.2	Algorithm . . . . .	103
5.3.3	Number of Packet Pairs and Length of Packet Train . . . . .	107
5.3.4	Impact of Errors in Effective Capacity Estimation . . . . .	109
5.3.5	Pre-dispersion and Pre-compression . . . . .	111
5.3.6	Error Detection . . . . .	113
5.4	Experiments . . . . .	114
5.4.1	Experiment Design . . . . .	114
5.4.2	Metrics for Performance Evaluation . . . . .	120
5.4.3	Discussion on Experiment Setup . . . . .	122
5.4.4	Accurate Sending Rate Control in WBest . . . . .	126
5.5	Analysis . . . . .	128
5.5.1	Data Collected . . . . .	128
5.5.2	Case Analysis . . . . .	131
5.5.3	Impacts of Packet Sizes . . . . .	142
5.5.4	Consistency Analysis . . . . .	143
5.5.5	Estimating the Standard Deviation in Available Bandwidth . . . . .	144
5.6	Summary . . . . .	146
<b>6</b>	<b>BROS: Buffer and Rate Optimization for Streaming</b>	<b>149</b>
6.1	Overview . . . . .	149
6.2	Model . . . . .	150
6.2.1	Model Definitions and Assumptions . . . . .	150
6.2.2	Model Validations . . . . .	156
6.2.3	Model Results . . . . .	159

## CONTENTS

---

6.3	Buffer and Rate Optimization for Streaming (BROS) . . . . .	161
6.3.1	Algorithm . . . . .	161
6.3.2	Approximate CDF of Available Bandwidth . . . . .	163
6.4	Evaluation . . . . .	165
6.4.1	Wireless Testbed . . . . .	165
6.4.2	Emulated Streaming (EmuS) Client/Server System . . . . .	165
6.4.3	Experiment Design . . . . .	168
6.5	Results . . . . .	170
6.6	Summary . . . . .	175
<b>7</b>	<b>Future Work</b>	<b>177</b>
<b>8</b>	<b>Conclusions</b>	<b>181</b>
	<b>Bibliography</b>	<b>187</b>

# List of Tables

2.1	IEEE 802.11a, b, and g WLAN Standards . . . . .	25
3.1	Taxonomy of Published Bandwidth Estimation Tools . . . . .	53
3.2	Summary of Bandwidth Estimation Tools . . . . .	55
3.3	IEEE 802.11 Physical Layer Parameters . . . . .	59
4.1	IEEE 802.11 Physical Layer Parameters (DSSS) . . . . .	79
4.2	Errors in the Bandwidth Estimation Model Compared with Simulations . . . . .	81
5.1	Evaluation Cases for Experiments. . . . .	118
5.2	Estimated Available Bandwidth (Median, in Mbps). . . . .	129
5.3	Intrusiveness (Median, in MBytes) . . . . .	130
5.4	Convergence Time (Median, in seconds) . . . . .	130
5.5	Mean and Standard Deviation of Relative Error (ratio), Intrusiveness (MBytes) and Convergence Time (seconds) of All Evaluated Cases . . . . .	141
6.1	Relative Error of Approximating Available Bandwidth using Normal Distri- bution . . . . .	164
6.2	Evaluation Setups . . . . .	169
6.3	Sessions for Each Case . . . . .	169
6.4	Buffer Underflow (fraction) . . . . .	174
6.5	Frame Loss Rate (percent) . . . . .	174
6.6	Buffer Delay (seconds) . . . . .	175

# List of Figures

1.1	The Block Diagram of the Dissertation . . . . .	5
2.1	Bit Error Rate as a Function of Signal-to-Noise Ratio [20] . . . . .	28
3.1	Bandwidth Related Terminology . . . . .	45
3.2	Packets Dispersion . . . . .	48
3.3	Probe Gap Model . . . . .	50
4.1	Throughput Comparison As Distance From Sender to Receiver Increases . .	68
4.2	Link Rate Adaptation Under Ricean Fading . . . . .	68
4.3	Probing, Crossing and Contending Traffic in a WLAN . . . . .	70
4.4	Capacity Estimation Using Packet Pair Techniques in a WLAN . . . . .	71
4.5	A Randomly Generated Topology with 50 Nodes and an AP in the Center .	79
4.6	Bandwidth Estimation Validation for an Ideal WLAN . . . . .	81
4.7	Packet Pair Estimations and CBR Throughput . . . . .	84
4.8	The Impact of Channel Data Rate . . . . .	86
4.9	The Impact of Channel Bit Error Rate (BER) . . . . .	86
4.10	Comparison of RTS/CTS with Basic Access for Achievable Throughput . .	87
4.11	Simulating and Modeling Standard Deviations of Estimation . . . . .	88
4.12	The Impact of Packet Sizes on the Standard Deviation of Bandwidth Esti- mation . . . . .	89

4.13 The Impact of Channel Data Rate on the Standard Deviation of Bandwidth Estimation . . . . .	90
4.14 The Impact of BER on the Standard Deviation of Bandwidth Estimation . . . . .	90
5.1 Network Path with Last Mile Wireless Network. . . . .	101
5.2 A Typical Last Hop Wireless Network. . . . .	105
5.3 Estimating Available Bandwidth using Average Dispersion Rate. . . . .	105
5.4 Relative Error Caused by Effective Capacity Estimation Errors. . . . .	110
5.5 Relative Error Caused by Pre-dispersion/compression . . . . .	113
5.6 An Example of a Sequential Run of Bandwidth Estimation Tools. . . . .	116
5.7 Rate Adaptation Behavior . . . . .	117
5.8 Analysis of Number of Packet Pairs . . . . .	118
5.9 Analysis of Length of Packet Train . . . . .	119
5.10 Typical Clients' Data Rate for Case 6 (Two Contending UDP Traffic) . . . . .	123
5.11 Typical AP's Data Rate for Case 6 (Two Contending UDP Traffic) . . . . .	123
5.12 Typical AP Retries' Data Rate for Case 6 (Two Contending UDP Traffic) . . . . .	124
5.13 Typical Downstream UDP Throughput for Case 6 (Two UDP Contending Traffic) . . . . .	124
5.14 Ping Result for Case 13 (Wireless Rate Adaptation) . . . . .	126
5.15 Ping Result for Case 14 (PSM Case) . . . . .	127
5.16 Evaluation of Sending Rate Control Mechanism in WBest . . . . .	128
5.17 Summary Results for Evaluation Case 0 (Idle Channel). . . . .	132
5.18 Summary Results for Evaluation Case 1 (One UDP Crossing Traffic of 4.6 Mbps). . . . .	132
5.19 Summary Results for Evaluation Case 2 (One UDP Contending Traffic of 4.6 Mbps). . . . .	133
5.20 Summary Results for Evaluation Case 3 (One TCP Crossing Traffic) . . . . .	133
5.21 Summary Results for Evaluation Case 4 (One TCP Contending Traffic) . . . . .	134

*LIST OF FIGURES*

---

5.22 Summary Results for Evaluation Case 5 (Two UDP Crossing Traffic of 2.3 Mbps each) . . . . .	134
5.23 Summary Results for Evaluation Case 6 (Two UDP Contending Traffic of 2.3 Mbps each) . . . . .	134
5.24 Summary Results for Evaluation Case 7 (Two TCP Crossing Traffic) . . . . .	135
5.25 Summary Results for Evaluation Case 8 (Two TCP Contending Traffic) . . . . .	135
5.26 Summary Results for Evaluation Case 9 (One UDP Crossing and One UDP Contending Traffic of 2.3 Mbps Each) . . . . .	135
5.27 Summary Results for Evaluation Case 10 (One TCP Crossing and One TCP Contending Traffic) . . . . .	136
5.28 Summary Results for Evaluation Case 11 (One UDP Crossing of 2.3 Mbps and One TCP Contending Traffic) . . . . .	136
5.29 Summary Results for Evaluation Case 12 (One TCP Crossing and One UDP Contending Traffic of 2.3 Mbps) . . . . .	136
5.30 Summary Results for Evaluation Case 13 (Wireless Rate Adaptation, Range 1-48 Mbps). . . . .	137
5.31 Summary Results for Evaluation Case 14 (Power Saving Mode). . . . .	137
5.32 Summary of All Experiments – Convergence Time versus Error. . . . .	140
5.33 Summary of All Experiments – Intrusiveness versus Error. . . . .	140
5.34 WBest Bandwidth Estimation versus Probing Packet Sizes . . . . .	143
5.35 WBest Bandwidth Estimation versus Crossing/Contending Packet Sizes . . . . .	144
5.36 CDF of Effective Capacity Error . . . . .	145
5.37 CDF of Available Bandwidth Error . . . . .	145
5.38 Estimated Standard Deviation in Available Bandwidth . . . . .	146
6.1 Buffer Model . . . . .	151
6.2 States of the Buffer Model . . . . .	152
6.3 Streaming Rate and Available Bandwidth . . . . .	152
6.4 Comparison of Full and Simplified Buffer Models . . . . .	157

6.5	Typical Rate Adaptation . . . . .	158
6.6	MTBBU and the CDF of the Streaming Rate . . . . .	158
6.7	Required Buffer Size versus MTBBU . . . . .	159
6.8	MTBBU versus Streaming Rate . . . . .	160
6.9	Buffer Size versus Streaming Rate . . . . .	160
6.10	Approximating Available Bandwidth using Normal Distribution . . . . .	164
6.11	Network Path with Last Mile Wireless Network. . . . .	166
6.12	Typical Buffer Size Fluctuation . . . . .	167
6.13	Frame Arrival and Playout . . . . .	167
6.14	BROS Streaming Rate . . . . .	171
6.15	BROS Buffer Size . . . . .	172
6.16	Summary Results for Evaluation Case 1 (Idle Channel) . . . . .	173
6.17	Summary Results for Evaluation Case 2 (Contending Traffic) . . . . .	173
6.18	Summary Results for Evaluation Case 3 (Crossing Traffic) . . . . .	173
6.19	Summary Results for Evaluation Case 4 (Rate Adaptation) . . . . .	174

# List of Algorithms

- 5.1 WBest Algorithm. . . . . 104
- 6.1 Buffer and Rate Optimization for Streaming (BROS). . . . . 162

# Chapter 1

## Introduction

### 1.1 Motivation

The combination of the decrease in price of wireless LAN access points (APs) and the increase in wireless link capacities has prompted a significant increase in the number of wireless networks in homes, corporate enterprises, and academic campuses. The promise of up to 54 Mbps capacity<sup>1</sup> from a wireless AP means that users now expect to have applications such as streaming video that require high bit rates run seamlessly from well-connected wired media servers to wireless media clients. However, since many streaming techniques are primarily designed for wired networks, streaming media applications may perform poorly when running over a complex network environment with a last mile wireless network.

In wired networks, streaming media quality is impacted by packet delay, jitter and packet loss due to network congestion. To mitigate the impact of network congestion, various techniques have been used to improve streaming media quality, such as initial capacity estimation, media scaling and playout buffer optimization. For example, some streaming products such as RealNetworks and Windows Streaming Media use network probes or manually configuration to provide estimates of the underlying network characteristics

---

<sup>1</sup>IEEE 802.11g

prior to making key decisions about the streaming rate for the video stream sent over the network [1, 2]. Media scaling is a method of adjusting the streaming media's data rate in response to the congestion levels in the networks [3, 4]. By adjusting the streaming rate, the application can reduce the impact of congestion in the network. Client side buffering provides the essential functionality of removing jitter effects and playback disruption caused by oscillations in the network bandwidth at the cost of initial start-up delay [1, 5].

Even though these techniques can reduce the degradation in quality when streaming over wireless networks, they do not perform as effectively as in wired networks. The streaming media quality is impacted not only by the network congestion, but also by the wireless network conditions. Recent research shows that the wireless network conditions, such as the wireless link layer rate adaptation, contention, and interference can significantly degrade the performance of streaming media applications by incurring rebuffer events and degraded perceptual quality [6, 7, 8, 9, 10].

Wireless media traits such as shared medium, bursty channel error, attenuation, and fading make media access in wireless data networks more complex than in wired networks. For example, the IEEE 802.11 Wireless LAN standard [11] uses a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) based Media Access Protocol (MAC) with multirate physical layer and Automatic ReQuest for retransmission (ARQ) to reduce and recover from frame loss in the physical layer. While these techniques can reduce the packet loss seen at higher layers, they may impact the performance of streaming media applications in multiple respects.

First, current packet pair techniques used by streaming media systems to estimate capacity to a wireless LAN client [8] are inadequate for providing good bandwidth information. The wireless layer mechanisms, such as the ARQ with random backoff timer and multirate adaptation can cause errors in bandwidth estimations. Moreover, assuming a static bandwidth condition, the packet pair probes used by typical streaming applications can not provide sufficient channel characteristics in wireless networks. Wireless phenomenon such as shifts in received signal strength and dynamic rate adjustments in

the MAC layer capacity wreak havoc on the assumption of static bandwidth. For example, a simple packet pair estimation can not provide the dynamic characteristics of the wireless networks, such as the effective capacity, available bandwidth and variance in available bandwidth.

Second, streaming rate control often does not perform well in wireless networks [8, 12]. For example, as for most media scaling techniques, the streaming rate control based on packet loss or delay may misinform the rate scaling decision and converge slowly in wireless networks [12]. In wireless networks, the packet loss and delay observed at higher layers are different than in wired networks because extra delay and lower packet loss may be caused by ARQ in the MAC layer. In addition, multirate adaptation in wireless networks may also directly change the capacity due to loss and delay not observed at the application layer.

Finally, the initial client-side playout buffer designed according to the characterizations of wired networks may not be large enough to overcome the oscillations in the available bandwidth in wireless networks. The multirate adaptation and ARQ in the wireless MAC layer create more changes in wireless networks. Playout buffer underflow can cause rebuffer events, extra buffer time, and significantly degrade the media quality [8].

In summary, the current techniques of bandwidth estimation, media scaling and playout buffer can not perform as expected in wireless networks, which significantly impacts the performance of streaming media applications. The streaming media application may result in a higher or much lower streaming bit rate than the available bandwidth with an insufficient initial buffer size. Therefore, the high loss rate or low quality streaming and high number of rebuffer events can degrade the perceptual quality of the streaming media applications.

Several researchers have proposed MAC layer, Transport layer [13, 14, 15] and even cross-layer approaches to improve the performance of streaming media applications over wireless networks. For example, cross-layer approaches are proposed to provide Unequal Error Protection (UEP) for streaming traffic in [16, 17, 18, 19]. By applying protection such as Forward Error Correction (FEC) to different frame types to reduce the packet loss

and delay on important video frames, such as I frames, these approaches can improve the quality of streaming video over wireless networks. However, these lower layer or cross-layer approaches usually require modification of the protocol stack or access to the lower layer information, which make them difficult to deploy in real systems. Moreover, these techniques usually provide proactive loss protection, which does not solve the issues of streaming rate selection and playout buffering in wireless networks.

Therefore, this dissertation presents the models, design, implementation, and evaluation of an application layer solution for improving streaming multimedia application performance in IEEE 802.11 wireless networks by using enhanced bandwidth estimation techniques. The solution includes two parts: 1) a new Wireless Bandwidth estimation tool (WBest) designed for fast, non-intrusive, accurate estimation of available bandwidth in IEEE 802.11 networks, which can be used by streaming multimedia applications to improve their performance in wireless networks; 2) a Buffer and Rate Optimization for Streaming (BROS) algorithm using WBest to guide the streaming rate selection and initial buffer optimization. WBest and BROS are implemented and incorporated into an emulated streaming client-server system, *Emulated Streaming (EmuS)*, in Linux and evaluated under a variety of wireless conditions. The evaluations show that with WBest and BROS, the performance of streaming multimedia applications in wireless networks can be significantly improved in terms of multimedia frame loss, rebuffer events and playout delay.

## 1.2 The Dissertation

The goal of the dissertation is to improve streaming media application performance over a network path with a last mile wireless network by utilizing improved bandwidth estimation techniques. The dissertation presents the modeling, design and evaluation of two enhanced modules, the *Wireless Bandwidth Estimation* module and the *Streaming Buffer and Rate Optimization* module. These two modules are designed to be able to work with most streaming systems and provide explicit functionalities for improving the streaming multimedia performance. Figure 1.1 depicts the block diagram of the dissertation. The

two enhanced modules are represented by dark grey blocks in the diagram. The bandwidth estimation module and the streaming buffer and rate optimization module reside on both the client and server. However, the client side optimizer module only performs buffer control functions. The relationships between modules and related network traffic are also illustrated in the block diagram.

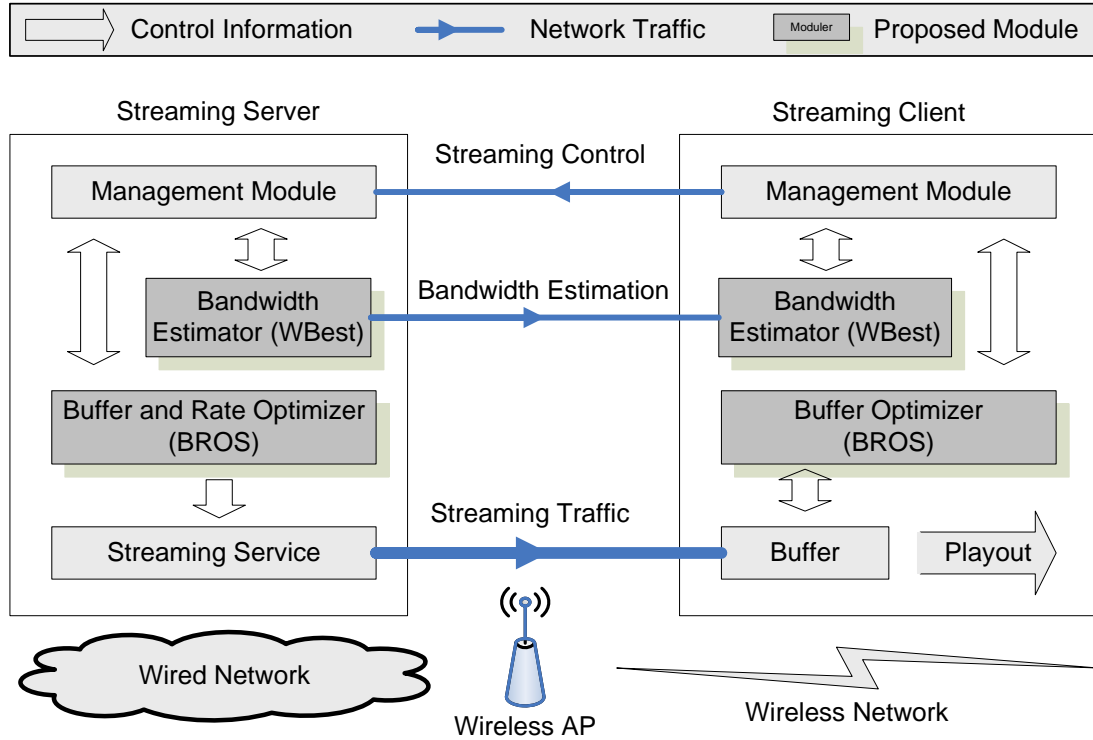


Figure 1.1: The Block Diagram of the Dissertation

The bandwidth estimation module uses WBest to estimate the bandwidth related metrics along the network path from streaming server to the client. Using packet dispersion techniques, WBest provides fast, non-intrusive, accurate estimation of available bandwidth in IEEE 802.11 networks. WBest applies a two-step algorithm: 1) a packet pair technique to estimate the effective capacity of the wireless network; 2) a packet train technique to estimate the achievable throughput and report the inferred available bandwidth. The server side WBest module initializes the packet pairs and packet train and sends them to the

client using UDP. On receiving the packet pairs and packet train, the client side WBest module processes the packets and reports the estimations back to the server. Related control messages and the estimation results are exchanged via a separate TCP control channel. The advantage of WBest is that it does not depend upon search algorithms to detect the available bandwidth but instead statistically detects the available fraction of the effective capacity, mitigating estimation delay and the impact of random wireless channel errors. WBest provides a broad range of bandwidth related information for the wireless networks, such as the effective capacity, available bandwidth, achievable throughput and variance of available bandwidth and achievable throughput. WBest is compared with other popular available bandwidth estimation tools in a wireless testbed under a variety of wireless and network conditions. The evaluation shows that current bandwidth estimation tools are significantly impacted by wireless network conditions, such as contention from other traffic and rate adaptation. On the other hand, WBest consistently provides fast available bandwidth estimation, with overall more accurate estimations and lower intrusiveness than other techniques over all conditions evaluated.

The streaming rate and buffer optimizer module uses the BROS algorithm to select the proper streaming rate and initial buffer size based on the bandwidth information provided by WBest. By taking the available bandwidth distribution into consideration, the server side streaming rate and buffer optimizer module selects the proper streaming rate and buffer size to mitigate the probability of buffer underflow events. The optimized buffer size is then sent back to the streaming client to control the playout buffer. The advantage of the BROS algorithm over existing buffer optimizing approaches using jitter or Poisson arrival models is that it takes the variation of available bandwidth into consideration, which usually has a greater impact on streaming performance than delay jitter in wireless networks. BROS is implemented in a streaming system (EmuS) and evaluated in an IEEE 802.11 wireless testbed under various wireless conditions. The evaluation shows that BROS can effectively select the best streaming rate and optimize the initial buffer size based on wireless network bandwidth conditions, thus achieving lower frame loss rate, fewer buffer

underflow events and lower initial delay than buffers based on static rate selection, static sizing, and jitter removal.

In general, analytical modeling may provide closed-form solutions that are easy to evaluate, but real systems usually have additional complexity and thus are hard to model precisely. Simulations can provide evaluations for the models and techniques in circumstances close to that of the real systems with good repeatability and scalability. However, simulations may not fully represent the complex network systems. Therefore, the combination of modeling, simulation and empirical measurement is used to re-enforce and evaluate our approach in multiple aspects. In the dissertation, the following methodologies are applied for the development and evaluation of WBest and BROS:

- **Analytical Models.** Three mathematical models are created in the dissertation: a model of packet dispersion in IEEE 802.11 wireless networks; the WBest model for performance and error analysis; and a Markov chain buffer model used by BROS.
- **Simulations.** NS-2 simulations are used to validate and evaluate the packet dispersion models. Customized simulation modules including wireless rate adaptation, multi-path fading and error models, are used to simulate realistic wireless network setups.
- **Empirical Measurements.** WBest and BROS are both implemented in the Linux system and evaluated in a wireless testbed under a variety of wireless and network conditions, including crossing traffic, contending traffic, rate adaptation and power saving mode. WBest is evaluated in comparison with typical existing techniques for bandwidth estimation. Additionally, an emulated streaming system (EmuS) is developed to include both WBest and BROS modules to evaluate the improvement of streaming multimedia performance in wireless networks.

### 1.3 Contributions

The main contributions of this dissertation are the design, simulation, implementation and evaluation of improvements in streaming media performance in wireless networks using

bandwidth estimation techniques. The specific contributions of the dissertation include:

1. Review and evaluation of current bandwidth estimation techniques in wireless networks. The applicability of currently publicly available bandwidth estimation techniques in wireless networks are reviewed and discussed. The evaluation shows that current bandwidth estimation tools are significantly impacted by wireless network conditions, such as ARQ, contention from other traffic and wireless rate adaptation. (Chapters 3, 4 and 5)
2. Analytical model of packet dispersion in IEEE 802.11 wireless networks. The analytical model is created to study the behavior of packet dispersion under different wireless network configurations, including ARQ, rate adaptation, contending and crossing traffic, and channel error. The model is validated with NS-2 simulations and empirical measurements in an IEEE 802.11 wireless testbed. (Chapter 4)
3. NS-2 simulator extension of IEEE 802.11 MAC rate adaptation. Receiver Based Auto Rate (RBAR) [20], a MAC layer rate adaptation protocol is re-implemented in NS-2 version 2.27. The documented implementation is available online<sup>2</sup>. (Chapter 4)
4. Wireless Bandwidth Estimation Tool (WBest). WBest is designed for fast, non-intrusive, accurate estimation of available bandwidth in IEEE 802.11 networks. WBest provides comprehensive bandwidth information of the wireless networks, such as the effective capacity, available bandwidth, achievable throughput and variance of available bandwidth and achievable throughput. The evaluation shows that WBest consistently provides fast available bandwidth estimation, with overall more accurate estimations and lower intrusiveness over all conditions evaluated. On average, WBest effectively reduces the average relative error by 82% to 86%, the intrusiveness by 70% to 90%, and the convergence time by 95% to 99%. Implemented as a shared library<sup>3</sup> in Linux, WBest can be easily imported to most applications. (Chapter 5)

---

<sup>2</sup>Downloadable from <http://perform.wpi.edu/downloads/#rbar>

<sup>3</sup>Downloadable from <http://perform.wpi.edu/downloads/#wbest>

5. Playout buffer model for available bandwidth oscillation in wireless networks. We create a Markov chain model for client side playout buffer size for streaming multimedia applications as a function of streaming rate and the distribution of available bandwidth in the wireless network. A primary advantage of the buffer model over existing jitter or Poisson arrival models is that it takes into consideration the variation in available bandwidth. (Chapter 6)
  
6. Buffer and Rate Optimization for Streaming (BROS) algorithm. BROS is designed to select the proper streaming rate and initial buffer size based on the available bandwidth estimations using WBest to reduce the buffer underflow events, buffer delay, and improve the frame loss rate for multimedia streaming application over wireless networks. The evaluation shows that BROS can effectively select the streaming rate and optimize the initial buffer size based on wireless network bandwidth conditions, thus achieving lower frame loss rate, fewer buffer underflow events and lower initial delay than other algorithms evaluated. For example, comparing BROS with similar streaming rate sessions with fixed and jitter removal buffer algorithms, BROS can reduce the buffer underflow events and the frame loss rate by close to 100%, and reduce the total buffer delay by about 80%. (Chapter 6)
  
7. Emulated Streaming (EmuS) client-server system with WBest and BROS support. We develop an emulated streaming server/client system, called Emulated Streaming (EmuS) in Linux with initial buffer and rate selection features. The streaming server supports multiple encoded layers and configurable playout buffer sizes. The performance information, such as buffer underflow, frame rate, frame loss, retransmission, etc. are reported during playback. By including WBest and BROS, we use EmuS to evaluate streaming multimedia performance under different wireless conditions. The source code is available online<sup>4</sup>. (Chapter 6)

---

<sup>4</sup>Downloadable from <http://perform.wpi.edu/downloads/#wstream>

## 1.4 Roadmap

The remainder of this dissertation is organized as follows: Chapter 2 provides background knowledge to the work in this dissertation; Chapter 3 discusses related research in the areas of streaming, bandwidth estimation and wireless networks; Chapter 4 presents the in-depth study and modeling of the packet dispersion techniques in IEEE 802.11 wireless networks; Chapter 5 presents the design, analysis and evaluation of WBest; Chapter 6 presents the streaming buffer model and the design, analysis and evaluation of BROS; Chapter 7 outlines possible future work; and finally Chapter 8 summarizes this dissertation and draws conclusions.

## Chapter 2

# Background

This chapter reviews the fundamental techniques and terminologies that are referred to in the thesis. Section 2.1 reviews the streaming multimedia techniques, such as the commercial applications, media scaling, and quality metrics. Section 2.2 reviews wireless network techniques, including general characteristics of wireless media, and the IEEE 802.11 Wireless LAN (WLAN) family.

### 2.1 Streaming Multimedia

Streaming multimedia is the technique that delivers media data directly from a server to client and starts the playout of the media as it is being received. This results in relatively short waiting times of only a few seconds buffering before the media starts playing at the receiver. With the support of a streaming protocol, clients can perform a series of playback controls, such as pause, fast forward, and rewind on the media content without downloading the entire media clip.

Unlike typical Internet traffic, streaming multimedia is sensitive to delay and jitter, but tolerates some data loss. Additionally, streaming multimedia typically prefers a steady data rate rather than the bursty data rate associated with window-based network protocols. Hence, streaming multimedia applications often use UDP rather than TCP. However, commercial streaming applications do use TCP protocol in some special cases, for example,

in the presence of firewalls.

This section reviews typical streaming applications, by using the example of Windows Media Services, as well as typical media scaling techniques and general streaming quality metrics that are used in the thesis.

### 2.1.1 Streaming Multimedia Applications

Given the fact that about 72% of the available video content on the Web are based on three major commercial streaming media technologies: Real Player, Windows Media Services, and Apple QuickTime [21], it is important to understand the behavior of commercial streaming applications. It is difficult to ascertain the exact streaming implementations hidden in the commercial applications due to the insufficient information available for the implementation of those commercial applications. However, to provide some fundamental understanding of the common features and behaviors of commercial streaming applications, this section reviews Windows Media Services based on Microsoft online documents and the results of previous research.

#### Streaming Protocols

Windows Media Services can stream media over several application-layer protocols: Real-time Streaming Protocol (RTSP), Microsoft Media Server (MMS) Protocol, Hypertext Transfer Protocol (HTTP), and multicast streaming. The MMS protocol, which is the proprietary streaming media protocol developed for earlier versions of Windows Media Services, is still in the most recent version of the software only for compatibility reasons. For RTSP and MMS the underlying transport protocol can either be the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP). The actual protocol used is chosen through a process called protocol rollover [22] based on server and/or client configuration.

For example, when Windows Media Player 9 Series attempts to connect to the server using a URL with an **mms://** prefix, the server automatically uses RTSP. If Fast Cache

is enabled on the server (the default condition for all new publishing points), the server tries to connect to the client using RTSP with TCP-based transport (RTSPT) first. If the Player does not support that protocol, then the server attempts to connect using RTSP with UDP-based transport (RTSPU). If that connection is also not successful, the server then attempts to connect using the HTTP protocol if the WMS HTTP Server Control Protocol plug-in is enabled. If Fast Cache is not enabled, the server first tries to connect to the client using RTSPU, then RTSPT, and finally HTTP. Some clients may be unable to connect using certain protocols for various reasons such as player version and network firewall settings.

Similarly, Real Player supports RTSP, Progressive Networks Audio (PNA) protocol, HTTP, and multicast streaming, while the PNA in the latest Real Server is only for compatibility with older versions of RealPlayer. For Apple QuickTime, RTSP, HTTP, and multicast streaming are supported. For both of Real Player and Apple Quicktime applications, the transport protocol can be either TCP or UDP, and certain algorithms are used to decide which transport protocol is used, respectively.

### **Playout Buffer**

Streaming multimedia is usually delivered to users over a wide range of network qualities and connection speeds. To mitigate errors caused by data being lost, delayed, and incompletely received packets during data transmission, Windows Media Services maintain a initial buffer with a default size of five seconds of playback equivalent data [23]. This initial buffer can be used to reduce jitter and recover lost packets by fast retransmission. The initial buffer size in seconds can also be customized at the player side to accommodate different network conditions. In general, with a large buffer, streaming applications can gain more tolerance towards unpredictable changes of network condition, while having the disadvantage of increasing the starting playback delay.

To reduce the starting delay caused by the initial buffer, Windows Media Services (for versions greater than Windows Media Services 9 Series) provides a new function, Fast

Start [23], which causes data to be sent faster than the actual encoding bit rate of a stream in order to quickly fill the buffer. After the buffer is filled, the bit rate returns to normal. Fast Start can reduce initial buffering time and the Player begins playing the stream sooner, but this may also cause network congestion and be TCP-unfriendly during the buffering period [2].

Moreover, network congestion or other network conditions can change. For example, a decrease in capacity in a Wireless LAN (WLAN) may exhaust the buffer during media playback, which forces the client to stop playback and fill the streaming buffer before it can continue the transmission. These unexpected rebuffer events cause media quality degradation.

Similarly, both Real Player and Apple QuickTime apply initial streaming buffer and fast buffering techniques [24, 25].

### **Media Scaling**

To improve the streaming media quality, Windows Media Service deploys Intelligent Streaming [26] to perform media scaling during the network congestion. Intelligent Streaming adjusts the bit rate of the content stream to counteract the changes in available bandwidth, thereby reducing the packet loss in network and ensuring a continuous presentation by reducing the rebuffer events.

Windows Media Services combines Multiple-Bit-Rate Encoding and Stream Thinning techniques to perform Intelligent Streaming. In Multiple-Bit-Rate Encoding, a number of discrete, user-definable audio and video streams are encoded into a single Windows Media clip. The streams are encoded from the same content, but each is encoded at a different bit rate. When Windows Media Player connects to a Windows Media Server to receive a multiple-bit-rate Windows Media file or broadcast stream, the server only sends the set of audio and video streams that is the most appropriate for estimated current bandwidth conditions. In addition, the Windows Media client and server can also decrease the bit rate to accommodate the current bandwidth by reducing media quality. This is referred to as

Stream Thinning in Windows Media Services. The server decreases the video frame rate first, and if the bit rate is still too high, the server stops sending video frames altogether.

Intelligent Streaming uses a series of strategies to modify the bit rate of the stream in response to the change of available bandwidth. As conditions become worse, the server attempts each strategy in the following list of options one by one until the bit rate is optimized for the current bandwidth [26]:

- The server and client automatically estimate the current available bandwidth, and then select and deliver the stream with the most appropriate bit rate.
- During transmission, if the available bandwidth is reduced, Windows Media Services switches to a stream with a lower bit rate. If bandwidth increases, it switches to a stream with a higher bit rate, but never higher than the original bit rate.
- If the bandwidth can no longer support streaming video, Windows Media Services uses Streaming Thinning techniques to degrade image quality to avoid rebuffering. After the server stops sending video frames, it uses Intelligent Streaming to maintain a continuous audio stream. If audio quality starts to degrade, the client reconstructs portions of the stream to preserve quality.

Packet Pair techniques, which will be discussed in detail in Section 3.2, are used to determine the bandwidth that is available for streaming when a client first connects to a server using RTSP or MMS with the UDP protocol [1]. However, the details on the bandwidth estimation and responding mechanisms are not publicly available to general users [2].

Real Player has a similar mechanism, known as SureStream analyzed in [27] and QuickTime also makes adjustments to the bit rate requirements of the stream by altering the quality level [2].

### 2.1.2 Media Scaling

Media scaling is a method of adjusting the streaming media's data rate. Typical media scaling techniques reviewed in this section include Temporal Scaling, Quality Scaling, and Spatial Scaling [28].

#### Typical Media Scaling Methods

Temporal scaling reduces the streaming data rate by decreasing the video frame rate [29]. For example, in Motion Compensated Prediction (MCP) [29], temporal scalability can be provided by strategic placement of reference frames and predicted frames and then selectively decoding the frames. Therefore, by reducing the number of frames that need to be decoded, the video data rate can also be decreased. Temporal Subband Coding (TSB) [30] provides lower frame-rate video by decoding temporal low-pass subbands, giving a natural multiresolution decomposition into frame rates that are halved at each analysis level. Motion-Compensated Temporal Subband Coding (MC-TSB) [31] includes motion compensation prior to the temporal subband coding to reduce the blurring caused by TSB and increase coding efficiency. Conklin *et al.* [32] compare these three major temporal scaling techniques and show MCP provides the best performance in terms of quality and bit rate.

Spatial scaling encodes a video into multiple levels that have the same frame rate and quantization level but different frame sizes. The streaming data rate can be decreased by reducing the video resolution. For example, Naveen *et al.* [33] uses Motion Compensated Multi-Resolution (MCMR) to transmit the High Definition (HD) video to a NTSC receiver. Benzler *et al.* [34] uses multi-resolution streams and choose the appropriate one for the current network conditions.

Quality scaling encodes a video into multiple layers with different quantization accuracy. A dynamic streaming data rate can be achieved by selecting different encoding layers depending on the available network bandwidth. In [35, 36], the server keeps a hierarchical set of streams as multiple layers. The different encoding layers are selected in response to

network congestion to provide TCP-Friendly congestion control to streaming video applications. In [37], the server estimates the TCP-Friendly rate based on packet loss rate and Round Trip Time (RTT) and chooses the appropriate quantization level for encoding.

Previous scaling methods can be used in combination. For example, [38] uses both the temporal scaling and spatial scaling methods for MPEG video coding. As discussed in Section 2.1, most of the commercial streaming applications, such as Real Player and Windows Media Services, use combined scaling methods.

Since this thesis focuses on streaming rate selection but not the media scaling methods, any methods discussed in this section can be combined with our rate selection algorithm.

### Media Scaling Policy

Media scaling is usually performed by the server in response to network congestion based on client feedback. However, the scaling can be triggered by different metrics, such as the packet loss rate, RTT, estimated network bandwidth, or application quality metrics. Moreover, the rate control mechanism on the server side is also critical for a media scaling algorithm.

In [35, 36, 39], the transport layer packet loss and RTT are fed back to the server as the media scaling trigger. For example, in [39], the feedback is sent by the receiver in one second intervals. Loss of more than two packets or a latency increase of 50% over a moving average of the previous five measurements is taken to indicate congestion. If four successive feedback messages indicate no sign of congestion, the streaming data rate is increased.

Some research focuses on rate control that assumes the transmission rate is determined by the network transport protocol, such as TCP and TCP-Friendly Rate Control (TFRC)<sup>1</sup> [40] protocol. For example, in MPEG-TFRCP [37], the TCP-Friendly rate is estimated by a TCP-Friendly rate equation and used to adjust the video streaming data rate. The estimator transmits RTCP feedback packets to the receiver at the regular interval of every five frames ( $5/29.97=0.167$  sec). However, the rate control is performed at predeter-

---

<sup>1</sup><http://www.icir.org/tfrc/>

mine multiples of RTT. The research compares several settings of the control interval such as 8-RTT, 16-RTT, 64-RTT and 96-RTT. The results show that frequent rate control introduces a great variation in the video quality, and therefore worse subjective video quality. On the other hand, for a longer control interval, the video application cannot follow the network conditions and transmits the video data with undesirable quality. Furthermore, as the interval becomes longer, users wait a longer time for a satisfactory video presentation. Finally, the research concludes that either 16-RTT or 32-RTT control interval is preferable for MPEG-TFRC to maintain a TCP-Friendly share and good perceived video quality. Streaming Media Congestion Control protocol (SMCC) [41] is an adaptive media streaming congestion management protocol in which the connection's packet transmission rate is adjusted according to the dynamic bandwidth share of the connection. The bandwidth share of a connection is estimated using algorithms similar to those introduced in TCP Westwood [42], which is assumed to be a TCP-friendly rate. Research [43, 44] also performs rate control for layered video streams based on the knowledge of the maximum available bandwidth. [43] develops a heuristic real time algorithm for adaptive coding rate control based on the maximum available bandwidth, while [44] uses control theory to help select the best rate. Both of [43] and [44] are designed to work over either TCP or TFRC transport protocols and assume the network transmission rate is determined by a TCP-Friendly rate.

Delgrossi *et al.* [28] monitor the video frame's packet arrival and when the number of lost or late packets exceeds a threshold, a scale down message is sent to the sender side. However, since this monitoring can not provide any information about the termination of congestion, the research simply scales up the stream when a certain time span after the previous scale down has elapsed. The research in [45] proposed a content-based video adaption (CBVA) that uses a priori information from the video stream, such as the frame information, to drive the adaptation policy. The CBVA combines quality and frame rate adaptation and the adaptation policy is guided by the principle of correlated priority between frame rate and quality. In some specially designed systems, such as in the Multi-

media Database System described in [46], the server adjusts the streaming quality based on the buffer utilization on the client side to reduce the data rate sent over the network.

For commercial streaming applications, the scaling mechanisms are not publicly available. However, several measurement research characterizes the congestion responsiveness of these applications. Chung *et al.* [27] measure the media scaling behavior of Real Player by using a Token Bucket Filter to emulate Internet congestion. The results show that most RealVideo UDP streams respond to Internet congestion by reducing the application layer encoding rate, and streams with a minimum encoding rate less than the fair share of the capacity often achieve a TCP-Friendly rate. Furthermore, the TCP API hides network information, such as loss rate and round-trip time, making it difficult to estimate the available capacity for effective media scaling. One result from this research is that it takes more than 20 seconds for Real Player to adjust its rate while streaming over TCP. Nichols *et al.* [2] uses a testbed to measure the congestion responsiveness of Windows Media Services, showing that Windows Media streaming is responsive to available capacity, but it is often unfair to TCP.

In wired networks, the scaling control mechanisms and related network metrics described above are efficient for detecting network condition changes and triggering media scaling to maintain good perceived quality. However, these mechanisms may not be as efficient as in wired network due to the following reasons:

- In wireless networks, as discussed in Section 2.2, the delay and packet loss are not only caused by network layer congestion, but also by changes in the wireless physical layer and MAC layer conditions, such as bursty errors, dynamic rate adaptation, MAC layer contention, etc. Packet loss and RTT alone no longer provide accurate congestion information.
- These wired mechanisms usually depend on client feedback information to adjust the streaming data rate because the assumption is that congestion often occurs in one direction, such as the downstream direction. However, in wireless networks, the wireless medium is shared by both upstream and downstream traffic. The feedback

could suffer as well when the wireless network conditions change, thereby delaying the media scaling actions.

- Given the fact that TCP and TFRC do not perform well [47, 13] due to the MAC layer contention and retries as discussed in Section 2.2, the TCP-Friendly rate does not accurately estimate the available bandwidth in wireless networks.

The streaming data rate is expected to scale down before it overloads the network bottleneck so that it will not unduly contribute to the congestion. In wired drop-tail networks, packet loss is usually caused by network congestion. A packet loss rate increase usually indicates that network is already congested, and is not sufficient to predict a change in network condition before it happens. Since bandwidth estimation techniques can provide an early indication of network condition change, they can be used as a better trigger mechanism for media scaling control.

Thus, our approach to control media streaming rate is based on enhanced bandwidth estimation techniques, which are discussed in detail in Chapter 5 and 6.

### 2.1.3 Performance Metrics of Streaming Multimedia

In general, multimedia content can tolerate some loss. However, packet loss, combined with packet delay and jitter in computer networks still impacts the streaming media quality. Other factors specific for streaming multimedia, such as rebuffer events and buffering time can also impact the user perceived quality. To evaluate the benefits of bandwidth estimation techniques on streaming multimedia over wireless networks, it is necessary to measure the streaming media quality. Since this research is mainly focused on the application and network performance, some video level quality measurements, such as Peak Signal Noise Ratio (PSNR), Video Quality Metric (VQM), and Subjective Measurement, are not appropriate. This section defines the terminologies used to measure media quality that will be used throughout the thesis.

**Service Rate**

Streaming multimedia data rate is determined by the content encoding rate and the available bandwidth, or service rate [48]. The service rate of a streaming session directly affects the PSNR of the streaming video [49], which has been shown to be related to user perceived quality [50].

**Rebuffer Event**

When network conditions change, a streaming media player might exhaust its buffered packets in spite of the initial buffering. The media player may pause the playback until it has the buffer filled up again. The number of rebuffer events reflects the network condition and can be used as a negative indicator of streaming media quality [48].

**Buffering Time**

The video is paused during a rebuffer event until the rebuffer is done. The duration of the rebuffering period varies based on the network condition. The longer the rebuffering period, the worse the streaming video performance [48]. Therefore, the total buffering time, which includes initial buffering time and the rebuffering time, or the average rebuffering time, can be used as indicators of streaming media quality.

**Application Packets Loss**

Application layer packet loss, which does not include those packets recovered by Forward Error Correction (FEC) or successful retransmission, prevents the media frame from being decoded correctly during playback, and therefore degrades video quality. Application layer packets that arrive late at the client may also be considered as lost. The number or the fraction of application packets lost during streaming can be used as indicators for streaming media application quality [48]. For example, the “reception quality” in Windows Media Player is defined as the percentage of packets that were not lost during the last 30 seconds [51].

## 2.2 Wireless Network

Wireless networks had been widely deployed over the last few decades. Most of the protocols and applications that were developed for wired networks have been transferred to wireless networks as de facto implementations. However, the wireless characteristics that differ from wired networks may impact the performance of these applications in wireless networks. To understand these characteristics of wireless networks, this section provides a general review of the wireless networks and an introduction to a popular wireless network, Wireless Local Area Network (WLAN).

### 2.2.1 Overview of Wireless Networks

In general, all wireless networks share similar physical characterizations due to the nature of the radio medium, such as high bit error rate caused by attenuation, interference, fading, and collisions. However, a variety of network standards are focusing on distinct purposes and operating environments and the design can vary significantly. To provide the background knowledge of wireless networks, this section summarizes the characteristics of wireless medium and general categorization of current existing wireless networks.

#### Characteristics of Wireless Media

The most important characteristics of wireless radio medium that differ from the wired network are as follows [52].

- **Shared Medium.** Compare with the wired media, wireless medium has natural broadcasting. Therefore, all the wireless transmissions share the same medium and wireless supports only half-duplex operation. Moreover, the shared medium causes more collisions and interference over the air, which can further degrade the network performance. Finally, the shared medium also makes it impossible to increase the capacity by adding media as in a wired network. With the wireless medium, the network is restricted to a limited available band for operation, and can not obtain new bands

or duplicate the medium to accommodate more capacity.

- Propagation. Wireless radio transmissions that propagate over the air expects attenuation, reflection, diffraction and scattering effects. The multipath fading caused by these effects results in time varying channel conditions, such that the received signal power varies as a function of time.
- Bursty channel errors. Due to the attenuation, interference, and fading effects, the wireless network expects a higher Bit Error Rate (BER) that can be  $10^{-3}$  or even higher.
- Location dependent carrier sensing. In wireless networks, such as the wireless LAN, the wireless performance is effected significantly by the location. For example, the hidden terminal and exposed terminal problem may impact the wireless performance significantly. A hidden terminal is one that is within the range of the intended destination but out of range of the sender. Therefore, collisions may happen at the destination if the sender and the hidden terminal transmit at the the same time because they can not detect each other. Similar, an exposed terminal is one that is within the range of the sender but out of interference range of the destination. A sender may unexpectedly backoff when an exposed terminal is transmitting, even if that transmission will not collide with the sender's transmission at the destination.

These wireless characteristics may degrade the wireless network performance extensively. Therefore, most of the wireless network standards implement a variety of error recover mechanisms, such as the Forward Error Correction (FEC), Automatic ReQuest for retransmission (ARQ) and rate adaptation, which are discussed in Sections [2.2.2](#).

### **Wireless Network Categorization**

The general way to categorize wireless data communication networks is based on the coverage range.

- Wireless Personal Area Networks (WPANs)

WPANs are small networks operating within a confined space, such as an office workspace or room within the home. The coverage range is usually less than 30 feet. For example, BlueTooth, which is defined under IEEE 802.15.1, can provide up to 720 Kbps capacity over less than 30 feet distance. Ultra Wideband (UWB, defined in IEEE802.15.3a), which is still under development, is designed to provide up to 480 Mbps throughput over a short distance [53].

- Wireless Local Area Networks (WLANs)

WLANs have broader range than WPANs, typically confined within office buildings, restaurants, stores, homes, etc. WLAN has become the most popular wireless data communication techniques as the production of the WLAN standards, such as IEEE 802.11 standard family, which is reviewed in detail in Subsection 2.2.2

- Wireless Metropolitan Area Networks (WMANs)

WMANs cover a much greater distance than WLANs, connecting buildings to one another over a broader geographic area. For example, the emerging WiMAX technology (802.16d today and 802.16e in the near future) will further enable mobility and reduce reliance on wired connections. Typical WMANs have a throughput up to 10-20 Mbps and cover a distance of approximately several miles[53].

- Wireless Wide Area Networks (WWANs)

WWANs have the broadest coverage range and are most widely deployed today in the cellular voice infrastructure to provide the capability of transmitting data. The most popular WWAN techniques include the currently available cellular 2.5G (Generation) data services, such as General Packet Radio Service (GPRS) and Enhanced Data Rates for Global Evolution (EDGE), and the next-generation cellular services based on various 3G technologies.

Out of these wireless network techniques, WLANs are the most widely deployed wireless networks that are being used for streaming multimedia applications. Therefore, the

research focuses only on WLANs. WLANs implement a highly reliable MAC/link layer by using retransmission, error correction, or link adaptation techniques to reduce the impacts caused by the high loss rate, high dynamic physical layer conditions. These techniques provide the wireless network with better performance for traditional Internet applications, such as Web, Email service and FTP service. However, these techniques may impact rate-based or time sensitive applications, such as streaming multimedia and interactive Internet telephone applications. To fully understand the techniques discussed in this thesis, it is important to review the standards of typical WLANs in the following section.

### 2.2.2 IEEE 802.11 Wireless Local Area Networks (WLANs)

IEEE 802.11 is limited in scope to the Physical (PHY) layer and Medium Access Control (MAC) sublayer. The IEEE 802.11 MAC layer begins with IEEE 802.3 Ethernet standard, while the PHY layer supports a few variations, such as Direct Sequence Spread Spectrum (DSSS), Frequency Hopped Spread Spectrum (FHSS), Orthogonal Frequency Division Multiplexing (OFDM) and InfraRed (IR). The IEEE 802.11 Standard [11] defines a family of Wireless Local Area Networks (WLANs), including 802.11b, 802.11a, 802.11g, etc. A brief comparison of these standards is given in Table 2.1. Note that all the standards use the same MAC layer specification, but different physical layer specifications.

IEEE 802.11 standards support both the infrastructure network topology and ad-hoc network topology. In an infrastructure network, there is a fixed infrastructure that supports communication between mobile stations and fixed stations via an Access Point (AP). Conversely, in an ad-hoc network, there is no fixed infrastructure. The mobile stations

Table 2.1: IEEE 802.11a, b, and g WLAN Standards

Standard	Maximum Data Rate	Frequency	Modulation Scheme
IEEE 802.11	2 Mbps	2.4 GHz	FHSS/DSSS/IR
IEEE 802.11a	54 Mbps	5 GHz	OFDM
IEEE 802.11b	11 Mbps	2.4 GHz	DSSS with CCK
IEEE 802.11g	54 Mbps	2.4 GHz	OFDM/DSSS

communicate directly with each other without the use of an AP. Ad-hoc networking is out of the scope of this thesis and will not be covered in this review.

### **IEEE 802.11 Distributed Coordination Function (DCF)**

In IEEE 802.11, the main mechanism to access the medium is the distributed coordination function (DCF), which is a random access scheme based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The standard also defines the optional Point Coordination Function (PCF), which is a centralized MAC protocol that uses a point coordinator to determine which node has the right to transmit. DCF is a mandatory component in all IEEE 802.11 compatible products, while PCF is an optional component and is not widely implemented. Therefore, the DCF access function is widely assumed [54] in most of cases. In this thesis, we also limit our investigation to the DCF scheme only.

### **CSMA/CA Mechanism**

DCF defines two techniques for frame transmission: the default two-way handshake, referred to as basic access mechanism, and an optional four-way handshake mechanism. In the basic access mechanism, a station that wants to access the channel monitors the channel to determine if another node is transmitting before initiating the transmission of a new frame. If the channel is idle for a distributed interframe space (DIFS), the frame is transmitted. Otherwise, the station defers the transmission for a random backoff time. The receiving station checks the CRC of the received frame and if the CRC is correct, the station sends an acknowledgment frame (ACK) after a period of time called the short interframe space (SIFS).

The four-way handshake mechanism is used to mitigate the hidden terminal problem. The four-way handshake mechanism involves the transmission of the request-to-send (RTS) and clear-to-send (CTS) control frames prior to the transmission of the actual data frame. A successful exchange of RTS and CTS frames attempts to reserve the channel for the time duration needed to transfer the data frame under consideration. On receiving an RTS frame, the receiver responds with a CTS frame after a SIFS time. After the successful

exchange of RTS and CTS frames, the data frame can be sent by the transmitter after waiting for a SIFS interval. If the CTS frame is not received within a predetermined time interval, the RTS is retransmitted following the backoff rules as specified in the basic access procedures described above. The RTS and CTS frames carry information about the time period of the data frame to be transmitted. All stations receiving either RTS/CTS, set a network allocation vector (NAV) containing information to indicate the period of time in which the channel will remain busy. Therefore, when a node is hidden from either the transmitting or the receiving node, by detecting just one frame among the RTS and CTS frames, it will appropriately delay further transmissions to avoid collisions.

### **Exponential Backoff Timer**

An Exponential Backoff Timer is used in DCF for deferring the data packets and RTS packet transmission. The timer is decremented only when the medium is idle and it is frozen when the medium is sensed busy. The slot size of the backoff timer is denoted by the time needed by any node to detect the transmission of a packet by any other node. At each frame transmission, the backoff time is uniformly chosen in the range  $(0, W - 1)$ . The value  $W$  is called the contention window and depends on the number of failed transmissions for a frame, i.e., for each packet queued for transmission, the contention window  $W$  takes an initial value  $W_{min}$  that doubles after each unsuccessful frame transmission, up to a maximum of  $W_{max}$ . The contention window remains at  $W_{max}$  for the remaining attempts. In addition, to avoid channel capture, a node must wait for a random backoff time between two consecutive frame transmissions, even if the medium is sensed idle in the DIFS time.

### **MAC Layer Retransmission**

The IEEE 802.11 DCF MAC layer retransmits RTS and DATA frames a number of times based on the frame size. The IEEE 802.11 standard suggests that the transmission attempts for the frame with a size less than the RTS Threshold is seven, and for the frame with a size larger than RTS Threshold is four. The RTS Threshold parameter is also used as an indicator of the utilization of RTS/CTS mechanism. If the DATA packet is smaller

than the RTS Threshold, the frame is considered as a short frame, and can be transmitted without the RTS/CTS exchanges. Moreover, if a station has an RTS Threshold value greater than the maximum allowed MTU, the RTS/CTS mechanism is simply disabled. Then all DATA frames are retransmitted following the short frame retry limit.

### IEEE 802.11 Multirate Physical Layer

The IEEE 802.11 medium access protocols provide support for multirate physical layer modulations. For example, the Extended Rate PHY (ERP) of IEEE 802.11g supports the payload data rates of 1 and 2 Mbit/s using DSSS modulation, the payload data rates of 1, 2, 5.5, and 11 Mbit/s using DSSS modulations, and additional payload data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbit/s using OFDM modulation.

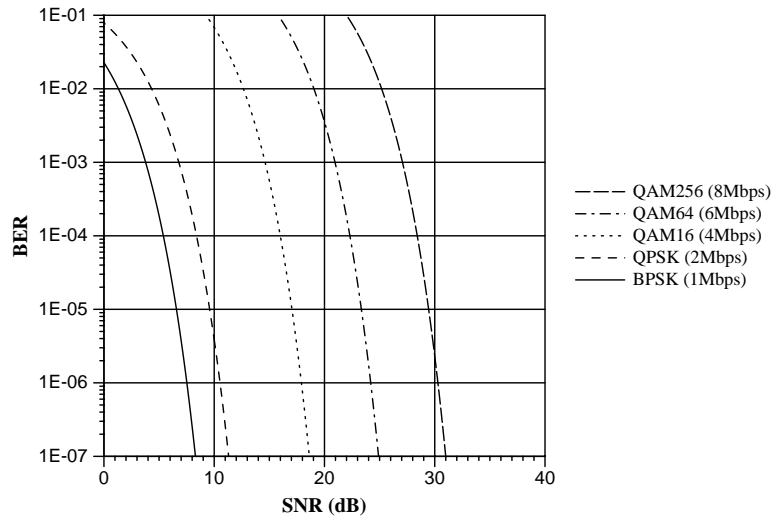


Figure 2.1: Bit Error Rate as a Function of Signal-to-Noise Ratio [20]

Figure 2.1 [20] shows the relation between the Bit Error Rate (BER) and Signal-to-Noise Ratio (SNR) for several modulation schemes and data rates. For a given SNR, the modulation scheme with a higher data rate has a higher BER. By adapting the data rate with different modulation schemes under different wireless network conditions, a low BER and therefore better performance can be produced. However, wireless rate adaptation

results in a dynamic capacity changes in wireless networks, which may impact the performance of rate-based applications, such as the streaming multimedia and Internet telephony over wireless network.

The rate adaptation mechanisms are based on either sender's inference or receiver's feedback of the current channel conditions. The adaptation schemes can be either SNR-based (few implementations) or statistics-based, such as number of retries, packet error rate (PER) or throughput based [55]. For instance, the scheme designed in [56] uses the statistical data of sender retries and the scheme in [20] uses the SNR data feedbacked from the receiver. Chapter 3 reviews the related rate adaptation mechanisms in detail.



## Chapter 3

# Related Work

This chapter reviews the research work related to the work in this thesis. Three corresponding research areas are covered, streaming multimedia performance, bandwidth estimation techniques and wireless network performance.

### 3.1 Streaming Multimedia

As discussed in Section 2.1, streaming multimedia quality is impacted by packet delay, jitter and loss due to the network congestion or other changes in network conditions. To mitigate the impact on quality by the network, various techniques have been used to improve streaming media quality, such as buffer optimization, streaming rate selection. This section reviews the research work in buffering, streaming rate selection, and performance study for streaming multimedia over wireless networks.

#### 3.1.1 Streaming Buffer

To provide better performance for streaming multimedia over best effort networks, such as the Internet and wireless networks, buffer techniques are often used on the server side, network (caching and proxy), and on the client side [57]. Client side buffering techniques play an important role in streaming multimedia. Generally, client side buffering provides the essential functionality of removing the jitter effects and playback disruption caused by

oscillations in the transmission rate at the cost of initial start-up delay [1, 5]. The oscillations in transmission rate may be caused by transport protocols, such as TCP and TFRC that apply the Additive Increase and Multiplicative Decrease (AIMD) based congestion control, the network congestion, or the connection rate adaptation in a wireless network.

Client side buffers can prevent playback disruptions when the available bandwidth is temporarily below the streaming data rate, unless the buffer is also empty [58]. As mentioned in Section 2.1.3, the number of rebuffer events, or the number of disruptions during playback is a critical performance quality metric. In general, the larger the buffer is, the lower the probability the buffer will underflow. However, the initial startup delay is also an important quality metric, especially for real time and/or interactive applications.

For non-realtime or non-interactive streaming applications, buffer overflow is not a critical issue because disk and memory capacity are outpacing the growth in bandwidth available to single stream flow [59]. However, buffer overflow is an issue for mobile devices, such as the PDAs and cellphones, which can still be subject to memory or disk space constraints.

There are a variety of strategies proposed to improve the effectiveness of client side buffering that include slowing down the media playout rate at the client to reduce its consumption rate and help prevent buffer underflow [60, 61, 62, 63, 64, 65, 66] and media scaling techniques as described in Section 2.1.2. Most of that research focuses on the minimum buffer size required in a particular streaming environment, while still keeping a low number of playback disruptions.

Buffer management is also associated with other research topics, such as the smoothing of Variable Bit Rate (VBR) encoding and VCR like functionality on the client side, such as rewinding or indexing to an arbitrary point, which may require additional buffer space at the client [67].

### Buffer Required for Flow Control and Jitter Removing

Zimmermann *et al.* [68] describe buffer underflow and overflow behavior in detail under the ideal network condition for their streaming media system. A simple flow control with stream on/off watermarks are proposed with equations 3.1 and 3.2:

$$WM_O \leq B - (R_N - R_C) \times T_d \quad (3.1)$$

where  $WM_O$  is the buffer overflow watermark,  $B$  is the buffer size,  $R_N$  is the streaming send rate,  $R_C$  is the consumption rate, and  $T_d$  is the network delay.

$$WM_U \geq R_C \times T_d \quad (3.2)$$

where  $WM_U$  is buffer underflow watermark.  $WM_O \geq WM_U$  must hold to make  $B$  the minimum buffer size required for the operating environment. However, in Equations 3.1 and 3.2, both the  $R_N$  and  $R_C$  are assumed as CBR, and all the buffered content is assumed to be playable, which is usually not true in the real world environment.

In [5], the minimal buffering requirement for different adaptation policies is studied. A minimum buffer requirement equation for TCP-Friendly AIMD protocol is developed as a function of the average of the achievable transmission rate, RTT and packet size, as shown in Equation 3.3:

$$\Delta = \frac{\alpha}{18MSS} \times R^2 \times RTT^2 \quad (3.3)$$

where  $\Delta$  is the minimum buffer size for removing the jitter caused by an AIMD protocol, MSS is the packet size, R is the average achievable transmission rate, RTT is the round trip time and  $\alpha$  is the increasing parameter of the AIMD protocol  $AIMD(\alpha, \beta)$ , with the TCP-Friendliness increase/decrease relationship  $\alpha = \frac{3(1-\beta)}{1+\beta}$ .

Research [5] also shows that the adaptation policies that maximize throughput are not suitable for interactive applications with high bit rates or long RTTs due to the long delay

caused by large buffer size.

In addition, there are a variety of strategies proposed to improve the effectiveness of client side buffering by optimizing the buffer size based on jitter removal, such as [69, 70, 71, 72]. To study the buffer size required for removing the jitter in networks, Yuang et al.[60, 73], Girod et al.[61, 64], and Laoutaris et al.[62] present Markov chain models based on Poisson arrival. However, in wireless networks, the arrival of streaming traffic can not be simply modeled as a Poisson distribution because the capacity changes in wireless networks causes a variance in the streaming traffic arrival rate. Therefore, the jitter removal buffer algorithm are not sufficient to avoid buffer underflow in wireless networks.

The buffer optimization approach presented in this thesis is based on estimation of the network condition but does not only consider a TCP-Friendly rate. This allows it to apply to some environments where the buffer underflow is not caused by transmission rate changes of AIMD protocols, such as in some congested wireless networks, where MAC layer rate adaptation dominates the transmission rate changes.

### **Buffer Required for VBR Traffic Smoothing**

Another function of the client side buffer is to smooth the VBR media content. Various techniques are used to reduce the traffic bursts of VBR encoded streaming content. Usually, a modest buffering capability is required at the receiver side [73, 74]. For example, a smoothing by temporal multiplexing algorithm is proposed in [74], whereby the VBR traffic is made more uniform by grouping frames and sending them at the average bandwidth. The smoothing algorithm also controls the critical bandwidth, which is defined as the minimum bandwidth required to guarantee the receiver's buffer will not underflow, to provide a smoother bandwidth and a more effective use of buffering.

Salehi *et al.* [75] propose a work-ahead smoothing algorithm to reduce traffic bursts by modeling the optimal transmission schedule which minimizes both the variance and peak rate at which data is sent to the client for a given buffer size.

Alternate flow control algorithms, such as Multi-Threshold Flow Control (MTFC) [76]

can also be used to improve buffer management and VBR traffic smoothing. By implementing multiple thresholds in the client side buffer, feedback messages are sent back to the server to adjust the sending rate. By combining the threshold control and a consumption prediction module, the algorithm proposes to achieve fewer rate adjustments and a higher buffer utilization.

Jenkac *et al.* [77] analyze the behavior of VBR media over VBR channels, such as the variable bit rate caused by wireless Radio Link Control (RLC) layer retransmissions in cellular networks. By modeling the deterministic and random channel conditions, the algorithm determines appropriate initial delays and buffer sizes for streaming video over variable bit rate wireless channels.

The research in this thesis focuses on wireless condition changes caused by MAC layer retransmission and dynamic rate adaptation, which is similar to the condition described in [77]. However, the goal of our buffer algorithm is to choose the optimal buffer size based on the current estimation of the network condition including the available bandwidth and variance of bandwidth, which promises to be applicable to a wider range of network conditions. In practice, the client side buffer may include the extra buffer space required for VBR traffic smoothing in addition to the buffer size optimized for bandwidth.

### 3.1.2 Characterization of Streaming over Wireless Networks

With the development of the streaming techniques and wireless networks, more wireless streaming research has been conducted recently. This section reviews related areas of characterization and performance study for streaming multimedia in wireless networks.

The research from Kuang *et al.* [6, 7] characterizes the Real Media traffic over an IEEE 802.11b wireless network. Multiple network layer data gathered shows that the Real Media performs well for excellent and good channel conditions, and performs poorly for fair and poor channel conditions. The IEEE 802.11b MAC layer retransmission mechanism is able to hide most physical layer burst errors from higher layer protocols, even in the poor channel condition where 67.5% of the media packets sent require at least one retransmission.

Furthermore, the Real Media application layer NACK-based (Negative Acknowledgment based) error control is effective in recovering missing packets. However, the network RTT changes and rebuffer events caused by MAC-layer retransmission and rate adaptation are not studied in their research.

Our recent study [8] presents results from experiments that stream Windows Media Service video over a wireless campus network and analyzes performance across application, network and wireless link layers. Some of the key findings include:

- Wireless LANs, i.e. IEEE 802.11g, make it difficult for streaming video to gracefully degrade as network performance decreases.
- Video streams with multiple encoding levels can more readily adapt to degraded wireless network conditions than can clips with a single encoding level.
- Under degraded wireless network conditions, TCP streaming can provide higher video frame rates than can UDP streaming, but TCP streaming will often result in significantly longer playout durations than will UDP streaming.
- Current techniques used by streaming media systems to determine effective capacity over wireless LANs are inadequate, resulting in streaming target bit rates significantly higher than can be effectively supported by the wireless network.

Ikkurthy *et al.* [78] characterize MPEG-4 traffic over IEEE 802.11b wireless LANs. A tradeoff was found between the packet sizes and the amount of information lost such that small packet sizes affect B and P frames most of the time. Conversely, small packet sizes produce bigger error bursts that affect I frames more. In summary, a packet size of 750 bytes was found as a good compromise.

The effects caused by mobility and shared AP wireless networks are also studied in [9]. By measuring the performance of streaming MPEG-4 video over a mobile wireless network, the authors conclude that streaming media performance can degrade significantly in the presence of user mobility. Furthermore, the performance degradation affects all clients that connect through the same AP, not just those who are mobile.

In summary, recent characterization research [7, 8, 9, 10] shows that the media scaling performance is limited when the optimal streaming rate is not correctly selected over wireless networks. Typical streaming typical rate selection used in media scaling is based on loss rate, round-trip time or a bandwidth estimate to adjust the streaming data rate to reduce the network impact on media performance. However, these measurements do not always provide clear indications of wireless network conditions and target rates for adaptation. For example, research [8, 10] show that retransmissions and rate adaptation at the wireless MAC layer may reduce the loss rate while increasing the round-trip times measured by the applications. Thus, wireless network conditions hidden from the application can cause bad media scaling decisions. Moreover, media scaling action is usually taken during degraded performance and therefore is not effective in avoiding performance degradation.

In addition, widely deployed client side playout buffer techniques adapt infrequently to network characterizations in wireless networks [8]. Most current playout buffer techniques use either a fixed sized buffer or choose the minimum buffer size necessary to remove delay jitter. However, for wireless networks with dynamic capacity changes, insufficient buffer sizes decided by jitter removal or static buffer size can produce an increased number of rebuffer events or substantial delays [7, 8, 10].

### 3.1.3 Streaming Performance Improvement over Wireless Networks

In addition to the media scaling techniques discussed in Chapter 3 and playout buffer techniques, this section reviews the related approaches to improve streaming performance for wireless networks, including novel transport layer protocols, cross-layer approaches, and bandwidth estimation approaches.

#### Transport Protocols

The performance of TCP Friendly protocols in wireless networks can be improved by using wireless loss differentiation techniques, such as in Cen et al [79]. This can allow a streaming

application to select the proper rate according to the TCP Friendly rate for the observed wireless network conditions. Using similar techniques, Chen and Zakhor [80] and Yang et al. [81] propose new transport protocols that use a TCP Friendly rate in a wireless networks to control the streaming rate.

In addition, Chen *et al.* [15, 80] propose MULTFRC that establishes multiple TFRC connections to achieve a higher utilization of the wireless networks. MULTFRC is validated via NS2 simulation and a 1xRTT (Radio Transmission Technology) CDMA network. However, the RTT increase may also cause performance degradation for some interactive streaming media applications. Other research targets streaming performance by lowering the network delay. For example, [13] presents a rate estimation method by modeling the optimal RTT in ad-hoc wireless networks. NS2 simulations show that RETFRC can significantly reduce the RTT in ad hoc networks, while still keeping a fair throughput.

However, as discussed by Kazantzidis and Gerla [12], using a TCP Friendly rate control provides only “trial-and-error” scaling, which is unreliable, converges slowly and cannot be used for initial streaming rate selection.

### **Cross-layer Approach**

Cross-layer approaches [82, 83] take the advantages of MAC and physical layer information, such as the rate adaptation, amount of forward error correction and retransmissions, to control the selection of the streaming rate. For example, Kazantzidis and Gerla [12] propose a link-network feedback architecture to provide cross-layer information for the streaming media applications to adapt their streaming rates. The research in [84] proposed a layered video encoding method that incorporates MAC layer rate adaptation to provide robust multicast streaming over IEEE 802.11 wireless networks. Thus each client in the same network receives different quality of video based on the current wireless connection bit rate. In addition, Yang et al. [85] combine the cross layer and TCP Friendly rate control approach to propose a new protocol that utilizes link layer loss information to improve the TCP Friendly rate control.

Alternative crossing layer approaches are proposed to provide Unequal Error Protection (UEP) for streaming traffic. For example, in [16, 17, 83], application layer information, such as the frame type, is made available to link layer. Therefore, by applying different protection to different frame types to reduce the packet loss and delay on important frames, such as I frames, these approaches can improve the quality of streaming video over wireless networks.

Li *et al.* [66] present a method to improve the performance by combining optimal power control policy and optimal playout control policy. For example, the transmitter may increase its power to overcome the channel interference and/or other impairment and successfully push packets to the receiver buffer, while the receiver buffer may slow down its playout rate in order to extend the time until underflow. The results show that the heuristic has only a slight performance loss as compared to the optimal joint power-playout control policy over the entire network condition range studied in the investigation.

However, cross-layer approaches require modifications to both end hosts and to protocol stacks, which may involve with multiple vendor implementations, thus making them difficult to deploy.

### **Bandwidth Estimation**

Bandwidth estimation approaches use application measurements to guide rate selection. Commercial streaming media applications, such as Windows Media Service, uses packet pair techniques to estimate the capacity and choose an appropriate streaming rate [1]. Most recent research from Beek *et al.*[86, 87, 88] applies packet pair or receiver side statistical bandwidth estimation techniques to guide the rate selection in wireless networks.

The advantage of the bandwidth estimation approach is that it usually does not depend upon lower layer information or new protocol stacks, thus can be more easily to deployed. In addition, with careful design, bandwidth estimation can avoid the “trial-and-error” problem caused by a TCP Friendly approach to rate selection. However, Li *et al.*[89], Lakshminarayanan *et al.*[90], and Angrisani *et al.*[91] show that traditional band-

width techniques designed for wired networks can not accurately estimate the bandwidth in wireless network. Moreover, these techniques usually provide capacity estimation, while streaming rate selection needs a broader range of bandwidth information, such as the available bandwidth and the variance in available bandwidth.

## 3.2 Bandwidth Estimation

Bandwidth estimation refers to the end-to-end measurement of bandwidth-related metrics, such as capacity, available bandwidth and bulk TCP transfer capacity, performed by the end hosts of a path without requiring administrative access to intermediate routers along the path. Several applications can benefit from knowing the bandwidth characteristics of their network paths. For example, peer-to-peer applications, overlay networks, Content Distribution Networks (CDN), intelligent routing systems, end-to-end admission control, and multimedia streaming applications can all benefit from bandwidth estimation techniques [92].

As discussed in Section 2.1, streaming multimedia applications usually prefer to be rate-based, and often use UDP with higher layer congestion control mechanisms. Traditional congestion control mechanisms, which use a measured increase of packets/frames lost and/or delay as the indicators of congestion in the network are not sufficient for the streaming applications that require explicit rate based congestion control mechanisms. Loss rate and RTT only provide some indicators of congestion, but do not provide the clear extent of the congestion in the network. For example, the applications are not able to know the amount of traffic that can be sent out without causing congestion. Therefore, it is hard for the streaming applications to make the proper congestion control decisions. Moreover, for streaming applications in wireless networks, the loss rate and delay may not be caused by network congestion, thus this may misinform the streaming systems. Having the knowledge of capacity and available bandwidth can make the congestion control of streaming applications more efficient and accurate.

This section reviews bandwidth estimation related metrics, techniques, taxonomy and

evaluations by extending the bandwidth estimation survey [92] to more recent and wider areas.

### 3.2.1 Bandwidth Related Metrics

The term *bandwidth* is often imprecisely applied to a variety of network throughput related concepts, such as *capacity*, *available bandwidth*, *bulk transfer capacity* and *achievable throughput*. Applications are usually concerned with different bandwidth related metrics. Therefore differentiating these concepts is important for the developing, evaluating and applying bandwidth estimation tools.

#### Capacity

Capacity is defined as the maximum possible bandwidth that a link or end-to-end path can deliver [92]. At the link layer, the transmission rate of each segment is usually fixed and constrained by the physical layer medium and the propagation delay. At the IP layer, each hop, which could be multiple link layer segments, delivers data at a rate lower than its nominal transmission rate due to the overhead of link layer encapsulation and framing. Prasad *et al.* [92] define the IP layer capacity by Equation 3.4:

$$C_{L3} = C_{L2} \frac{1}{1 + \frac{H_{L2}}{L_{L3}}} \quad (3.4)$$

where  $C_{L3}$  is the IP layer capacity,  $C_{L2}$  is the link layer capacity,  $H_{L2}$  is the total link layer overhead, and  $L_{L3}$  is the size of an IP packet.

Given a certain type of link layer network, the authors assume that the link layer overhead is fixed. Therefore, the IP layer capacity of a hop is defined based on the Maximum Transmission Unit (MTU) of the IP layer network.

Furthermore, the authors also define the end-to-end capacity  $C$  as

$$C = \min_{i=1, \dots, H} C_i \quad (3.5)$$

where  $C_i$  is the capacity of  $i$ -th hop, and  $H$  is the number of hops in the end-to-end path. The hop with the minimum capacity is the *narrow link* on the path.

Some link layer technologies, such as IEEE 802.11 WLAN as described in Section 2.2.2, do not operate with a constant transmission rate. The capacity definitions in Equation 3.4 and 3.5 can be only used for those techniques during the time intervals in which the capacity remains constant.

### Available Bandwidth

Available Bandwidth is defined as the maximum unused bandwidth at a link or end-to-end path in a network, which depends on not only the link capacity, but also the traffic load, and is typically a time-varying metric [92].

In [92], the available bandwidth  $A_i$  of a hop  $i$  of a end-to-end link over a certain time interval is given by the unutilized fraction of capacity:

$$A_i = (1 - u_i)C_i \quad (3.6)$$

where  $u_i$  is the average utilization of hop  $i$  in the given time interval, and the  $C_i$  is the capacity of hop  $i$ . By extending the available bandwidth definition to an  $H$ -hop path, the authors define the available bandwidth of the end-to-end path,  $A$ , as the minimum available bandwidth of all  $H$  hops:

$$A = \min_{i=1,\dots,H} A_i \quad (3.7)$$

The hop with the minimum available bandwidth is called the *tight link* of the end-to-end path. The narrow and the tight link are both indicate the bottleneck of a network. However, based on the definitions, they are not necessarily at the same hop [92].

### Bulk Transfer Capacity

Bulk Transport Capacity (BTC) [93] defines a metric that represents a network's ability to transfer significant quantities of data with a single congestion-aware transport connection (e.g., TCP). Thus, the BTC is the maximum long term average throughput obtainable by a single flow of an ideal TCP implementation on an end-to-end network path. The *ideal TCP implementation* here means that the TCP must implement all standard congestion control algorithms specified in IETF RFC 2581 [94]. However, RFC 2581 leaves several implementation details open, so different implementations on these details will yield non-comparable measures of BTC. Therefore, any BTC measurement must specify the details about the congestion control algorithms that are not specified in RFC 2581.

In RFC 3148 [93], the BTC of a end-to-end path is defined as:

$$BTC = data\_sent / elapsed\_time \tag{3.8}$$

where *data\_sent* represents the unique data bits transferred, which does not include header bits or emulated header bits or retransmitted data, and *elapsed\_time* is the measurement interval.

BTC is different from available bandwidth in term of bandwidth metrics. BTC is TCP-specific, but available bandwidth does not depends on a specific transport protocol. Furthermore, BTC takes bandwidth sharing with other TCPs into the consideration, but available bandwidth assumes the average traffic load is constant and estimates the bandwidth available to the additional traffic [92].

### Achievable Throughput

Achievable throughput is defined as the throughput of a host-to-host path under a completely specified set of conditions, such as transmission protocol, end host hardware, operating system, tuning method and parameters, etc. [95]. Achievable throughput is extremely application specific, and thereby represents the throughput that an application in this

specific setting might achieve.

Achievable throughput is different from the available bandwidth because the bottleneck could be in an end host, so achievable throughput may or may not correlate with available bandwidth. In addition, achievable throughput only takes into account the portion of the capacity that can be used by the specific application. For instance, a TCP friendly protocol may yield a lower achievable throughput than the available bandwidth as discussed above, while a UDP application may yield a higher achievable throughput than available bandwidth if it aggressively takes the bandwidth from other TCP-based applications. The definition of achievable throughput is close to the BTC measurement from the bandwidth sharing point of view. However, achievable throughput allows the use of parallel connections and could apply to both transport protocols with or without congestion control, i.e. TCP and UDP protocols.

Achievable throughput can be used as a guideline for local application configurations to fully utilize the available bandwidth without interfering with other traffic. For instance, Jin *et al.* [95] suggest to use parallel TCP streams under certain conditions to achieve a fully utilized available bandwidth.

## Summary

In summary, the general relationship between the terminologies used as bandwidth related metrics can be presented as:

$$Capacity > Achievable\ Throughput \geq Available\ Bandwidth > BTC$$

Figure 3.1 illustrates the relationship with the presence of crossing traffic. The capacity is the maximum possible bandwidth that includes the volume used by the crossing traffic. The achievable throughput could get a higher volume than the Available Bandwidth, which is the  $Capacity - Crossing\ Traffic$ . The reason is that Achievable Throughput considers the bandwidth that may be aggressively taken from responsive crossing traffic, such as TCP. Furthermore, the BTC only considers TCP traffic, which will take the congestion control into consideration. Therefore, the BTC volume is likely to be smaller than Available

Bandwidth since it needs to maintain a TCP-Friendly share with the crossing traffic in the network.

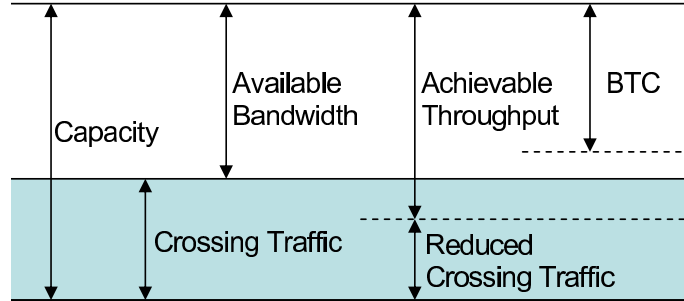


Figure 3.1: Bandwidth Related Terminology

### 3.2.2 Bandwidth Estimation Techniques

Based on the taxonomy and detailed review of current available bandwidth estimation techniques from the survey paper [92], and other related research, this section discusses four major active bandwidth estimation techniques: Variable Packet Size (VPS) probing, Packet Dispersion, Self-loading Probing, Probe Gap Model (PGM) and other related methodologies.

#### Variable Packet Size (VPS) Probing

*VPS* measures the capacity of each hop along an end-to-end path. *VPS* was first proposed by Bellovin [96] and first implemented in *pathchar* [97] by Jacobson in 1997. Subsequent research including *clink* [98], *pchar* [99], and *ACCSIG* [100] improved *VPS* and implemented it in several ways.

The RTT to each hop in the network can be approximated by the summary of three delay components: serialization delays, propagation delays, and queuing delays. *VPS* makes the following assumptions to utilize the RTT information to estimate the capacity of each hop:

- VPS assumes that each hop of a path increases the one-way delay of a packet by a serialization latency given by  $L/C$ , where  $L$  is the packet size and  $C$  is the hop's capacity.
- By sending multiple packets of the same size to each hop of the network, VPS assumes at least one packet will not encounter any queuing delay.
- Propagation delays are independent of the packet size and are constant for each hop.

Therefore, the minimum RTT,  $T_i(L)$ , for a given packet size  $L$  up to hop  $i$  consists of only two parts, the propagation delay and the serialization latency, which can be represented as [92]:

$$T_i(L) = \alpha + \sum_{k=1}^i \frac{L}{C_k} = \alpha + \beta_i L \quad (3.9)$$

where  $C_k$  is the capacity of  $k^{th}$  hop,  $\alpha$  is the propagation delay up to hop  $i$ , and  $\beta_i$  is the slope of the minimum RTT up to hop  $i$  against packets size  $L$ , given by:

$$\beta_i = \sum_{k=1}^i \frac{1}{C_k} \quad (3.10)$$

Therefore, by computing the serialization latency at each hop using Equation 3.9, the capacity of each hop  $i$  can be estimated as:

$$C_i = \frac{1}{\beta_i - \beta_{i-1}} \quad (3.11)$$

The required RTTs for different packet sizes can be measured by sending ICMP messages to the network. VPS sends out ICMP messages with the Time-To-Live (TTL) field of IP header set to force the packets to expire at a particular hop. The router at that hop will discard the expired packet and return a Time Exceeded ICMP messages to the sender, which can be used to measure the RTT to a particular hop.

The VPS model has some advantages compared to other related bandwidth estimation techniques. First, VPS is able to measure the network capacity in an uncooperative en-

vironment, meaning it does not need special software on both the source and destination. Additionally, the VPS technique can measure the entire network path at each hop along the path. Finally, because VPS sends a large number of probing packets and records the minimum traversal times, it can mitigate the effects caused by crossing traffic [101].

However, the VPS model has several important limitations. First, most of VPS tools rely on a functional ICMP implementation at each router along the measured network path. Second, this technique measures bandwidth in a single direction, from the local host to the remote endhost. Moreover, the large number of probing packets generated by the tools adds considerable stress and interference to the network path which implies that the tools might be unscalable, slow and inflexible to bandwidth changes [101]. Finally, recent research [102] shows that the VPS tools may yield significant capacity underestimation errors if the measured path includes store-and-forward layer-2 switches.

### Packets Dispersion

Packet dispersion techniques, such as packet pair or packet train probing, measure the end-to-end capacity of a network path. The packet pair dispersion techniques were first introduced in [103, 104, 105]. Subsequently research and tools, such as *bprobe/cprobe* [106], *nettimer* [107, 108], *sprobe* [101] and *pathrate* [109, 110] improved the packet pair/train dispersion techniques in several ways.

Packet pair dispersion sends two packets with the same size back-to-back into the network. After the packets traverse the narrow link, the time dispersion between the two packets is linearly related to the narrow link capacity. Packet train dispersion probing extends packet pair probing by using multiple back-to-back probing packets, however, the concepts are similar to that with of single pair.

Figure 3.2 [92] illustrates the basic concept of packet dispersion. The most important assumption of packet dispersion techniques is that there is not crossing traffic during the packet pair probing. When packets of size  $L$  with initial dispersion  $\Delta_{in}$  go through the link of capacity  $C_i$ , the dispersion after the link  $\Delta_{out}$  becomes [92]:

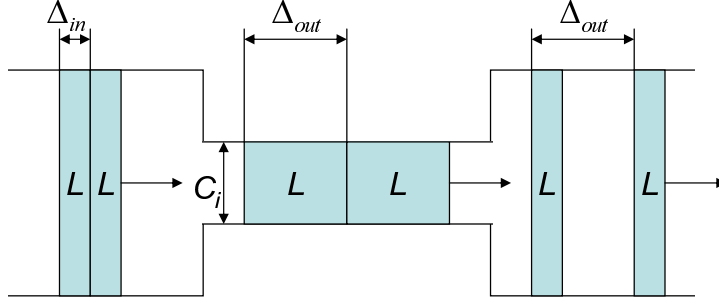


Figure 3.2: Packets Dispersion

$$\Delta_{out} = \max(\Delta_{in}, \frac{L}{C_i}) \quad (3.12)$$

After packets go through each link along an  $H$  hop end-to-end path, the final dispersion  $\Delta_R$  at the receiver is:

$$\Delta_R = \max_{i=1, \dots, H} (\frac{L}{C_i}) = \frac{L}{\min_{i=1, \dots, H} C_i} = \frac{L}{C} \quad (3.13)$$

where  $C$  is the end-to-end capacity. Therefore, the end-to-end path capacity can be estimated from  $C = L/\Delta_R$ .

Compared to other bandwidth estimation techniques, packet dispersion techniques usually have a faster measurement time, and induce less stress on the network path. However, the effects caused by crossing traffic may significantly degrade the accuracy of the link capacity measurement [92]. Several statistical filtering methodologies are proposed to mitigate the effects caused by crossing traffic. For instance, [109] analyzes the local modes of the packet pair dispersion distribution and uses a lower bound of the path capacity measured with long packet trains. [106, 111] proposed methods to detect the local modes in the packet pair bandwidth distribution. [112] uses delay variations instead of packet pair dispersion, and peak detection rather than local mode detection. Another disadvantage of packet dispersion techniques is that it requires the tool to be executed on both end-hosts of the network path, which makes it hard to apply to uncooperative environments.

Packet dispersion techniques have been used in some commercial applications. For instance, as discussed in Section 2.1, Windows Media Service uses a packet train probing of three packets to estimate the end-to-end capacity before beginning the streaming from server to client.

### Self-loading Probe

Self-loading techniques, including the *Self-loading Periodic Streams (SLoPS)* [113] and *Train of Packet Pairs (TOPP)* [114, 115], measure the available bandwidth of the end-to-end network path. Self-loading techniques are also known as *self-induced congestion* from the definition in [116]. There are several tools that implement a variety of self-loading techniques, such as *pathload* [113], *Packet Transmission Rate (PTR)* [117] and *pathChirp* [116]. In addition, some methodologies, such as *Bfind* [118] *Cartouche* [119] and *Pathneck* [120], use similar self-loading probe techniques together with hop by hop delay measurements to locate the bottleneck in an end-to-end network path.

Self-loading techniques probe the end-to-end network path using multiple rate traffic. When the probing rate exceeds the available bandwidth, the probing packets become queued at the tight link router, which results in an increased delay on the receiver side. On the other hand, if the probing rate is lower than available bandwidth at the tight link, the probing packets will go through the tight link without causing an increased delay. By analyzing the packet delay at the receiver, the available bandwidth at the tight link can be obtained at the turning point probing rate, at which the queuing delay starts increasing. The changing of the probing rate can be managed in different ways. For example, *SLoPS* [113] uses a binary search to adjust the probing rate, *TOPP* [114, 115] and *Bfind* [118] use a linearly increased probing rate, while *pathchirp* [116] uses an exponentially increased probing rate.

The self-loading technique assumes FIFO queuing at all routers along the path. Also it assumes that the average rate of cross traffic changes slowly and is constant for the duration of the measurement [121]. However, most of the self-loading tools can detect a change in

the available bandwidth during the measurement by reporting a grey region [113].

The self-loading technique may stress the network path due to self-induced congestion. Furthermore, depending on the implementation, the probing may take a long time to detect the available bandwidth.

### Probe Gap Model (PGM)

The Probe Gap Model (PGM) uses a concept similar to packet dispersion probing. However, PGM measures the available bandwidth by estimating the cross traffic at the tight link. Examples that uses PGM techniques includes *delphi* [122], *Initial Gap Increase (IGI)* [117], *Spruce* [121] and the methods proposed in [123].

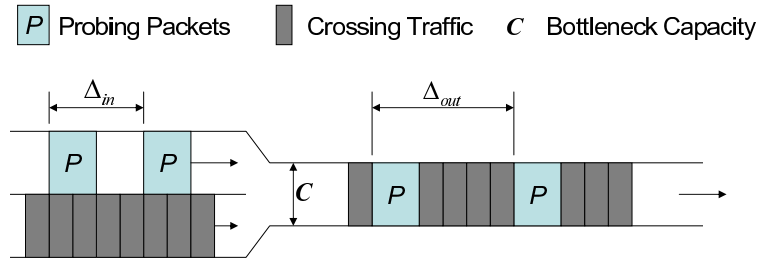


Figure 3.3: Probe Gap Model

PGM assumes a single bottleneck which is both the narrow and tight link for that path and that the queue is not empty between the two packets in a probing packet pair. Thus, as shown in Figure 3.3 [121], a probing pair is sent with a initial time gap time  $\Delta_{in}$ , and reaches the receiver with a receiving time gap  $\Delta_{out}$ .  $\Delta_{out}$  is the time taken by the bottleneck to transmit the second probing packet in the pair and the crossing traffic that arrived during  $\Delta_{in}$ . Therefore, the time spend on transmitting the crossing traffic at the bottleneck is  $\Delta_{out} - \Delta_{in}$ . If the bottleneck capacity  $C$  is known, the rate of the crossing traffic  $R_c$  can be presented as [121]:

$$R_c = \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}} \times C \quad (3.14)$$

and the available bandwidth  $A$  can be computed as:

$$A = C \times \left(1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}}\right) \quad (3.15)$$

Compared with the available bandwidth estimation using a self-loading probe, PGM usually has a fast measurement time and lower stress on the end-to-end network. PGM assumes that the narrow link capacity is known and constant. However, this may not be true in special cases such as in wireless networks. Therefore, the unknown-capacity problem limits the usage of PGM in a uncooperative environment.

### BTC and Achievable Bandwidth Estimation Techniques

Several other bandwidth estimation tools are proposed to measure Bulk Transport Capacity (BTC) and achievable throughput. For instance, *Treno* [124] uses UDP or ICMP packets to emulate TCP traffic to get end-to-end delay information. Thus, the BTC of the end-to-end path can be estimated by applying flow control and congestion control algorithms that are similar to TCP. *Cap* [125] is another BTC measurement tool that uses UDP to emulate both TCP and ACK packets to measure the BTC of an end-to-end network path.

Other benchmarking tools, such as *TTCP*<sup>1</sup>, *Netperf*<sup>2</sup> and *Iperf*<sup>3</sup> are also used to estimate TCP achievable throughput. These tools usually use large TCP transfers and/or parallel TCP connections, and therefore can cause stress on the network under test.

### Passive Wireless Characterization Estimation

As discussed in Section 2.2, the MAC layer retransmission and rate adaptation make it much more difficult to estimate bandwidth over wireless networks. However, recent research [126] proposes a passive measurement methodology to estimate the link characteristics based on wireless link quality and contention status of the IEEE 802.11 network. Moreover, [127] presents a similar approach to estimate per-neighbor available bandwidth. It

<sup>1</sup><http://www.pcausa.com/Utilities/pcattcp.htm>

<sup>2</sup><http://www.netperf.org/netperf/NetperfPage.html>

<sup>3</sup><http://dast.nlanr.net/Projects/Iperf/>

measures running average transmission and decay of successfully transmitted MAC frames, then calculates the throughput normalized to a pre-defined packet size. By normalizing the estimated available bandwidth, the author solves the problem of wireless throughput of a packet depending on the size of the packet. In addition, two case studies, the ad hoc network admission control and a rate based flow control, are presented as the applications of the bandwidth estimation techniques.

However, without sending probing packets to the network, the estimation is limited to the local wireless network only. In addition, the estimation is usually implemented in the MAC layer, which does not include higher layer information, such as congestion in the IP layer. Therefore this approach can not be used to guide the application unless combined with other bandwidth estimations techniques.

### **Taxonomy of Bandwidth Estimation Tools**

Based on the taxonomy in the survey [92] and the reviews in this section, Table 3.1 summarizes most of the currently published bandwidth measurement tools, along with the target bandwidth metrics and the basic technology. Additionally, a mostly up-to-date taxonomy is also available at CAIDA's webpage<sup>4</sup>.

### **3.2.3 Evaluation of Bandwidth Estimation Techniques**

Evaluation of bandwidth estimation techniques is not easy because of the mixture of measurement metrics and mixture of techniques based on a variety of assumptions. Therefore, only those techniques with similar measurement target metrics can be compared and evaluated side by side.

Some general evaluation criteria may include:

- Accuracy. How close does the measurement result compare to the real network condition?
- Convergence time. How long does it take to get meaningful results?

---

<sup>4</sup><http://www.caida.org/tools/taxonomy/>

Table 3.1: Taxonomy of Published Bandwidth Estimation Tools

Tool	Measurement metrics	Methodology
pathchar [97]	Per-hop Capacity	Variable Packet Size
clink [98]	Per-hop Capacity	Variable Packet Size
pchar [99]	Per-hop Capacity	Variable Packet Size
bprobe [106]	End-to-end Capacity	Packet Dispersion
nettimer [108]	End-to-end Capacity	Packet Dispersion
pathrate [110]	End-to-end Capacity	Packet Dispersion
sprobe [101]	End-to-end Capacity	Packet Dispersion
cprobe [106]	End-to-end Available Bandwidth	Packet Dispersion
TOPP [115]	End-to-end Available Bandwidth	Self-loading Probe
PTR [117]	End-to-end Available Bandwidth	Self-loading Probe
pathload [113]	End-to-end Available Bandwidth	Self-loading Probe
pathChirp [116]	End-to-end Available Bandwidth	Self-loading Probe
delphi [122]	End-to-end Available Bandwidth	Probe Gap Model
IGI [117]	End-to-end Available Bandwidth	Probe Gap Model
Spruce [121]	End-to-end Available Bandwidth	Probe Gap Model
BFind [118]	Bottleneck Locator <sup>a</sup>	Self-loading Probe
Pathneck [120]	Bottleneck Locator	Self-loading Probe
Cartouche [119]	Bottleneck Locator	Self-loading Probe
Treno [124]	Bulk Transfer Capacity	Emulated TCP Throughput
cap [125]	Bulk Transfer Capacity	Emulated TCP Throughput
Ttcp	Achievable TCP Throughput	TCP Connection
Iperf	Achievable TCP Throughput	Parallel TCP Connections
Netperf	Achievable TCP Throughput	Parallel TCP Connections

<sup>a</sup>Bottleneck Locator combines the Available Bandwidth estimation with ICMP like hop by hop measurement to allocate the bottleneck in the end-to-end network path.

- Intrusiveness. How much probing traffic is sent into the network and will this traffic stress the network? The intrusiveness should consider not only the amount but also the burstiness of the probing traffic.
- Robustness. Is the tool robust enough to achieve an accuracy result in complex network environments, such as multiple bottlenecks or wireless networks?
- Usability. Does the tool need to be installed on both ends of the network path or need support by intermediate routes?

There are a number of evaluation and comparison studies for most of the popular bandwidth estimation tools. For instance, in [121], Spruce, IGI and Pathload are evaluated from the aspects of accuracy, failure patterns, probe overhead, and implementation issues by measurement in 400 different Internet-wide paths. The measurement results show that first, Spruce is more accurate than Pathload and IGI; almost 70% of Spruce’s measurements had a relative error smaller than 30%. Second, Pathload tends to overestimate the available bandwidth whereas IGI becomes insensitive when the bottleneck utilization is large. Finally, Pathload generated between 2.5 and 10 MBytes of probing traffic per measurement while the average per-measurement probing traffic generated by IGI is 130 KB and that generated by Spruce is 300 KB.

Hu *et al.* [117] compare IGI, PTR with Pathload for both accuracy and convergence time using 13 network paths with different capacities and RTTs. The results show that those three methodologies provide fairly similar accuracy. However, the IGI/PTR method is on average more than 20 times faster than Pathload for their setup.

Easwaran *et al.* [128] compares Pathload, IGI and pathChirp in terms of their accuracy, intrusiveness and overhead in a network simulation environment. A  $2^k$  factorial design is used to analyze the importance of the packet size, number of trains, number of packets per train and frequency of runs in these performance metrics. The results show that all three tools perform very well in terms of accuracy when only UDP cross traffic is present. However, pathChirp performs poorly in the scenario where only one TCP flow is present. In addition, the authors also found that IGI has the best (smallest) convergence time followed by pathChirp and Pathload.

A general summary of the currently available bandwidth estimation techniques is listed in Table 3.2. The accuracy, convergence time and intrusiveness are compared across the major active bandwidth estimation techniques and the passive wireless characterization technique. The comparison is not based on precise measurement; instead, it is only based on the general review of the techniques in each category. For instance, the VPS technique of sending multiple variable size packets into the network, is considered as a “High” in-

intrusiveness approach. VPS takes a relatively long time to decide the per-hop capacity, which results in a “Slow” convergence time. In addition, the result is expected to be impacted by the Layer 2 equipment, as well as by cross traffic, which results in a “Medium” accuracy. Similarly, for Probe Gap Model and Packet Dispersion, the estimation can be done by only a few packets and in a short time period, thus both have “Low” intrusiveness and “Fast” convergence time. However, the Packet Dispersion technique is more sensitive to the crossing traffic, therefore it has a “Medium” accuracy, while Probe Gap Model has a “High” accuracy. For Self-loading Probe techniques, depending on the convergence methodologies, which could be linear, exponential or binary search, it could have variable intrusiveness. However, the searching nature of self-loading techniques results in “Slow” convergence time in general. Finally, the “Remark” column lists the additional applicable characterizations of each technique. Thus, even without precise measurement, the table can still be used by applications as a basic guideline for selecting an appropriate bandwidth estimation technique.

Table 3.2: Summary of Bandwidth Estimation Tools

Methodology	Measurement Metrics	Accuracy	Converge	Intrusiveness	Remark
Variable Packet Size [97, 98, 99]	Per-hop Capacity	Medium	Slow	High	Uses ICMP
Packet Dispersion [101, 106, 108, 110]	End-to-end Capacity	Medium	Fast	Low	
Self-loading Probe [113, 115, 116, 117]	End-to-end Available Bandwidth	High	Slow	Varies	
Probe Gap Model [117, 121, 122]	End-to-end Available Bandwidth	High	Fast	Low	Need to know capacity
Emulated TCP [124, 125]	Bulk Transfer Capacity	Medium	Fast	Low	Only for TCP
TCP Connection(s)	Achievable TCP Throughput	High	Slow	High	Only for TCP
Passive Wireless [126, 127]	Wireless Available Bandwidth	Medium	Fast	None	Passive Wireless only

However, as discussed in [129], evaluation of the effectiveness of these techniques should consider the actual accuracy and latency constraints of real applications. Different classes

of applications may focus on different criteria. For instance, our research is focused on the bandwidth estimation techniques used for streaming media applications over wireless networks, which may not require high accuracy, but instead, need a fast convergence time. The evaluation criteria used in our research is described in Chapter 5 in detail.

### 3.3 Wireless Network Performance Study

Wireless network performance research can be done in number of ways, such as using analytical models, simulations, emulations and measurements. This thesis focuses on the performance improvement of streaming applications over wireless networks by using bandwidth estimation techniques. The performance study methodologies used in this dissertation include the modeling of bandwidth estimation over wireless networks and performance evaluation by simulation and measurement in wireless networks. This section reviews the related work of wireless network performance studies that are referred to or used in this thesis.

#### 3.3.1 Analytical Modeling

Modeling of wireless network performance can provide a low cost, fast way to analyze wireless conditions with varied configurations. However, as discussed in Section 2.2, the wireless network performance is affected by many factors, such as the signal attenuation, fading, interference, bit errors and contention. Accurate modeling of wireless network performance in a complex configuration is still a challenge. Most of the modeling research has focused on throughput and delay based on different assumptions.

The research in [130] uses Markov chain models to analyze DCF operation and calculates the saturated throughput of the 802.11 protocol. The model assumes an idealistic channel condition of collision-only errors and unlimited packet retransmissions, such that a lost packet is retransmitted until its successful reception. In addition, the model assumes a fixed number of stations in the network, and the network operates in saturation conditions, i.e. the transmission queue in each station is assumed to be always nonempty.

Based on the derivation from the Markov chain model, the probability  $\tau$  that a station transmits in a randomly chosen time slot can be presented as:

$$\tau = \frac{2(1 - 2p)}{(1 - 2p)(W + 1) + pW(1 - (2p)^m)} \quad (3.16)$$

where  $W$  is the initial contention window size,  $m$  is the maximum number of backoff stages, and  $p$  is conditional collision probability:

$$p = 1 - (1 - \tau)^{n-1} \quad (3.17)$$

where  $n$  is the number of stations in the network.

The author proves that there is a unique solution for  $\tau$  and  $p$  from the nonlinear system presented by Equation 3.16 and 3.17. Therefore,  $\tau$  and  $p$  can be obtained by numerical techniques.

The throughput  $S$  is modeled by

$$\begin{aligned} S &= \frac{E[\text{payload transmitted in a slot time}]}{E[\text{length of a slot time}]} \\ &= \frac{P_s P_{tr} E[P]}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr} (1 - P_s) T_c} \end{aligned} \quad (3.18)$$

where  $P_{tr}$  is the probability that there is at least one transmission in the time slot:

$$P_{tr} = 1 - (1 - \tau)^n \quad (3.19)$$

$P_s$  is the probability that a transmission occurring on the channel is successful:

$$P_s = \frac{n\tau(1 - \tau)^{n-1}}{P_{tr}} = \frac{n\tau(1 - \tau)^{n-1}}{1 - (1 - \tau)^n} \quad (3.20)$$

The average length of a slot time is given by:

$$E[\text{length of a slot time}] = (1 - P_{tr})\sigma + P_{tr}P_sT_s + P_{tr}(1 - P_s)T_c \quad (3.21)$$

where  $T_s$  is the average time the channel is sensed busy because of a successful transmission and  $T_c$  is the average time the channel is sensed busy by each station during a collision. Equations 3.22, 3.23 and Equations 3.24, 3.25 give the value for  $T_s^{bas}$ ,  $T_c^{bas}$  and  $T_s^{rts}$ ,  $T_c^{rts}$ , which are  $T_s$  and  $T_c$  of the basic access case and RTS/CTS access mechanism, respectively:

$$T_s^{bas} = H + E\{P\} + sifs + \delta + ack + difs + \delta \quad (3.22)$$

$$T_c^{bas} = H + E\{P\} + difs + \delta \quad (3.23)$$

$$\begin{aligned} T_s^{rts} &= rts + sifs + \delta + cts + sifs + \delta + H + E\{P\} \\ &\quad + sifs + \delta + ack + difs + \delta \end{aligned} \quad (3.24)$$

$$T_c^{rts} = rts + difs + \delta \quad (3.25)$$

where  $rts$ ,  $cts$ ,  $ack$ ,  $H$  and  $E\{P\}$  are the transmission times of RTS, CTS, ACK, packet header (physical layer plus MAC layer) and data packets, respectively, and  $E\{P\} = P$  for a fixed packet size.  $\delta$  is the propagation delay.  $sifs$  (Short Interframe Space),  $difs$  (Distributed Interframe Space) and other specific values for DSSS and FHSS are listed in Table 3.3.

Recent research, such as in [54, 131, 132, 133, 134], extend the analytical model in [130] in a number of ways. For instance, Wu [131] extends the analysis to include the finite packet retry limits as defined in the IEEE 802.11 standard. Research in [54] also shows that the average delay (the service time) of a single hop ad hoc network at saturation can be modeled based on the Markov chain model used in [130]. Chatzimisios [132] calculates the packet delay without considering any packet dropping due to retry limits. In their

Table 3.3: IEEE 802.11 Physical Layer Parameters

	DSSS	FHSS
$W_{min}$	32	16
$W_{max}$	1024	1024
MAC header	34 bytes	34 bytes
Phy header	24 bytes	16 bytes
ACK	38 bytes	30 bytes
CTS	38 bytes	30 bytes
RTS	44 bytes	36 bytes
Slot time	20 $\mu$ sec	50 $\mu$ sec
SIFS	10 $\mu$ sec	28 $\mu$ sec
DIFS	50 $\mu$ sec	128 $\mu$ sec

follow-on research [133], which uses a performance model of 802.11 DCF by means of the Markov chain model similar to the one from [131], the authors consider the effect of retry limits and calculates the packet delay, the packet drop probability and the packet drop time. Their successive research in [134] further extends the existing model to include the effect of transmission errors.

As modeled in [133], the average packet delay  $E[D]$  of a packet that is not discarded, is given by:

$$E[D] = E[X] \times E[\text{length of a slot time}] \quad (3.26)$$

where  $E[X]$  is the average number of slot times required to successfully transmit a packet and is given by:

$$E[X] = \sum_{i=0}^m \left[ \frac{(p^i - p^{m+1}) \frac{W_i + 1}{2}}{1 - p^{m+1}} \right] \quad (3.27)$$

where  $(1 - p^{m+1})$  is the probability that the packet is not dropped and  $(p^i - p^{m+1}) / (1 - p^{m+1})$  is the probability that a packet that is not dropped at stage  $i$ .

In addition, the research from Calì [135, 136, 137] models the theoretical IEEE 802.11 network capacity and by designing a new dynamic backoff algorithm, the authors im-

proved the network throughput close to the theoretical throughput limit. Similarly, the research [138, 139, 140] focuses on the modeling of the theoretical maximum throughput of IEEE 802.11 networks with different physical layer modulation techniques and therefore, different data rates.

### 3.3.2 Network Simulations

Network Simulation had been widely used in wireless network performance studies and for validations of performance modeling. For instance, Bianchi [130] uses a customized simulation program developed in C++. Chatzimisios [134] uses commercially available simulation suites OPNET<sup>5</sup> and Carvalho [54] used NS2<sup>6</sup> to validate their analytical models. The other popular wireless network simulators include Parsec/GloMoSim<sup>7</sup>, and QualNet<sup>8</sup>, which is the commercial version of the Parsec simulator.

Among this set of simulators, NS2 is the most widely used open source network simulator. In addition to the basic IEEE 802.11 MAC and physical layer implementation in NS2 [141], there are number of extended modules publicly available. For instance, the Dynamic Rate Adaptation with Ricean fading modules, and GPRS module are reviewed in this section.

#### Rate Adaptation Simulation

The Auto Rate Fallback (ARF) protocol [56] was the first commercial implementation of a MAC that utilizes the rate adaptation feature. With ARF, senders attempt to use higher transmission rates after consecutive transmission successes (which indicate high channel quality) and revert to lower rates after failures. Under most channel conditions, ARF provides a performance gain over pure single rate IEEE 802.11.

In [20], a protocol termed Receiver Based Auto Rate (RBAR) is proposed. The core idea of RBAR is for receivers to measure the channel quality using physical layer analysis

---

<sup>5</sup><http://www.opnet.com/>

<sup>6</sup><http://www.isi.edu/nsnam/ns/>

<sup>7</sup><http://pcl.cs.ucla.edu/projects/glomosim/>

<sup>8</sup><http://www.scalable-networks.com/products/>

of the RTS message. Receivers then set the transmission rate for each packet according to the highest feasible value allowed by the channel conditions and send the rate information via the CTS packet back to the sender. Moreover, as the RTS/CTS messages are sent at the base rate so that all nodes can overhear them, overhearing nodes are informed of the modified data transmission times so that they can set their backoff timers accordingly. However, the RBAR is only available in RTS/CTS access mode but not for the basic access mode because the mandatory request of RTS/CTS exchanges. Similar research from [142] also uses signal strength measurement. However, it only uses the sender's received signal strength, and therefore does not request the RTS/CTS access mode.

The Opportunistic Auto Rate (OAR) protocol [143] is presented to better exploit durations of high-quality channels conditions. The key mechanism of the OAR protocol is to opportunistically send multiple back-to-back data packets whenever the channel quality is good.

However, these solutions of multi-rate adaptation modules are not always publicly available in NS2. One of the available versions of multi-rate simulation of OAR and RBAR is provided by [143] in NS2 2.1b7, which can be downloaded from the Rice Networks Group webpage.<sup>9</sup> Our recent research [144] re-implements the algorithm in NS 2.27 and extends the physical layer parameters with the specification of the Lucent OriNOCO wireless PC card<sup>10</sup>. Section 4.2 discusses the RBAR module for NS 2.27 in detail and uses it to demonstrate the issues of using packet dispersion techniques in wireless networks.

### 3.3.3 Performance Measurements

Modeling and simulation of the wireless network performance usually provides analysis in controlled environments. However, the wireless network's uncertainties make it difficult to achieve accurate models. Thus, wireless measurement and physical emulations are important methods in conjunction with performance models.

Wireless measurements can be performed either on the Access Point (AP), mobile

---

<sup>9</sup><http://www-ece.rice.edu/networks/>

<sup>10</sup><http://www.agere.com/client/wlan.html>

host, or by a special designed network monitoring/sniffing system. For instance, the research in [145] characterizes user behavior and wireless network performance in a public IEEE 802.11 network at a conference by collecting Simple Network Management Protocol (SNMP) traces from the APs. Similarly, research in [146, 147, 148] analyzes either metropolitan area or campus wide wireless networks by collecting AP system log and SNMP information. In addition, Ho *et al.* [149] present VISUM, a scalable framework for wireless network monitoring based on similar methodology. VISUM relies on a distributed set of agents within the network to monitor network devices and therefore supports a much larger scale of networks.

Wireless measurement can be applied to the mobile host. Wireless Research API (WRAPI) [150] is a software library that allows applications running in user-space on mobile hosts (and APs) to query/set information in the IEEE 802.11 network. WRAPI provides an interface for applications to monitor the WLAN in real time by interacting with Network Driver Interface Specification (NDIS) stack of Windows XP. Since WRAPI does not make direct contact with the hardware driver, it is hardware independent and supports all 802.11b and g compliant hardware in Windows XP systems. However, WRAPI can not provide detailed information, such as packet level statistical information and does not work in promiscuous mode, which limits its capability as a network monitoring tool. Recent research [8] uses WRAPI to capture the WLAN performance information, including wireless layer Received Signal Strength Indicator (RSSI), MAC layer retry count, multiple retry count, ACK failure count, and duplicated frame count.

To get MAC level frame information of a wireless network, a wireless sniffing system is usually used. A wireless sniffer can be installed on a measured host, but in most cases, it is installed on an independent device, such as a mobile computer or a PDA system. Therefore the sniffer can monitor the wireless network in promiscuous mode without interfering with the stations under measurement. Wireless sniffers can capture not only the data frames, but also management frames, such as beacon frames, and RTS/CTS/ACK frames. However, the wireless sniffer requires special hardware and driver support. The

most popular wireless sniffer and analyzer software includes Ethereal<sup>11</sup>, Kismet<sup>12</sup> and some commercially available wireless sniffers such as Sniffer Wireless (Used to be Network Associates Sniffer)<sup>13</sup>, AiroPeek NX<sup>14</sup>, etc. Wireless sniffers have been widely used in wireless performance research, such as the independent sniffer used in the measurement of streaming media over wireless research [6, 9], and the on host software sniffer used in the link level measurement research for a wireless roof network [151]. Moreover, in the network monitor research in [152], a complete wireless sniffer system is implemented and used to characterize a typical computer science department WLAN traffic.

However, wireless measurement is usually performed under an uncontrolled environment with random errors and fading effects, the results are usually difficult to accurately reproduce. Therefore, a wireless channel emulator is usually used to create controlled and repeatable channel conditions. The commercial channel emulators, such as PROPSim<sup>15</sup> and Spirent<sup>16</sup>, are designed to support fine-grained emulation of the wireless channel between either a pair of devices or between a small number of base stations. Judd *et al.* [153] present a similar physical channel emulation by using the Digital Signal Processing (DSP) engine to model the effects of signal propagation (e.g. large-scale attenuation and small-scale fading) on each signal path between wireless interfaces. The approach is used by [151] to emulate the multipath fading in a 802.11 wireless network.

In this thesis, we create an IEEE 802.11b/g wireless testbed<sup>17</sup> to validate our models and evaluate the algorithms. By controlling the wireless AP and client's configuration and applying variable traffic loads, we perform measurements under multiple wireless network conditions. Additionally, we use an independent wireless sniffer to validate the configurations in our wireless testbed. Chapter 5 and 6 describe the testbed in detail.

---

<sup>11</sup>Renamed to Wireshark, online at <http://www.wireshark.org/>

<sup>12</sup><http://www.kismetwireless.net/index.shtml>

<sup>13</sup><http://www.sniffer.com>

<sup>14</sup>[http://www.wildpackets.com/elements/AiroPeek\\_NX.pdf](http://www.wildpackets.com/elements/AiroPeek_NX.pdf)

<sup>15</sup><http://www.propsim.net/>

<sup>16</sup><http://www.spirentcommunications.com/>

<sup>17</sup><http://perform.wpi.edu/wsml/>



## Chapter 4

# Packet Dispersion in IEEE 802.11 Wireless Networks

This chapter presents the in-depth study of packet dispersion techniques in IEEE 802.11 wireless networks. Combining an analytical model of packet dispersion, simulations and measurement studies in wireless networks, two packet dispersion measurements, *effective capacity* and *achievable throughput* are introduced. Section 4.1 gives a brief overview of packet dispersion in wireless networks. Section 4.2 introduces rate adaptation and fading extensions to NS-2 simulations and discusses the issues of bandwidth estimation in wireless networks by using the simulations. Section 4.3 provides a packet dispersion model for IEEE 802.11 networks and the model validation using simulations and measurements. Section 4.4 uses the model to analyze packet dispersion and defines the terminologies of effective capacity and achievable throughput in wireless networks. Finally, Sections 4.5 summarizes the chapter. The model and measurements studied in this chapter are used by our bandwidth estimation tool in Chapter 5.

## 4.1 Overview

The differences in wired and wireless packet dispersion are the major source of wireless bandwidth estimation errors. Thus, reducing measurement errors and improving performance in wireless local area networks (WLANs) requires a better understanding of packet dispersion in wireless networks.

While many research models have been developed for wireless networks, few consider WLAN bandwidth estimation. Moreover, current research tends to focus on simplified conditions such as fixed wireless capacities or error free wireless networks [154] to create tractable models. While previous research [90] has demonstrated the impact of IEEE 802.11 packet size and rate adaptation on bandwidth estimation tools, it is difficult to improve the bandwidth estimation tools without an in-depth model of wireless packet dispersion. Therefore, this investigation puts forth both an analytic and a simulation model for WLANs that includes packet dispersion under conditions such as channel contention, fading, BER and dynamic rate adaptation. The analytical model captures WLAN packet dispersion behavior to study the impact of such channel conditions and wireless configuration parameters such as packet sizes, link rate and RTS/CTS on the mean and variance of bandwidth estimation results. Using the packet dispersion model, two wireless packet dispersion measures, *effective capacity* and *achievable throughput* are introduced. This chapter also shows that in a saturated WLAN a fluid flow model is not applicable because of the probability-based fairness for channel access across WLAN nodes. The packet dispersion model is validated using network measurements in a wireless 802.11b testbed and an NS-2 simulator modified to include dynamic rate adaptation in the face of challenging environmental conditions. Armed with analytic models, simulation tools and network measurements, this chapter provides a preliminary study of bandwidth estimation techniques based on a WLAN using packet dispersion and provides insight into possible improvements to WLAN bandwidth estimation techniques.

## 4.2 Packet Dispersion Issues in Wireless Networks

### 4.2.1 Rate Adaptation Simulation

We use NS-2 simulations to illustrate the issues of packet dispersion techniques in wireless networks. However, while NS-2<sup>1</sup> provides IEEE 802.11 components such as CSMA/CA, MAC layer retries, contention, propagation and error models, it lacks a rate control algorithm (RCA). Since the 802.11 standard [11] does not specify a specific RCA, each WLAN card manufacturer is free to implement their own RCA. RCAs adjust link rates based on the signal strength or by reacting to accumulated statistics, such as number of retries, packet error rate or throughput [55, 155]. Auto Rate Fallback (ARF) [56], the first commercial RCA implementation, raises the data rate after consecutive transmission successes and lowers the data rate after link layer transmission failures. Under most wired channel conditions, ARF outperforms fixed-rate 802.11, but when transmission failures are caused by wireless link layer congestion, ARF can have a negative impact [156].

Receiver Based Auto Rate (RBAR) [20] uses RTS frame analysis to measure channel quality. An RBAR receiver determines the highest feasible frame transmission rate that channel conditions can tolerate and notifies the sender of the chosen rate via a CTS frame. Since RTS/CTS messages are sent to the AP, all wireless nodes become aware of the new transmission rate and set their backoff timers accordingly. However, RBAR is not available in basic mode where RTS/CTS is disabled.

Starting with an RBAR simulation module provided by [143] for NS-2 2.1b7,<sup>2</sup> RBAR was re-implemented in NS 2.27. We extended the physical layer parameters using the specifications of the Lucent OriNOCO wireless PC card.<sup>3</sup> Our documented RBAR implementation is available online.<sup>4</sup> Figure 4.1 provides NS-2 throughput results versus separation distance for two simulated wireless nodes moving away from each other. Average throughput is measured using 1000-byte packets for a single CBR flow with RTS/CTS enabled.

---

<sup>1</sup>The Network Simulator - NS-2. Online at <http://www.isi.edu/nsnam/ns/>

<sup>2</sup>Downloadable from <http://www-ece.rice.edu/networks/>.

<sup>3</sup><http://www.agere.com/client/wlan.html>

<sup>4</sup><http://perform.wpi.edu/downloads/#rbar>

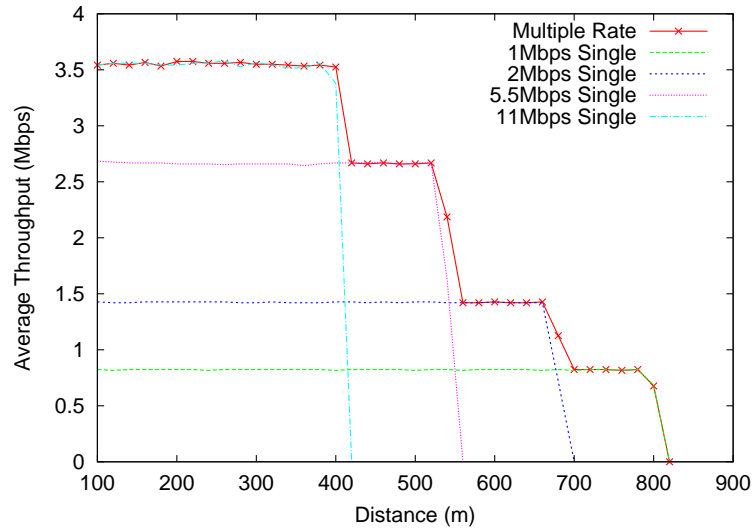


Figure 4.1: Throughput Comparison As Distance From Sender to Receiver Increases

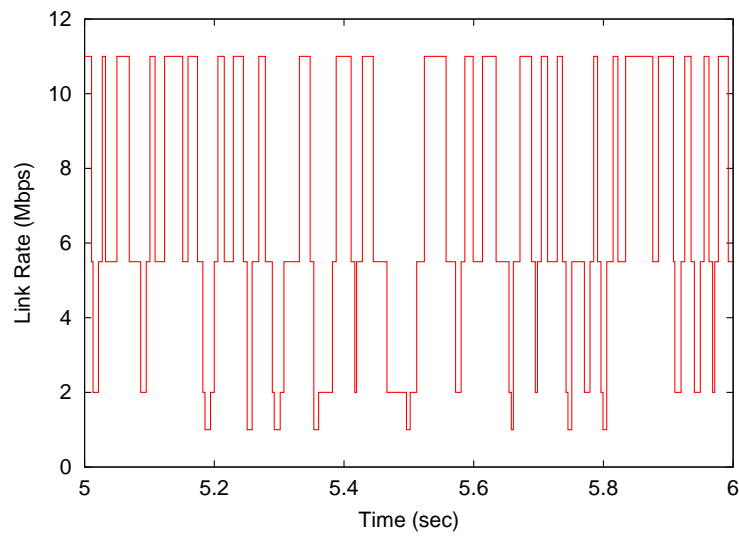


Figure 4.2: Link Rate Adaptation Under Ricean Fading

The fixed-rate approaches (1, 2, 5.5 and 11 Mbps) have a relatively fixed throughput as the distance increases until the link is dropped when the nodes move out of transmission range. RBAR (labeled “Multiple Rate”) dynamically adjusts the rate downward as distance increase.

To more accurately simulate physical condition effects on RCAs, an additional NS-2 extension modeled Ricean (or Rayleigh) fading [157] was implemented and imported into NS 2.27. Figure 4.2 shows simulated effects of Ricean fading for two wireless nodes 390 meters apart where, with fading turned off, RBAR would fix the data rate at 11 Mbps. The figure tracks RBAR dynamically adjusting the rate between 11, 5.5, 2 and 1 Mbps in response to fading strength variability as a function of time.

#### 4.2.2 The Impact of Wireless Networks

This section discusses wireless physical layer and MAC layer issues that may cause bandwidth estimation techniques to perform poorly.

Most wireless MAC layers use frame retries or Forward Error Correction (FEC) to recover lost frames. IEEE 802.11 networks retransmit up to a fixed number of times with exponential backoff between retransmissions. While frame retries reduce packet loss, frame retries increase the variance in packet delay that yields packet dispersion inconsistencies and large variations in time measurements. Namely, dispersion between packet pairs can be compressed or expanded when traversing a wireless AP even without congestion in the network or without changes in the link capacity.

Figure 4.3 depicts a typical network topology for studying packet dispersion in a WLAN. To characterize the effects of wireless traffic on packet dispersion, the wireless network traffic is divided into probing, crossing and contending traffic. *Probing traffic* is the packet pairs or trains sent along the estimated network path through the AP to the client (1). Wireless channel conditions and other traffic may vary the probing traffic dispersion behavior and produce estimation errors.

While *crossing traffic* does not contend with probe packets, crossing traffic does share

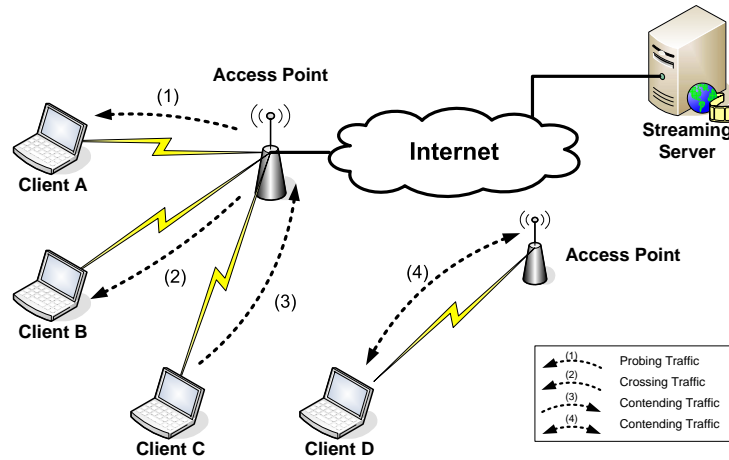


Figure 4.3: Probing, Crossing and Contending Traffic in a WLAN

the bottleneck and thereby strongly impacts the accuracy of bandwidth estimates on the WLAN. Figure 4.3 shows crossing traffic coming from the AP to associated clients (2). After subtracting contending effects from other wireless traffic, wireless crossing traffic shares the bandwidth with the probing traffic. However, even though statistically contending effects caused by crossing traffic indirectly impact bandwidth estimates, this impact can be captured by packet dispersion techniques. Since several statistical filtering methodologies have been proposed to mitigate the effects of cross traffic [106, 110], crossing traffic effects in WLANs are not considered further in this chapter.

*Contending traffic* accesses the shared wireless channel and competes with probe packets on the estimated path. Figure 4.3 shows contending traffic sent by clients to the same AP (3) and between other clients and APs (4) within interference range (referred to as co-channel interference). To avoid channel capture, which is a channel being monopolized by a single node, or subset of nodes in a given geographic region, 802.11 uses random backoff between two successive frames sent from the same node. When packet pairs arrive back-to-back at the AP, the AP delays the second packet by inserting a random backoff time between the packets. Thus, bandwidth estimates using packet dispersion on 802.11 networks are vulnerable to contending traffic that transmits during the delay between the two packets and further delays the second packet in the pair.

Dynamic rate adaptation impedes bandwidth estimation methods because these techniques assume a fixed capacity during the measurement. Figure 4.2 shows WLAN capacity varying frequently under bad channel conditions. Hence, wireless bandwidth estimation changes with the same granularity. Figure 4.4 uses NS-2 wireless simulations with RTS/CTS enabled to illustrate the impact of network conditions on packet pair estimation techniques. Each simulation sends continuous packet pairs downstream over a single hop wireless 802.11b network. Both the packet pair and the contending traffic send 1000-byte packets. In all cases, contention is simulated as a 1 Mbps upstream CBR flow. For the ideal channel, simulation errors and fading are disabled. In the fading channel, Ricean propagation from Section 4.2.1 is used. For the BER channel, a uniform bit error rate of  $5.0 \times 10^{-4}$  is used. Each CDF curve represents estimates from 1000 packet pairs sent over the wireless network.

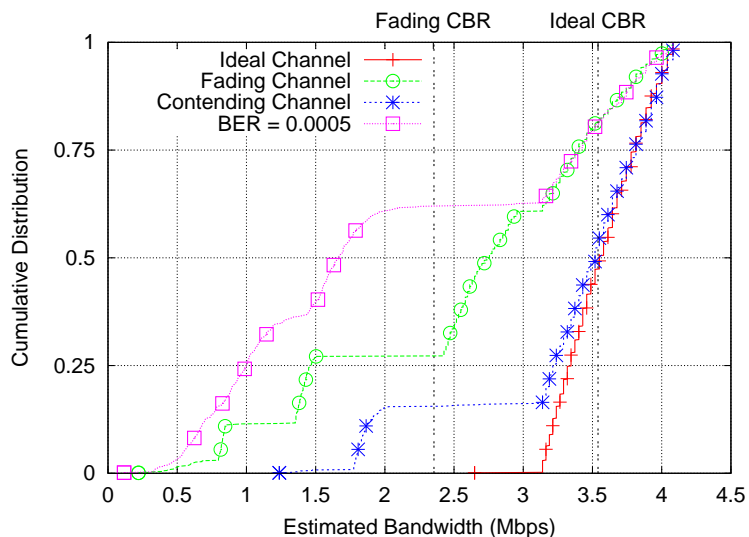


Figure 4.4: Capacity Estimation Using Packet Pair Techniques in a WLAN

In Figure 4.4, the estimated bandwidth of the ideal channel is uniformly distributed over the range of 3.1 Mbps to 4.1 Mbps due to the random backoff between two successive packets. The multiple mode distribution in the fading channel case is due to dynamic rate adaptation. The strong offset on the capacity estimation at about 1.8 Mbps for

the contending channel is due to delay induced by contending packets. The estimated bandwidth results with bit errors yield a continuous cumulative distribution function (CDF) under the 1.8 Mbps range due to frame retries and exponential backoff delay between consecutive retransmissions. However, the step trend between 1.8 Mbps and 3.1 Mbps is similar to the distribution of the contending channel. The ‘Ideal CBR’ and ‘Fading CBR’ vertical lines represent average CBR throughputs which approximate average capacity in the ideal and fading channel cases, respectively. Compared to the CBR throughputs, the packet pair estimates are spread over a wide range. This clearly shows the packet dispersion technique is significantly impacted by wireless channel conditions.

### 4.3 Wireless Network Packet Dispersion Model

This section develops an analytical model based on existing IEEE 802.11 wireless network models to explore the relationship between packet dispersion and WLAN conditions.

Capturing packet transmission delay is key to modeling bandwidth estimation techniques that use packet pair (or train) dispersion. The bottleneck (both the narrow and the tight link) on the end-to-end network path is assumed to be the last hop WLAN. While not necessarily true for all flows, this assumption decouples wireless behavior from other issues and simplifies the wireless analysis. To further simplify modeling WLAN packet pair dispersion, no crossing traffic is assumed.

#### 4.3.1 Packet Dispersion Model

The model characterizes the dispersion  $T$  between two packets in a packet pair in terms of the average,  $E[T]$ , and the variance,  $V[T]$ , of packet dispersion for a given wireless network that includes packet size, link rate, BER and access methods. Our packet dispersion model is built from two Markov chain models: 1) Bianchi [130] uses a Markov model that assumes an idealistic, collision-free channel with a number of stations to analyze DCF operation. To simplify the model, frame retransmissions are considered unlimited such that frames are retransmitted until successful transmission and the 802.11 channel is saturated with

each station always having a frame to send. 2) Chatzimisios et al [134] extend this model to include transmission error effects. For a given BER, their model derives the probability  $\tau$  that a station transmits in a randomly chosen time slot as:

$$\tau = \frac{2(1-2p)(1-p^{m+1})}{W_{min}(1-(2p)^{m+1})(1-p) + (1-2p)(1-p^{m+1})} \quad (4.1)$$

where  $W_{min}$  is the initial contention window size,  $m$  is the maximum number of backoff stages, and  $p$  is conditional packet error probability:

$$p = 1 - (1-\tau)^{n-1}(1-BER)^{L+H} \quad (4.2)$$

$n$  is the number of contending stations in the network,  $L$  and  $H$  are the packet and packet header sizes (physical layer plus MAC layer) in bits. Since the authors prove there exists a unique solution for  $\tau$  and  $p$  from the nonlinear system presented by Equation 4.1 and 4.2, these two probabilities can be obtained by numerical techniques.

Characterizing any given MAC layer time slot as either idle, collision, error, or successful, the average length of the slot time is given by:

$$E[slot] = (1-P_{tr})\sigma + P_{tr}P_sT_s + P_{tr}P_cT_c + P_{tr}P_{er}T_{er} \quad (4.3)$$

where  $P_{tr}$  is the probability that there is at least one transmission in the time slot:

$$P_{tr} = 1 - (1-\tau)^n \quad (4.4)$$

$P_s$  is the probability that a transmission occurring on the channel is successful:

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}(1-PER) \quad (4.5)$$

and PER is the packet error rate, computed from the BER as  $PER = 1 - (1-BER)^{L+H}$ .

The probability  $P_c$  of a collision when two or more stations simultaneously transmit is:

$$P_c = 1 - \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n} \quad (4.6)$$

and the probability  $P_{er}$  that a packet is received in error is:

$$P_{er} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n} PER \quad (4.7)$$

In Equation 4.3,  $\sigma$  is the idle slot time,  $T_s$  is the average time the channel is busy because of a successful transmission,  $T_c$  and  $T_{er}$  are the average time the channel is sensed busy by each station during a collision or packet error, respectively.

Equations 4.8-4.11, defined in [130], define  $T_s^{bas}$ ,  $T_c^{bas}$ ,  $T_s^{rts}$ , and  $T_c^{rts}$ , which are  $T_s$  and  $T_c$  for the basic access and RTS/CTS access, respectively:

$$T_s^{bas} = H + E\{L\} + sifs + \delta + ack + difs + \delta \quad (4.8)$$

$$T_c^{bas} = H + E\{L\} + difs + \delta \quad (4.9)$$

$$T_s^{rts} = rts + sifs + \delta + cts + sifs + \delta + H \\ + E\{L\} + sifs + \delta + ack + difs + \delta \quad (4.10)$$

$$T_c^{rts} = rts + difs + \delta \quad (4.11)$$

$rts$ ,  $cts$ ,  $ack$ ,  $H$  and  $E\{L\}$  are the transmission times of RTS, CTS, ACK, packet header (physical layer plus MAC layer) and data packets, respectively.  $E\{L\} = L$  for a fixed packet size.  $\delta$  is the propagation delay.  $sifs$  (Short Interframe Space),  $difs$  (Distributed Interframe Space) and other specific values for DSSS are defined in the IEEE 802.11 Standard [11]. Since  $T_{er}$  assumes only basic access [134],  $T_{er} = T_c = T_s$ . This is incorrect if RTS/CTS is enabled.

As modeled in [134], the average packet delay  $E[D]$  of a packet that is not discarded, is given by:

$$E[D] = E[X] \times E[slot] \quad (4.12)$$

where  $E[X]$  is the average number of slot times required to successfully transmit a packet and is given by:

$$E[X] = \sum_{i=0}^m \left[ \frac{(p^i - p^{m+1}) \frac{W_{i+1}}{2}}{1 - p^{m+1}} \right] \quad (4.13)$$

$(1 - p^{m+1})$  is the probability that the packet is not dropped,  $(p^i - p^{m+1})/(1 - p^{m+1})$  is the probability that a packet that is not dropped at the stage  $i$ , and  $W_i$  is the contention window size at stage  $i$ .

From these previous models, we build a new model for wireless packet dispersion. The dispersion  $T$  between two packets in a packet pair is the delay between the arrival times of the first and second packets. Since the model must include both the delay before the transmission of the second packet,  $E[D]$ , and the time to transmit it,  $T_s$ , the dispersion is represented by [54]:

$$E[T] = E[D] + T_s \quad (4.14)$$

where  $T_s$  is modeled by Equation 4.8 or Equation 4.10.  $E[D]$  is a function of the average slot time length given in Equation 4.3 and the average number of slot times to transmit a data packet. Since  $E[D]$  depends on  $n$  (the number of nodes in the contention domain), the wireless link rate  $C_l$  and average packet size  $L$ , we have:

$$E[D] = d(C_l, L, n) \quad (4.15)$$

Similarly, to include the impact caused by wireless link rate  $C_l$  and the probe packet size  $L$ , Equation 4.8 and Equation 4.10 are modified as:

$$T_s = t_s(C_l, L) \quad (4.16)$$

Thus, the packet dispersion estimation  $C_{est}$  can be computed as:

$$E[C_{est}] = \frac{L}{E[T]} = \frac{L}{d(C_l, L, n) + t_s(C_l, L)} \quad (4.17)$$

Note that while  $C_{est}$  is the average bandwidth estimate from packet dispersions, it does not equal throughput. Throughput, the average achievable data rate, takes into consideration the probability of transmitting and the probability of successful transmission.

WLAN contending traffic causes extra delay to the probing packets. For packet dispersion techniques, this extra delay can result in an under-estimate of the capacity. The impact caused by contending traffic is more sensitive to the number of nodes in the network than the traffic load at the individual nodes. Assuming each WLAN node always has data to send,  $E[D]$  includes the contending traffic based on the number of WLAN nodes.

Wireless channel conditions can be characterized by received signal strength indication (RSSI), SNR, and BER. However, modeling the effects of such channel conditions on packet dispersion is left as future work. Instead, our simplified model only uses BER to characterize the channel condition and assumes the other wireless factors impact BER. As the number of backoffs increases,  $E[D]$  increases exponentially until it successfully transmits or until the retry limit has been exceeded.

The impact of the channel condition on the bandwidth estimation is evaluated by modeling the packet dispersion variance,  $V[T]$ . Assuming the variance is caused by contention and errors, similar to Equation 4.14:

$$\begin{aligned} V[T] &= V\{D + T_s\} = \sum_{i=0}^m (\bar{D}_k - E[D])^2 P_i \\ &= \sum_{i=0}^m \left[ \sum_{k=0}^i \frac{E[slot](W_k + 1)}{2} + iT_* - E[D] \right]^2 P_i \end{aligned} \quad (4.18)$$

where  $P_i = (p^i - p^{m+1})/(1 - p^{m+1})$ , is the probability that a packet is not dropped at

stage  $i$ .  $\bar{D}_k$  is the average delay for  $k$  stage backoff given by  $\bar{D}_k = \sum_{k=0}^i \frac{E[\text{slot}](W_k+1)}{2} + iT_*$ , where  $T_*$  is the average delay time due to a collision or packet error:

$$T_*^{rts} = \frac{T_c^{rts} P_c^{rts} + \bar{T}_{er}^{rts} \bar{P}_{er}^{rts}}{P_c + \bar{P}_{er}^{rts}} \quad (4.19)$$

$$T_*^{bas} = T_c^{bas} = T_{er}^{bas} \quad (4.20)$$

The average delay caused by a packet error for RTS/CTS access method  $\bar{T}_{er}^{rts}$  can be modeled as:

$$\bar{T}_{er}^{rts} = \frac{T_c^{rts} (P_{er}^{rts} + P_{er}^{cts}) + T_s^{rts} (P_{er}^{data} + P_{er}^{ack})}{\bar{P}_{er}^{rts}} \quad (4.21)$$

and the expected overall probability of a packet error for RTS/CTS access  $\bar{P}_{er}^{rts}$  can be modeled as:

$$\bar{P}_{er}^{rts} = P_{er}^{rts} + P_{er}^{cts} + P_{er}^{data} + P_{er}^{ack} \quad (4.22)$$

where the  $P_{er}^{rts}$ ,  $P_{er}^{cts}$ ,  $P_{er}^{data}$ ,  $P_{er}^{ack}$  are the probabilities that a packet error occurs in RTS, CTS, DATA and ACK packets, respectively.

Given that the capacity function  $C_{est} = L/T$  is twice differentiable and the mean and variance of  $T$  are finite, the variance of the estimated capacity can be approximated by the Delta method using second-order Taylor expansions:<sup>5</sup>

$$V[C_{est}] \approx V[T] \left[ \left( \frac{L}{T} \right)' \right]_{E[T]}^2 = V[T] \left( \frac{L}{E^2[T]} \right)^2 \quad (4.23)$$

### 4.3.2 Model Validation

This section provides two distinct sets of validation results for the packet dispersion model. First, the ideal WLAN channel dispersion model with no contending traffic or bit errors

---

<sup>5</sup><http://en.wikipedia.org/wiki/Variance>

is validated via both NS-2 simulations and wireless testbed measurements. Then with contention and BER included in the models, a large set of simulations with randomized wireless nodes are used to validate the more complex packet dispersion model. Table 4.1 lists the set of MAC layer parameters used for all instances of the dispersion model and all the reported simulations.

Parameterization for the packet dispersion model required creating programs based on the equations in Section 4.3 to obtain the numerical solutions for  $p$  and  $\tau$  since no closed-form solutions exist. Furthermore, the computation of the times for  $T_s$  and  $T_c$  was modified to account for the lower transmission rate of the PLCP header [11].

The ideal WLAN scenario consists of an AP and a single wireless client with both basic and the RTS/CTS access methods possible. In this case, since  $E[slot]$  is simply the slot time  $\sigma$  and  $E[D]$  is the backoff between two successive packets with contention window size  $W_{min}$ , the delay model simplifies to:

$$E[D] = \frac{E[slot](W_{min} + 1)}{2} = \frac{\sigma(W_{min} + 1)}{2} \quad (4.24)$$

The ideal simulations varies the packet size from 100 to 1500 bytes with the wireless capacity set to 11 Mbps. The wireless testbed consists of a Windows XP PC sending packet pairs over a 100 Mbps channel through a Netgear 802.11b AP/router to a Windows XP laptop wirelessly connected to the AP via a Dell TrueMobile 1300 Mini PCI card. The PC sends at full capacity, with packet pairs at the specified packet size. The wireless receiver computes estimations using packet pair dispersion.

Figure 4.6 graphs the bandwidth estimation results from the models, simulations and measurements for the ideal WLAN scenario. For each packet size in either RTS/CTS or basic access mode (BAS), the simulation results and the error bar in the figure are the average and standard deviation from 500 packet pair estimations. The measurement includes 100 packet pair dispersions with the same packet sizes<sup>6</sup> and channel rate. For basic access,

---

<sup>6</sup>Due to the Ethernet MTU limit, 1460 application layer bytes are used instead of 1500 bytes as the maximum packet size in the measurement.

Table 4.1: IEEE 802.11 Physical Layer Parameters (DSSS)

	DSSS modulation
$W_{min}$	32
$W_{max}$	1024
MAC header	34 bytes
Phy header	24 bytes
ACK	38 bytes
CTS	38 bytes
RTS	44 bytes
Slot time	20 $\mu$ sec
SIFS	10 $\mu$ sec
DIFS	50 $\mu$ sec

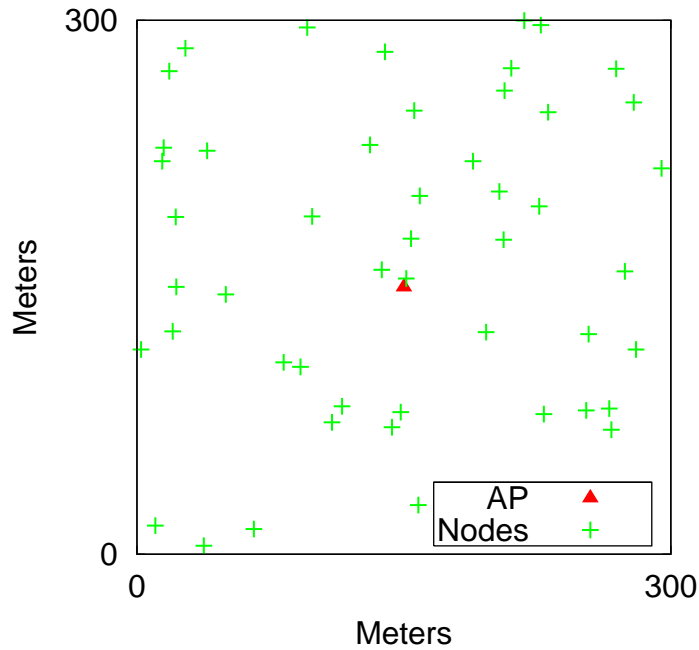


Figure 4.5: A Randomly Generated Topology with 50 Nodes and an AP in the Center

the model, simulation and measurement results all closely match. This indicates that the model and simulation both provide high fidelity compared to real 802.11b networks. With RTS/CTS enabled, the measurement results are slightly higher than the model and simulation. In-depth analysis shows that this difference is because the testbed sends management frames, such as RTS/CTS/ACK at 2 Mbps, which is higher than the 1 Mbps base rate used in the model and simulation.

A random simulation topology was created (see Figure 4.5) to study the packet dispersion model with contention and bit errors. Since all the nodes are within transmission range of each other, there are no hidden terminals in this topology. Bandwidth estimation is computed with  $L/\bar{T}$ , where  $\bar{T}$  is the average dispersion time from 500 packet pairs.

The number of sending nodes is increased from 2 to 50 to increase the contention level in the models. By assuming every node always has traffic to send, the model estimates packet dispersion under saturation conditions. To simulate saturation, an upstream 10 Mbps CBR flow is sent from each wireless node to the AP, while the packet pair traffic is sent downstream from the AP to a single node. The wireless data rate is fixed at 11 Mbps and both the CBR flows and the packet pair probes send 1500-byte packets. To avoid severely impacting the estimation results, packet pairs are sent at a lower rate of 100 Kbps.

All the contention simulations without errors were repeated with a typical wireless network bit error rates of  $1 \times 10^{-5}$ . With  $B_{mod}$  and  $B_{sim}$  as the modeled and simulated bandwidth estimations, respectively, the relative error  $E$  for each topology from 2 to 50 nodes is defined as:

$$E = \frac{|B_{sim} - B_{mod}|}{B_{sim}} \quad (4.25)$$

and the mean and standard deviation of error are defined as the average and standard deviation of the  $E$  values. Table 4.2 summarizes the dispersion model with contention validation results performed under different channel conditions and shows a close match between the model and simulations for both ideal and bit error channels. Further model

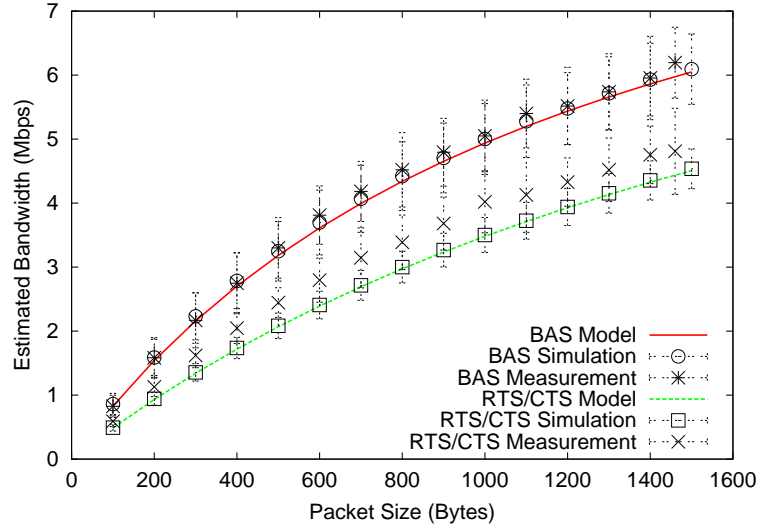


Figure 4.6: Bandwidth Estimation Validation for an Ideal WLAN

parameter tests comparing modeled throughput to simulated throughput generally yield a close match. Additional details of the parameter validation process are in [158].

## 4.4 Analysis

### 4.4.1 Packet Dispersion in 802.11

Since the model developed does not extend over the whole network path, the analysis focuses on a WLAN with the assumptions that all packet dispersion occurs at the AP and that there is no crossing traffic in the downstream direction.

Understanding packet dispersion in wireless networks requires separating the non-saturated and saturated scenarios. Given a non-saturated WLAN with low BER where

Table 4.2: Errors in the Bandwidth Estimation Model Compared with Simulations

	Error Free		$BER = 10^{-5}$	
	RTS/CTS	Basic	RTS/CTS	Basic
Mean Error	8.05%	4.90%	9.40%	7.67%
Stdev	6.72%	4.28%	5.30%	3.82%

the probability of packet pair dispersion due to contending traffic is relatively low, the packet pair dispersion estimate represents the maximum channel capability for forwarding traffic for a given packet size. However, this capability includes overhead caused not only by packet headers, but also by the random delay between successive packets, MAC layer contention backoff, MAC layer retries and basic two way hand-shake (DATA/ACK) or four hand-shake (RTS/CTS/DATA/ACK). Emphasizing this difference, the term *effective capacity* indicates the maximum capability of the wireless network to deliver network layer traffic. Unlike in wired networks, wireless dynamic rate adaptation alters effective capacity by adjusting the packet transmission rate. Therefore, effective capacity is defined as a function of time and packet size:

$$C_e = \frac{\int_{t_0}^{t_1} \frac{L}{\bar{T}(t)} dt}{t_1 - t_0} \quad (4.26)$$

where  $\bar{T}(t)$  is the average packet pair dispersion at time  $t$ . Moreover, given discrete packet pair samples, the effective capacity is:

$$C_e = \frac{\sum_{i=1}^n \frac{L}{T(i)}}{n} \quad (4.27)$$

where  $n$  is the number of samples from packet pair measurements and  $T(i)$  is the dispersion of the  $n$ th packet pair.

However, in a wireless network with considerable contending traffic or BER, MAC layer retries due to bit errors and collisions between the probing traffic and contending traffic add delay to packet dispersion. Hence, average packet pair dispersion represents the average time used to forward one single packet. This represents the traffic the network can forward given the contending traffic and BER. The rate computed from the average packet pair dispersion is not the available bandwidth because it includes the impact of the contending traffic. This metric, referred to as *achievable throughput* for the current level of contending traffic, is:

$$A_t \frac{L}{\frac{1}{n} \sum_{i=1}^n T(i)} \quad (4.28)$$

MAC layer retries caused by contention and BER are major sources of achievable throughput degradation. Achievable throughput is greater than available bandwidth because it aggressively takes bandwidth from the crossing traffic and it represents the average throughput along the same direction as the probing traffic. Therefore, the following relationship exists among the available bandwidth ( $A$ ), achievable throughput ( $A_t$ ) and effective capacity ( $C_e$ ):  $A \leq A_t \leq C_e$ . Moreover, in a non-saturated WLAN that has available bandwidth for new traffic, the achievable throughput can be modeled using a fluid model because contending effects can be ignored if total throughput in the wireless network is less than the effective capacity.

A saturated wireless network is caused by multiple non-responsive traffic sources, such as UDP flows, transmitting above the flow's fair-share bandwidth. However, in a saturated wireless network no bandwidth is available, and each node contends with other traffic to access the wireless channel. Overall throughput is reduced by contending effects and achievable throughput represents the fair share of the effective capacity for all the active contending nodes.

To illustrate achievable throughput in a saturated wireless network, packet pair results are compared with CBR throughput using the simulation topology in Figure 4.5. Achievable throughput is computed from the dispersion time of 500 packet pairs with a sending rate of 100 Kbps and a 10 Mbps CBR flow. The contending traffic at each node is 10 Mbps, and the packet size for packet pairs, contending traffic and CBR traffic are all 1500 bytes. Figure 4.7 shows that the packet pair estimates are nearly the same as the average CBR throughput for both the model and simulations. In this saturated scenario, CBR throughput represents achievable throughput.

Packet train techniques apply the same packet dispersion ideas to packet pair dispersions. However, the large number of packets in a train make it more vulnerable to contending traffic. Therefore, packet train dispersion in wireless networks does not measure

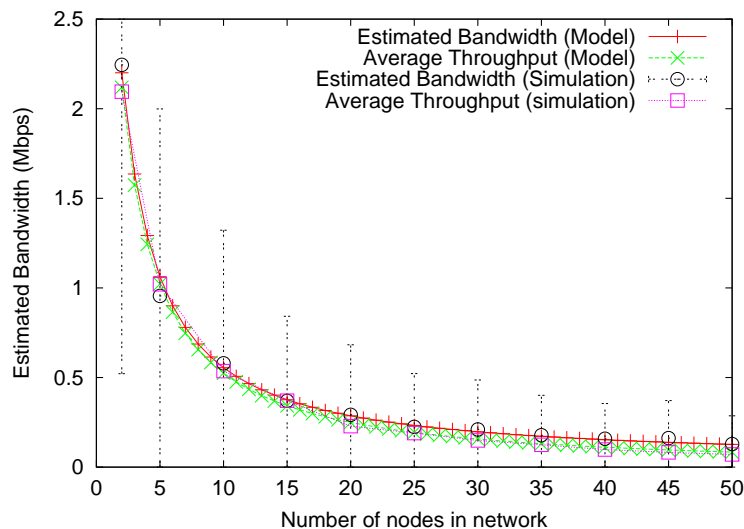


Figure 4.7: Packet Pair Estimations and CBR Throughput

the effective capacity, but rather indicates the achievable throughput.

Wireless networks are a mixture of contending, bit errors and rate adaptation conditions. It is difficult to distinguish packet dispersion results that are impacted by MAC layer retries from results due to WLAN rate adaptation. Even though the achievable throughput can be estimated, it can be difficult to determine the effective capacity from the estimation results in such mixed channel conditions. Therefore, other techniques may be needed to remove MAC layer retries caused by contention and BER to get more accurate effective capacity estimates.

#### 4.4.2 Analysis of the Estimation Results

As discussed in [127], packet size significantly impacts the measurement of wireless network throughput because of the wireless overhead. Similarly, probe packet size effects estimation results dramatically. Generally, as packet size increases relative overhead due to headers decreases. For example, Figure 4.6 depicts the effective capacity of an ideal channel at 11 Mbps, with both basic and RTS/CTS access methods. To effectively estimate bandwidth, probing packet size must be close to the packet size of the applications that use the bandwidth estimation. For example, streaming video should use a probing packet size close to

the video packet size to pick an effective streaming rate.

MAC layer rate adaptation impacts the effective capacity significantly. However without knowing wireless channel conditions and the vendor-implemented rate adaptation algorithm, it is difficult to model the practical effects of rate adaptation. Figure 4.8 illustrates the relationship between effective capacity and the channel rate in a ideal condition with 1500 byte packets for both basic and RTS/CTS access methods. Although the adaptation algorithm and channel conditions may vary the result of rate adaptation, the relationship between the channel rate and the effective capacity still holds because of the statistical nature of the model. Therefore, the model can be used to predict the effective capacity by using the average channel rate instead of a fixed data rate.

Bit errors reduce achievable throughput in wireless networks because MAC layer retries reduce the efficiency of the wireless network. Moreover, packet drops due to exceeding MAC layer retry limits also directly reduce the achievable throughput in wireless networks. Figure 4.9 shows the packet dispersion results of the model and simulation for 1500-byte packets sent on a 5-node (contending node) wireless basic access network with BER ranging from  $1 \times 10^{-7}$  to  $1 \times 10^{-3}$ . Achievable throughput decreases as the BER increases. As the BER reaches approximately  $1 \times 10^{-3}$ , the wireless network gets almost no achievable throughput. This is because the high BER of  $1 \times 10^{-3}$  results in a packet error rate closed to one, thus almost no packets can be successfully transmitted even with maximum number of retries.

The RTS/CTS four-way handshake lowers the impact of hidden terminals by reducing the cost of collisions while introducing considerable WLAN overhead. Without considering the hidden terminal problem, RTS/CTS can still improve the network average throughput under high traffic load conditions. Figure 4.10 uses the model to illustrate the crossover point for 1500-byte packets where RTS/CTS gets higher achievable throughput compared to basic access for different link rates. The crossover point is measured as the number of fully loaded nodes in the wireless network. The higher the link data rate, the more likely basic mode will have a higher throughput than RTS/CTS. For example, RTS/CTS will

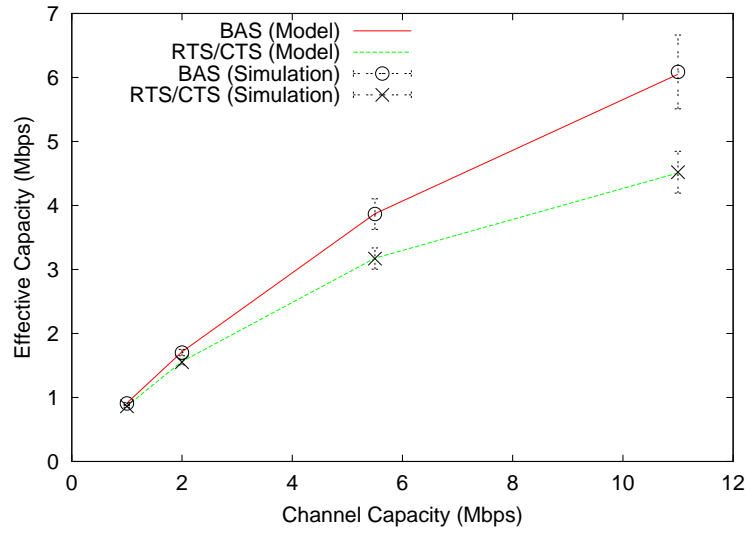


Figure 4.8: The Impact of Channel Data Rate

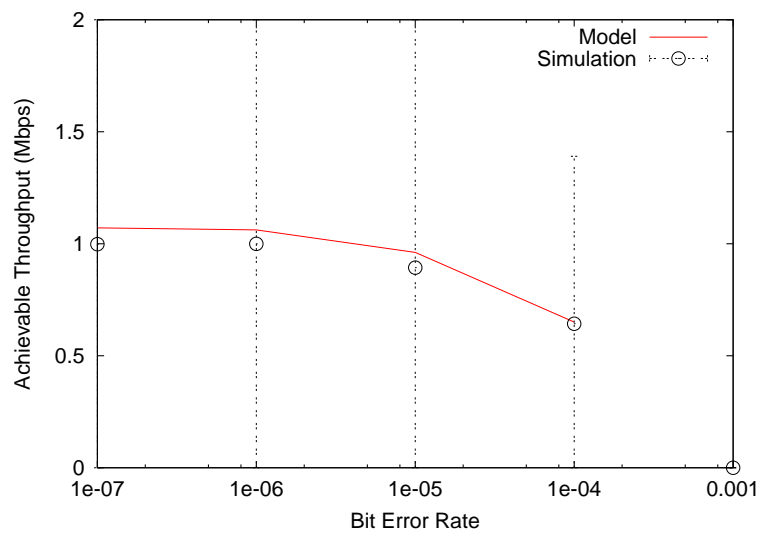


Figure 4.9: The Impact of Channel Bit Error Rate (BER)

only have a higher throughput if there are more than 57 fully loaded nodes in an 11 Mbps network. Moreover, BER increases the crossover point where RTS/CTS achieves higher throughput than basic access. This figure demonstrates why RTS/CTS is disabled in most wireless networks.

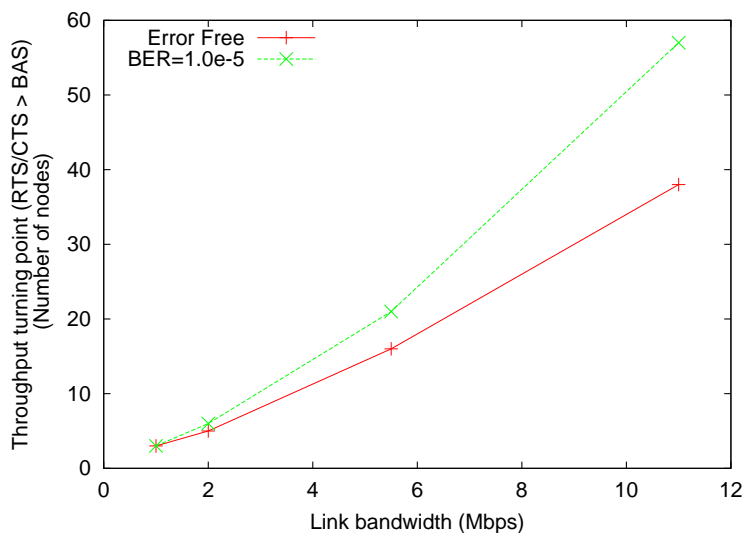


Figure 4.10: Comparison of RTS/CTS with Basic Access for Achievable Throughput

#### 4.4.3 Analysis on the Variance of the Bandwidth Estimation

The packet dispersion model provides the variance and standard deviation of the bandwidth estimates. Figure 4.11 shows the standard deviation of the estimations from the model and simulations with 1500-byte packets and basic access. The standard deviation of the simulated estimation is computed based on 500 packet pair dispersions. As the traffic load increases, the standard deviation decreases because more contending sources more evenly distribute backoff delay across multiple estimates. However, for fewer than five nodes, the modeled standard deviations do not match the simulation results. This is because the variance of a randomly selected number of backoff time slots in the contention window is not included in Equation 4.18. With high traffic load, the variance from multiple random backoff time slots can be safely ignored because it is relatively small compared to the

variance due to the number of retries. However, retry probability is low for the network with fewer than five nodes. Thus the time slot variance dominates the overall variance and causes the mismatch between the model and simulation.

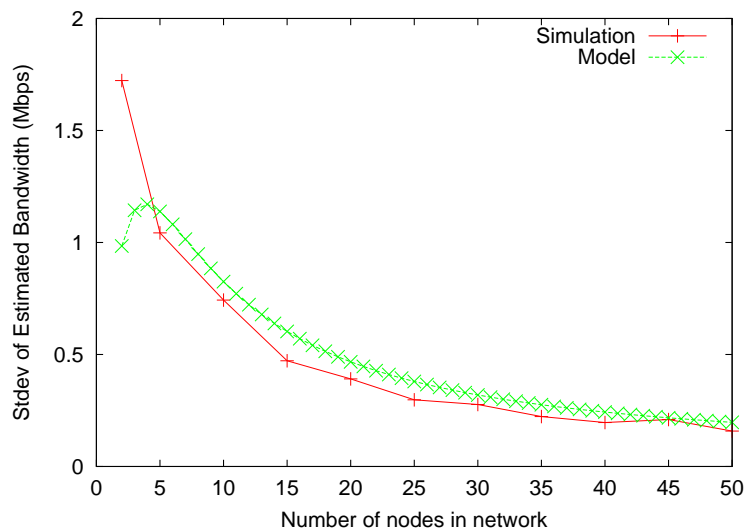


Figure 4.11: Simulating and Modeling Standard Deviations of Estimation

Analysis of variance of the bandwidth estimations is helpful for designing new bandwidth estimation algorithms, such as for deciding the number of packet pairs in an estimate or the length of a packet train. Furthermore, packet dispersion variance also provides additional information for inferring network conditions, such as the traffic load and the bit error rate.

Packet size also affects the variance in the bandwidth estimations. Larger packet sizes yield a relatively larger variance. Figure 4.12 depicts the standard deviation of packet pair estimations in a basic access, 5-node wireless network, with no errors and  $BER = 10^{-5}$ . The BER curve shows a higher standard deviation than the error free channel for the same packet size. This is because packet error rate increases with BER, and this raises the probability of MAC layer retries which produces more packet pair estimation variance.

Figure 4.13 shows the standard deviation of packet pair estimations with 1500-byte packets in a 5-node wireless network with no errors and  $BER = 10^{-5}$ . The variance of

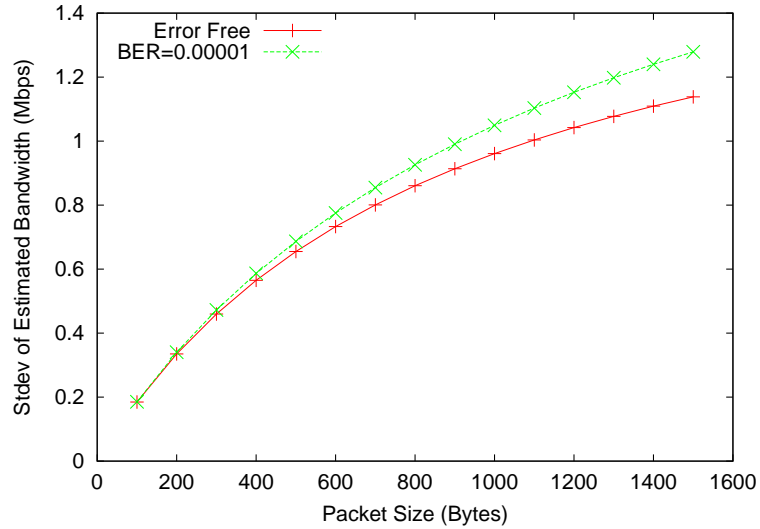


Figure 4.12: The Impact of Packet Sizes on the Standard Deviation of Bandwidth Estimation

bandwidth estimations increases as the channel data rate increases. This implies that the higher the link data rate, the higher the relative error in the estimation. Compared to the channel without errors, the channel with errors has a higher variance for all data rates. This is because the bit errors cause more MAC layer retries and more variance in the estimation results.

Bit errors impact not only the packet dispersion result in wireless networks, but also its variance. Figure 4.14 shows the standard deviation for 1500-byte packets on a 5-node 11 Mbps wireless network with BER ranging from  $1 \times 10^{-7}$  to  $1 \times 10^{-3}$ . For BERs less than  $10^{-5}$ , the standard deviation of the bandwidth estimations increases as the BER increases. The variance starts to decrease as BER increases over  $10^{-5}$ . This is because the number of retries reaches the retransmission limit, therefore reducing the variance in the backoff delay across multiple packet pairs. In fact, for a BER higher than  $10^{-4}$ , the packet drop rate is so high that only a few packets get through the network (with a large number of retries). Note, the RTS/CTS access method has a lower standard deviation than the basic method in all cases.

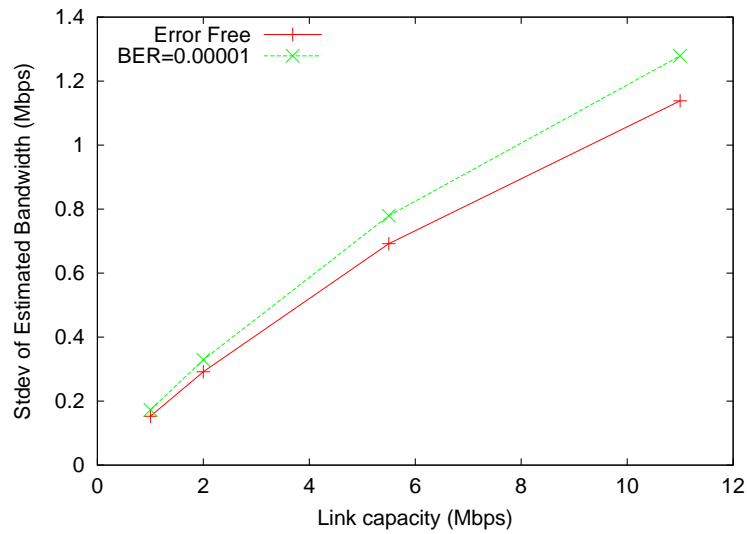


Figure 4.13: The Impact of Channel Data Rate on the Standard Deviation of Bandwidth Estimation

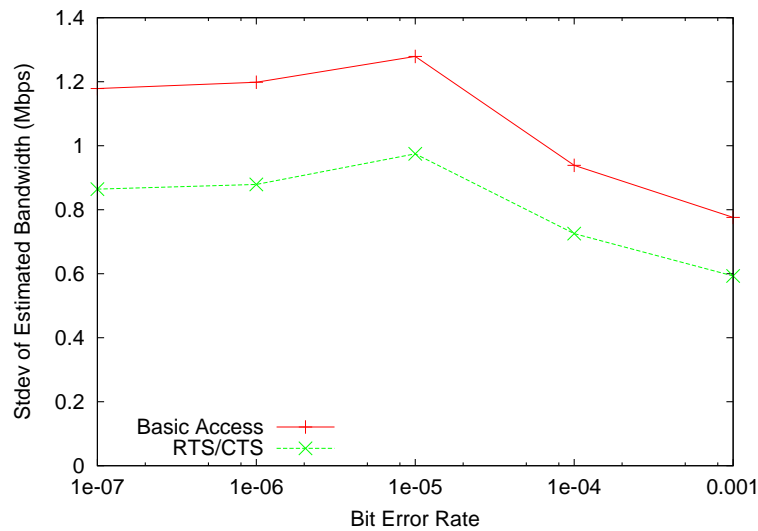


Figure 4.14: The Impact of BER on the Standard Deviation of Bandwidth Estimation

## 4.5 Summary

This chapter presents an analytic model to investigate packet dispersion behavior in IEEE 802.11 wireless networks. The packet dispersion model is validated by an extended NS-2 simulator and with wireless 802.11b testbed measurements. Utilizing the packet dispersion model, the following observations can be made:

1. Packet dispersion measures the *effective capacity* and the *achievable throughput* of a wireless network instead of the capacity as in a wired network. Effective capacity, defined as a function of packet size and time, represents the ability of a wireless network to forward data over a given time period. Achievable throughput is the maximum throughput that a node can achieve when contending with other existing traffic on a wireless network.
2. Wireless channel conditions, such as packet sizes, link data rate, BER and RTS/CTS access method impact the bandwidth estimation results and the variance of the results. The packet size and link data rate have a positive correlation with both the bandwidth estimations and variances of the estimations. The BER of the channel has a negative correlation with the bandwidth estimations and a positive correlation with variances of the estimations. RTS/CTS reduces estimated bandwidth and the variance of the estimations.

With the model of packet dispersion and the studies of the impacts of wireless network conditions, a novel bandwidth estimation tool for wireless networks, the Wireless Bandwidth Estimation tool (WBest) is developed in Chapter 5.



## Chapter 5

# WBest: Wireless Bandwidth Estimation Tool

This chapter presents a new Wireless Bandwidth Estimation Tool (WBest) designed for fast, non-intrusive, accurate estimation of available bandwidth and variance of available bandwidth in IEEE 802.11 networks. WBest applies a two-step algorithm: 1) a packet pair technique to estimate the effective capacity of the wireless network; 2) a packet train technique to estimate the achievable throughput and report the inferred mean and standard deviation of available bandwidth. Section 5.1 gives a brief overview of the bandwidth estimation techniques in wireless networks. Section 5.2 discusses the requirements of multimedia streaming applications. Section 5.3 discusses the WBest algorithm and related issues. Section 5.4 describes the experimental setup used to evaluate WBest. Section 5.5 analyzes the experimental results. Finally, Section 5.6 summarizes the chapter.

### 5.1 Overview

Due to the shared nature of wireless network communication and MAC layer mechanisms such as wireless layer retries and dynamic rate adaptation, bandwidth estimation is far more challenging when the underlying network includes wireless networks. Fluctuating

wireless channel conditions cause variability in wireless capacity and available bandwidth, and other wireless factors such as reception signal strength and bit error rates (BER) due to path loss, fading, interference and contention limit the effective bandwidth that the wireless network actually provides. While providing satisfying results on wired networks, current bandwidth estimation tools have been shown [90, 91, 127, 154] to be adversely impacted by IEEE 802.11 wireless network conditions.

Tools that provide only capacity estimates are not useful for Internet applications that adjust their traffic rate in response to other concurrent flows. Moreover, applications such as multimedia streaming need an available bandwidth estimate within a few seconds to avoid client-side buffer underflows and to satisfy users waiting to use the application. This implies a much faster convergence time requirement than some bandwidth estimation tools provide. The variability of the wireless channel conditions implies that multiple bandwidth estimation invocations are typically used within a single application stream. This adds an additional requirement that a bandwidth estimation tool must be minimally intrusive so as to not adversely impact the application's performance during measurements.

As discussed in Chapter 3, most available bandwidth estimation techniques are designed to provide accurate bandwidth information for wired networks at the cost of long convergence times and high intrusiveness. Recent research has proposed improvements to bandwidth estimation specific to wireless networks. Most of these wireless bandwidth estimation techniques are based on improvements to existing bandwidth estimation techniques designed for wired networks. For example, *SenProbe* [159] and *AdhocProbe* [160] present the capacity estimations for wireless sensor and Ad-hoc networks. They are based on *CapProbe*, which applies packet delay and dispersion techniques to estimate the capacity in wired networks. However, *SenProbe* and *AdhocProbe* only provide an estimation of capacity and are designed for Sensor and Ad-hoc networks not infrastructure networks. *ProbeGap* [90] estimates the fraction of time that a link is idle by probing for gaps in the one-way packet delays, and then use this idle fraction to estimate the available bandwidth in broadband access networks including IEEE 802.11 networks. However, *ProbeGap* does

not provide capacity estimation and needs to use third party capacity estimation tools, such as pathrate [109]. *DietTOPP* [161] uses a reduced TOPP [115] algorithm with a modified search algorithm to determine available bandwidth in wireless networks. While these techniques [90, 161] have shown promise in improving the accuracy of available bandwidth estimation in wireless networks, they do not consider the improvement of convergence time and intrusiveness. The evaluations from research [90] and [161] focus only on the accuracy in comparison with existing wired approaches, such as pathload [113] and spruce [121].

In addition to application layer bandwidth estimation techniques, Shah et al. [127] and Lee et al. [162] present cross layer approaches to estimate the available bandwidth in IEEE 802.11 wireless networks. For instance, [162] uses MAC layer NAV to measure the busy and idle times and uses these MAC layer measurements to infer the available bandwidth in the wireless network. [127] measures the channel busy time and uses it to characterize the impact of contention and fading, thus providing link layer available bandwidth information. However, both [127] and [162] were developed based on simulations and have not been evaluated in real systems. Moreover, their dependence upon cross layer techniques makes them difficult to develop widely.

The issues of inaccurate results, high intrusiveness and long convergence time make it difficult to apply current bandwidth estimation mechanisms to applications, such as multimedia streaming, over wireless networks and leads to the development of the Wireless Bandwidth Estimation tool (WBest). To address accuracy and convergence, WBest employs packet dispersion techniques to provide capacity and available bandwidth information for the underlying wireless networks. Chapter 4 models packet dispersion behavior in wireless networks under varying conditions. Using an analytical model, two packet dispersion measures, *effective capacity* and *achievable throughput*, were shown to be suitable for wireless networks. Combining these two metrics, WBest employs a two-step algorithm to determine available bandwidth. In the first step, a packet pair technique estimates the effective capacity of the wireless network. In the second step a packet train scheme determines achievable throughput and infers available bandwidth. By modeling WBest, this

research investigates the tradeoffs of accuracy and convergence time, and possible sources of error to optimize the algorithm. Thorough evaluation in a wireless testbed shows WBest performs better in terms of accuracy, intrusiveness and convergence time than three current available bandwidth estimation tools: *IGI/PTR* [117], *pathChirp* [116] and *pathload* [113]. WBest fits the practical needs of many applications such as multimedia streaming that require low cost and accurate bandwidth estimations.

## 5.2 Requirements of Wireless Streaming Applications

Bandwidth estimation techniques have been widely studied in recent years. However, there are few studies that discuss applying such techniques in real applications. Different applications and network environments may have distinct requirements for bandwidth estimation. For instance, network management systems require accurate bandwidth estimations and usually can tolerate a long convergence time. Streaming or interactive applications, on the other hand, prefer to have low convergence time and do not demand precisely accurate bandwidth estimations. It is difficult to design a general purpose bandwidth estimation tool for all application types. Therefore, to evaluate the applicability of a particular bandwidth tool the application and the network context must be considered.

The previous bandwidth estimation tools usually target network management, thus the emphasis is on accurate bandwidth results. The applied bandwidth metrics may be either capacity or available bandwidth of the backbone networks. To describe the differences between the bandwidth estimation tool required by multimedia streaming applications and general purpose bandwidth estimation tools, the following evaluation criteria are considered: measured metrics, accuracy, convergence time, intrusiveness, robustness and usability in wireless networks.

The bandwidth metrics used in bandwidth estimation tools in wired networks need to be redefined in wireless networks. For example, the capacity is not constant in wireless networks, instead, the *effective capacity* [163] that takes the dynamical capacity changes into consideration is preferred. Similarly, as discussed in Section 5.3, the available bandwidth

is not defined as the capacity excluding the amount of cross traffic. Instead, it should take both capacity sharing and contending effects into consideration. However, most current available bandwidth estimation tools do not include these issues. For example, IGI [117] assumes a known, fixed capacity. For a multimedia streaming application using bandwidth estimation to adapt the sending rate and optimize the client side buffer, the most useful bandwidth metrics are the available bandwidth and the variance in available bandwidth. Therefore, tools that only estimate capacity can not be used satisfactorily by streaming applications. Moreover, tools only designed for wired networks need to be improved before they can be applied to streaming applications in wireless networks.

For multimedia streaming applications, estimation accuracy is no longer a primary concern any longer. The reason is that streaming media applications usually scale the sending rate in steps instead of smoothly. Thus, any bandwidth estimation result with a granularity less than the streaming media encoding levels is sufficient for controlling the media scaling. For example, for a media stream encoded at multiple levels of 700 Kbps, 1.2 Mbps, 2.5 Mbps and 5 Mbps, an available bandwidth estimation of 3.54 Mbps will trigger media scaling down to 2.5 Mbps. However, any estimation result between 2.5 Mbps and 5 Mbps will trigger the same media scaling. Thus, a maximum acceptable estimation error for the 3.54 Mbps estimation result can be computed as  $\min(3.54 - 2.5, 5 - 3.54) = 1.04$  Mbps, which indicates an accuracy limit of 1.04 Mbps for this case. In addition, a higher frequency of media scaling implies a lower perceived quality [164]. Since both the effective capacity and the available bandwidth change dynamically in wireless networks, to reduce unnecessary media scaling actions, a time-based average measurement or a distribution in bandwidth are preferred over an accurate instantaneous bandwidth measurement. Therefore, an accurate instantaneous estimate is not a critical requirement for streaming media applications.

The convergence time for a bandwidth estimation is a major concern for streaming media applications over wireless networks. Multimedia streaming applications need an available bandwidth estimate within a few seconds to avoid client-side buffer underflows

and to satisfy users waiting to use the application. This implies a much faster convergence time requirement than most current bandwidth estimation tools provide. Even though the media scaling does not need to execute at the same frequency as the bandwidth estimation, a short convergence time may provide more estimations in the same time period, thus may provide a better chance for a filtering or smoothing algorithm to find a reasonably accurate average estimation of bandwidth. A shorter convergence time improves the capability to capture the variation in the effective capacity or available bandwidth. As shown in recent research [10], the variation in the wireless available bandwidth may degrade the video performance even if the average bandwidth is sufficient for the streaming rate.

Intrusiveness is another major concern for evaluating the bandwidth estimation techniques over wireless networks. Streaming applications tend to perform bandwidth estimation frequently during the whole streaming session. Therefore, low intrusiveness is critical for reducing the impact caused by the probing traffic itself. Available bandwidth may be reduced due to the probing traffic that saturates the wireless network. As a result of bandwidth reduction, the performance of streaming multimedia applications can also be impacted by the heavy probing traffic.

Another important issue related to accuracy, convergence time and intrusiveness are the *competing effects* caused by the probing traffic. With self-loading or packet dispersion techniques, the probing traffic will temporarily increase the queuing delay of the crossing traffic. Responsive crossing traffic, such as TCP flows, will respond to the increases in RTT and reduce the sending rate. Therefore, the available bandwidth estimation will be an overestimate if the convergence time is long enough for the TCP flows to reduce their sending rates. As discussed in [165], TCP throughput  $B$  can be approximated as the equation:  $B = 1/RTT \sqrt{3/2bp}$ , where  $p$  is the probability that a packet is lost and  $b$  is the number of packets that are acknowledged by a received ACK. If we assume that the probing traffic will not overflow the AP queue, the TCP congestion control response to the RTT changes will reduce the TCP throughput in a few RTTs. For example, the time for a packet train with 30 packets to pass an AP with an effective capacity of 6 Mbps is about

58 ms. If we assume the TCP RTT is in the same range, the throughput of the TCP traffic will decreased to almost half according to the equation. Therefore, we attempt to complete the bandwidth estimation in a few RTTs so that the TCP crossing traffic is not unduly impacted by the congestion caused by probing traffic. In fact, this competing effect is not specific to streaming applications or wireless networks, but is an issue for all bandwidth estimation tools. Possible solutions include reducing the intrusiveness and convergence time, or approximating the amount of overestimation and compensate for it in the final estimation of available bandwidth.

Robustness and usability are mandatory for all bandwidth estimation techniques. Since streaming servers and clients are designed to work in client/server mode, the usability of working in uncooperative environments is not an issue any longer. However, to assure the applicability to streaming applications, the bandwidth estimation tool should have a relative consistent convergence time and intrusiveness under various channel conditions. That is, the applications expect the convergence time and intrusiveness to be bounded by upper limits, so that the application can predict the cost in term of time and intrusiveness on performing bandwidth estimation.

In summary, multimedia streaming in wireless networks requires a bandwidth tool with fast convergence time, low intrusiveness and reasonable accuracy. For the initializing bandwidth estimation of multimedia applications, we expect the bandwidth estimation to be completed in a few RTTs so that it will not add extra delay to the starting delay of the streaming playout. For estimations during streaming, we expect convergence times smaller than the buffer time of the streaming application, which means an in-time adaptation before the buffer underflow. Given the expected convergence times of a few RTTs, most general purpose bandwidth estimation tools are not satisfactory for wireless multimedia applications.

### 5.3 WBest algorithm

This section introduces WBest, an algorithm to estimate both effective capacity and available bandwidth on a network path where the last hop is over IEEE 802.11b/g Distributed Coordination Function (DCF) wireless networks. Applying WBest to Point Coordination Function (PCF) wireless network is left as future work.

Figure 5.1 depicts a typical network environment where an application server with a wired Internet connection sends traffic along the network path to a client with a ‘last mile’ wireless connection. To provide better performance, such as to perform media scaling and buffer optimization for a multimedia stream, the application server needs to know the capacity and available bandwidth on the flow path to client A. As discussed in Chapter 4, to characterize the wireless network impact for study, the network traffic is categorized as probing, crossing and contending, as depicted in Figure 5.1. Even though the contending and crossing traffic have different impacts on the probing traffic, the bandwidth estimation tool should be able to handle both of them correctly. In general, capacity estimation should avoid estimation errors caused by crossing and contending traffic. However, available bandwidth estimation should capture the available bandwidth reduction due to both crossing and contending traffic.

#### 5.3.1 Assumptions

To simplify the bandwidth estimation algorithm, the following assumptions are made. These assumptions and possible resultant errors are discussed in more detail later in this section.

1. Assume the last hop wireless network is the *super bottleneck link*<sup>1</sup> on the end-to-end network path. Here the *super bottleneck link* means the last hop wireless network has the smallest available bandwidth (tight link) and the effective capacity of last hop is less than the available bandwidth of all the prior hops along the network path. That

---

<sup>1</sup>The *Bottleneck link* usually indicates both the smallest available bandwidth (tight link) and the smallest capacity (narrow link).

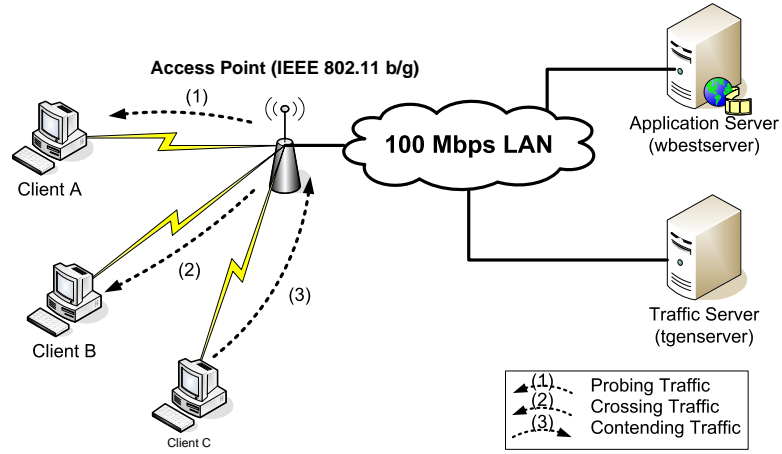


Figure 5.1: Network Path with Last Mile Wireless Network.

is, the following relationships hold:

$$A \leq C_e \leq \min_{i=1, \dots, H-1} (A_i) \quad (5.1)$$

where  $A$  and  $C_e$  are the available bandwidth and effective capacity of the last hop, respectively,  $H$  is the number of hops, and  $A_i$  is the available bandwidth and capacity of the  $i$ th hop. As discussed by Lao, Dovrolis and Sanadidi [166], even with a single bottleneck assumption, PGM technique may underestimate the available bandwidth in a multihop network path. This underestimation occurs because the probing traffic may be dispersed at non-bottleneck hops where the available bandwidth is less than the probing rate. For WBest, by assuming that there are no hops along the network path that have an available bandwidth less than the maximum probing rate  $C_e$ , a packet train sent at rate  $C_e$  is likely to arrive at the last hop at the rate of  $C_e$  [110]. Therefore, the available bandwidth of the super bottleneck wireless network is not underestimated because the probing packets are not dispersed at non-bottleneck hops. Effectively, the super bottleneck assumption reduces available bandwidth estimation for a multihop case into estimating available bandwidth for a single hop case, that of the last wireless hop. As shown in [166], PGM techniques are accurate in this scenario.

The super bottleneck assumption usually holds for most company-wide, campus-wide or hot-spot wireless networks. In practice, the over-provisioning in the backbone network capacity and the usage of Content Distribution Networks (CDNs) increase the available bandwidth in core Internet routers. While on the other hand, because of the nature of shared wireless media, it is difficult to over-provision the wireless network capacity by deploying more wireless APs. This limits the increase in wireless capacity compared with the increase in core Internet routers. Additionally, for these large scale wireless networks, the effective capacity may often be further reduced because of the impact of rate adaptation and bit errors. These facts make last hop wireless networks the super bottleneck for many network paths in practice. If this assumption does not hold, as for some home wireless networks with a broadband Internet connection, the packet train with sending rate  $C_e$  will get dispersed before the last hop and arrive at the last hop with a lower rate than  $C_e$ . This, in turn, will cause a conservative under-estimate of the bandwidth which is typically a better outcome for multimedia streaming applications than an aggressive, over-estimate.

2. Assume First-Come First-Server (FCFS) queues at all routers along the end-to-end path. In addition, we assume that the wireless AP at the super bottleneck has a single FCFS queue. This assumption holds for most network routers and for IEEE 802.11b/g networks currently being deployed. However, for the developing IEEE 802.11e standard, the multiple priority-based queuing mechanism disturbs the FCFS packet dispersion model. For example, high priority traffic may depart before the lower priority probing traffic, thus may produce errors in packet dispersion measurements.
3. Assume no significant changes in network conditions between the two steps (estimation of the effective capacity and estimation of the available bandwidth) of the WBest algorithm. The changes in network conditions due to rate adaptation or mobility may impact the estimation results. For instance, as discussed in Section 5.3.4, the available bandwidth estimation may include errors if the  $C_e$  estimated in the first step

is not applicable for the available bandwidth estimations in the second step. Moreover, the roaming between wireless APs due to mobility could cause route changes, thus impacting the packet dispersion behavior and causing errors in the estimation. However, given WBest algorithm convergence times of milliseconds, the magnitude of these changes is assumed to be minimal over the timescales measured by WBest.

4. Assume packet pairs or trains do not overflow any of the router queues along the flow path. A queue overflow at the last hop wireless AP will impact the accuracy of the estimation results. The possibility of queuing loss is reduced by limiting the number of probing packets sent into the network to be under the queue size of typical wireless APs [167].

### 5.3.2 Algorithm

Similar to the hybrid approach used in research [168], WBest applies a two-step algorithm as shown in Algorithm 5.1 to estimate both effective capacity and available bandwidth. In the first step (lines 1-2),  $n$  packet pairs are sent to estimate the effective capacity  $C_e$ . Effective capacity [158], the maximum capability of the wireless network to deliver network layer traffic, is a function of time and the packet size:

$$C_e = \frac{\int_{t_0}^{t_1} \frac{L}{T(t)} dt}{t_1 - t_0} \quad (5.2)$$

where  $L$  is the packet size, and  $T(t)$  is the packet dispersion at time  $t$ . To use packet dispersion in a discrete environment,  $T_i$ , the  $i$ th packet dispersion at time  $t$ , is used to represent  $T(t)$ .

To minimize the impact of crossing and contending traffic and capture the impact of rate adaptation on measurements of effective capacity, the median of the  $n$  packet pair capacity estimates is used to approximate  $C_e$  in the estimation time period:

$$C_e = \text{median}(C_i), \quad i = 1, \dots, n \quad (5.3)$$

**Algorithm 5.1** WBest Algorithm.

---

**Require:**  $n > 0$  {Measure effective capacity ( $C_e$ )}

- 1: Send  $n$  packet pairs to client
- 2:  $C_e \leftarrow \text{median}(C_i, i = 1, \dots, n)$

**Require:**  $m > 0, C_e > 0$  {Measure available bandwidth ( $A$ )}

- 3: Send packet train with length  $m$  at rate  $C_e$  to client
- 4:  $R \leftarrow \frac{L}{\text{mean}(T_i, i=1, \dots, m-1)}$
- 5: **if**  $R \geq \frac{C_e}{2}$  **then**
- 6:    $A \leftarrow C_e \left[ 2 - \frac{C_e}{R} \right]$
- 7: **else**
- 8:    $A \leftarrow 0$
- 9: **end if**
- 10:  $p \leftarrow$  packet loss rate in train {Error correction}
- 11: **if**  $p > 0$  **then**
- 12:    $A \leftarrow A \times (1 - p)$
- 13: **end if**

---

where  $C_i$  is the estimation result of packet pair  $i$  and  $C_i = \frac{L}{T_i}$ .

For the second phase of the WBest algorithm (lines 3-13), a packet train of length  $m$  is sent at rate  $C_e$  to estimate available bandwidth. Similar to PGM, a fluid flow model is used to estimate the relationship between available bandwidth and dispersion rate. In a wired network with constant capacity  $C$ , the available bandwidth  $A(t)$  at time  $t$  can be derived by the relationship of  $A(t) = C - S(t)$ , where  $S(t)$  is the amount of cross traffic at time  $t$ . In a wireless network, this relationship holds for the instantaneous capacity  $C(t)$  at time  $t$ :

$$A(t) = C(t) - S(t) \tag{5.4}$$

In practice, the available bandwidth averaged over a short time scale is preferred for most applications. Therefore, for the network with dynamic capacity, such as in wireless networks, we derive the time-based average relationship using the linearity of the expectation from Equation 5.4:

$$E[A(t)] = E[C(t) - S(t)] = E[C(t)] - E[S(t)] \tag{5.5}$$

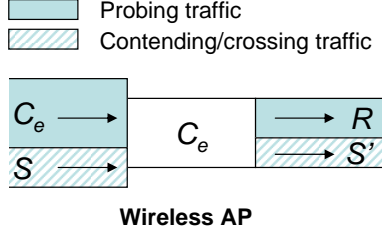


Figure 5.2: A Typical Last Hop Wireless Network.

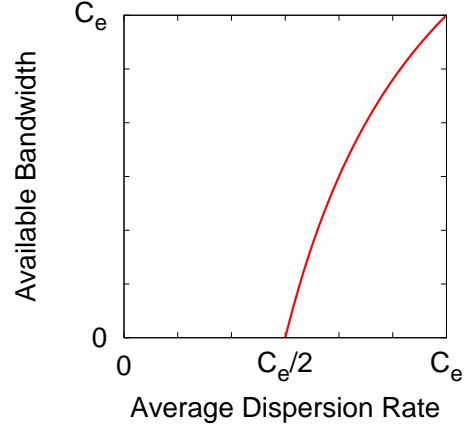


Figure 5.3: Estimating Available Bandwidth using Average Dispersion Rate.

where  $E[A(t)]$ ,  $E[C(t)]$  and  $E[S(t)]$  are the expected available bandwidth, capacity and bandwidth reduction caused by crossing/contending traffic over the estimation period, respectively. Giving the assumption that the wireless network is stationary during the estimation, we have  $E[C(t)] = C_e$  by the definition of effective capacity. Therefore, if we use  $A$  and  $S$  to represent the expectations of available bandwidth and bandwidth reduction caused by crossing/contending traffic during the estimation period, we have:

$$A = C_e - S \quad (5.6)$$

Furthermore, assumption 1 means the arriving rate of probing traffic before the last hop is  $C_e$ . Based on the fair sharing of the FCFS queue, the probing traffic shares the same ratio of the total amount of traffic before and after the AP queue:

$$\frac{C_e}{C_e + S} = \frac{R}{R + S'} = \frac{R}{C_e} \quad (5.7)$$

where as shown in Figure 5.2,  $R$  is the average dispersion rate of the probing traffic at the receiver,  $S'$  is the bandwidth shared by the crossing and contending traffic after competing with the probing traffic and we have  $R + S' = C_e$ . Therefore, we can estimate  $S$ , the expected bandwidth reduction caused by crossing/contending traffic from the dispersion

rate  $R$  using Equation 5.7. Combining Equations 5.6 and 5.7, the available bandwidth is:

$$A = C_e \left(2 - \frac{C_e}{R}\right) = 2C_e - \frac{C_e^2}{R} \quad (5.8)$$

For a wireless network, *achievable throughput* [158] is the average dispersion rate at the receiver for a probing rate of  $C_e$ . Using Equation 5.8, Figure 5.3 shows the relationship between available bandwidth and achievable throughput. Any achievable throughput less than half of  $C_e$  implies zero available bandwidth, and an achievable throughput of  $C_e$  implies an idle wireless network.

In addition to the available bandwidth estimation, WBest also reports the approximate variance of the available bandwidth. Given the available bandwidth Equation 5.8 is twice differentiable and the mean and variance of  $R$  are finite, the variance of available bandwidth can be approximated by the Delta method using second-order Taylor expansions<sup>2</sup>:

$$V[A] \approx V[R] \left[ \left(2C_e - \frac{C_e^2}{R}\right) \right]_{E[R]}^2 = V[R] \left(\frac{C_e}{R}\right)^4 \quad (5.9)$$

Therefore, based on the effective capacity, average and variance dispersion rates, we can estimate the variance and standard deviation of the available bandwidth.

Packet losses on the wireless network and along the network path impact WBest accuracy. Some tools, e.g. *pathload*, discard estimates when packets losses occur to avoid errors in the estimation computation. However, this implies longer measurement times or at least more variance in measurement times. Instead of discarding estimates when packet losses occur, WBest detects packet loss in both packet pairs and packet trains and removes the appropriate pair from the computation. For a packet train, loss rate  $p$  is recorded and the available bandwidth estimate reduced (lines 10- 13 of Algorithm 5.1).

WBest’s major advantages stem from statistically detecting the relative available fraction of the effective capacity in the wireless network instead of using search algorithms to measure the absolute available bandwidth. Most current available bandwidth mechanisms

---

<sup>2</sup><http://en.wikipedia.org/wiki/Variance>

detect absolute available bandwidth by measuring the delay changes in the probing traffic. However, random changes in packet delay due to wireless network conditions make it difficult to determine clear packet delay trends. This reduces the accuracy and increases the convergence time, intrusiveness and instability of the estimation mechanism. By avoiding a search algorithm to determine the probing rate, WBest is designed to converge faster and yield less estimation error. Instead of probing for the absolute rate, WBest estimates available bandwidth using the effective capacity.  $(2 - \frac{C_e}{R})$  in Equation 5.8 is treated as the available fraction of  $C_e$  available to all wireless flows. Derived from the ratio of the effective capacity to the average dispersion rate, the available fraction statistically removes random errors while still capturing the impact of crossing/contending traffic, rate adaptations, and power saving strategies inherent in wireless networks.

### 5.3.3 Number of Packet Pairs and Length of Packet Train

The number of packet pairs in the first step of WBest and the number of packets in the packet train in the second step play important roles in the accuracy, convergence time and intrusiveness of the algorithm. Generally, more packet pairs and longer packet trains improve accuracy at the cost of higher convergence time and more intrusiveness.

WBest seeks to minimize convergence time and intrusiveness at a given accuracy level. The confidence interval (I) and the modeled variance  $\sigma$  [158] can be used to estimate the minimum required number of packet pairs using:

$$n = \frac{Z^2 \sigma^2}{I^2} \quad (5.10)$$

where  $Z$  is a constant determined by the confidence level. For example, assume a streaming session wants to bound the effective capacity estimate within 500 Kbps to match the granularity of encoded scaling levels for the multimedia stream. To bound the effective capacity estimate within 500 Kbps with 95% confidence, Equation 5.10 indicates at least 6 ([5.34]) samples are needed. This is based on  $\sigma = 0.59$  Mbps for an 11 Mbps wireless channel rate and a packet size of 1500 bytes with  $Z = 1.96$  and  $I = 500$  Kbps [158].

Similarly, the number of packets  $m$  in the packet train can also be computed. With the same available bandwidth estimation bounds and given a modeled maximum  $\sigma = 1.38$  Mbps [158] for an 11 Mbps channel link rate and packet size of 1500 bytes with contending traffic,  $Z = 1.96$  and  $I = 500$  Kbps, the minimum train size  $m$  is 30 ( $\lceil 29.26 \rceil$ ). As real network conditions may change unexpectedly, Equation 5.10 only provides an approximation on the sample sizes needed.

The number of packets in a train also impacts the time scale and sensitivity of available bandwidth estimations. In general, the estimation of available bandwidth represents the average estimation during the measurement period [92]. As a major part of the convergence time, the time  $T_m$  spent to estimate available bandwidth depends on the number of packets  $m$  in the train.  $T_m$  can be approximated using  $m$  and packet size  $L$  as  $T_m = m * L / C_e$ . Furthermore, the probability crossing traffic gets included in the bandwidth estimation is related to the length of the train. Assume CBR crossing traffic is sent at rate  $S$  with at least one packet caught by the packet train:

$$\begin{aligned}
 S * T_m / L &\geq 1 \\
 S &\geq L / T_m = C_e / m
 \end{aligned}
 \tag{5.11}$$

The sensitivity of the available bandwidth estimation can be defined based on the number of packets in the train, which has a negative relationship with the train length. For instance, to catch crossing traffic sent at rate  $C_e/10$ , a packet train with at least 10 packets is needed.

Selecting the number of packet pairs and train length is complicated in practice because the bottleneck queue size also limits the number of packet pairs and the length of the packet train. The *pathrate* queue size probing method [110] can be used to detect buffer limitations along the flow path. However, this probing method increases the intrusiveness and measurement time and is not appropriate for many applications. Since the WBest packet train sending rate is set to the effective capacity of the wireless Access Point (AP),

the probability of queue overflow in the network is determined by the queue size at the last hop wireless AP. Previous research [167] indicates that current wireless AP queue lengths range from 40 to 300 packets. Thus, WBest simply limits the packet train to less than 40 packets. To further avoid queue overflow due to packet pairs, WBest inserts a 10 millisecond gap between pairs to reduce the packet pair probing rate during capacity estimation.

#### 5.3.4 Impact of Errors in Effective Capacity Estimation

The effective capacity estimate in the first step of WBest impacts the available bandwidth estimate in the second step. If  $C'_e$  denotes the estimated effective capacity from the first step and  $C_e$  is the actual effective capacity, the fluid flow model from Equation 5.7 yields:

$$\frac{C'_e}{C'_e + C_e - A} = \frac{R}{C_e} \quad (5.12)$$

By defining the error ratio  $Y$  as  $C'_e = C_e(1 + Y)$ , the dispersion rate is:

$$R = \frac{C'_e C_e}{C'_e + C_e - A} = \frac{(1 + Y)C_e^2}{(2 + Y)C_e - A} \quad (5.13)$$

Mimicking the derivation from Equation 5.8, the estimated available bandwidth,  $A'$ , is  $A' = 2C'_e - \frac{C_e^2}{R}$ . The relationship between estimated available bandwidth  $A'$  and real available bandwidth,  $A$ , then becomes:

$$A' = A(1 + Y) - C_e Y(1 + Y) \quad (5.14)$$

To study available bandwidth estimation errors due to error in effective capacity estimation, relative error,  $E$ , is defined as:

$$E = \frac{A' - A}{A} \quad (5.15)$$

Positive and negative values for relative error  $E$  denote over-estimation and under-estimation

of the available bandwidth, respectively. From Equation 5.15, the relative error in available bandwidth estimation is  $E = Y - Y(1 + Y)C_e/A$ . Figure 5.4 shows in fractional form the relative error of estimated available bandwidth and the relative error in capacity estimation for three distinct cases. Effective capacity estimations that are too high always under-estimate the available bandwidth. Effective capacity estimations that are too low can result in either an over-estimate or under-estimate of the available bandwidth, depending upon on the actual available bandwidth in the network. Errors in effective capacity estimation can be bounded by modeling [158] or by measurement, e.g., using the range of the results from the capacity estimation step to approximate the relative error on the estimation of available bandwidth. Moreover, for applications where a conservative estimate of the available bandwidth is desirable, such as in multimedia streaming, a higher effective capacity estimator can be used (e.g., using the top 10% in Equation 5.3 for  $C_e$  instead of the median) to minimize the potential performance degradation caused by over-estimating the available bandwidth of the underlying network.

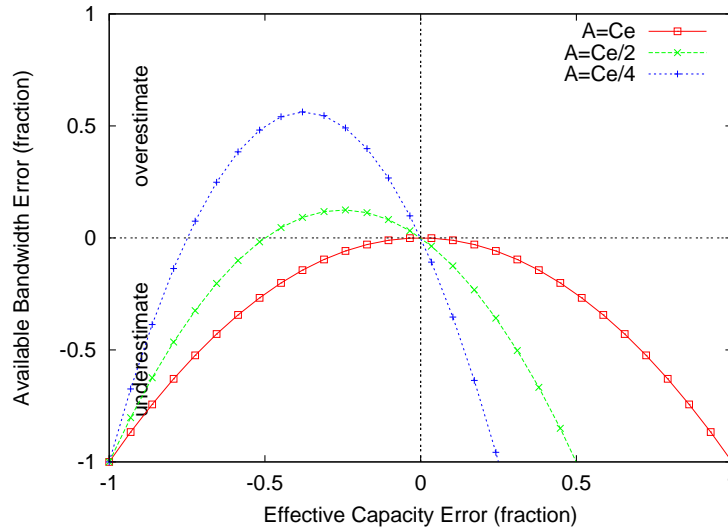


Figure 5.4: Relative Error Caused by Effective Capacity Estimation Errors.

### 5.3.5 Pre-dispersion and Pre-compression

Although it is assumed that the last mile is the bottleneck of the network path defined in Equation 5.1, the packet train may still arrive at the AP with a lower rate or higher rate than  $C_e$ . We call this rate  $R_p$  the pre-dispersion or pre-compression rate because it happens before the packet train arrives at the last hop wireless network.

The possible sources of a pre-dispersion could be a link with an available bandwidth  $A_i$  less than the packet train rate  $C_e$ , which is also the effective capacity of the last hop wireless network. Therefore, to analyze the impact caused by the pre-dispersion and pre-compression behaviors, we can use the same fluid model as Equation 5.7 if the pre-dispersion/pre-compression rate  $R_p$  is greater than or equal to the available bandwidth  $A$ :

$$\frac{R_p}{R_p + C_e - A} = \frac{R}{C_e} \quad (R_p \geq A) \quad (5.16)$$

In the case of pre-dispersion rate,  $R_p$  is less than the available bandwidth, the probing traffic will not be further dispersed at the wireless hop. Therefore, Equation 5.7 is not applicable and we have  $R = R_p$ . We define  $X$  as the ratio of pre-dispersion/pre-compression on the probing traffic, such that we have  $R_p = C_e(1 + X)$ , where a positive  $X$  denotes a pre-compression and a negative  $X$  denotes a pre-dispersion. Following the same derivation we can get the dispersion rate after passing the last hop as:

$$R = \begin{cases} \frac{C_e R_p}{R_p + C_e - A} = \frac{C_e^2(1+X)}{(2+X)C_e - A} & (R_p \geq A) \\ C_e(1 + X) & (R_p < A) \end{cases} \quad (5.17)$$

As described in Equation 5.8, the estimated available bandwidth with pre-dispersion or pre-compression is defined as  $A' = 2C_e - \frac{C_e^2}{R}$ . Thus, by representing  $R$  using  $C_e$ ,  $A$  and  $X$ , we can derive the relation between estimated available bandwidth  $A'$  and real available bandwidth  $A$  from Equation 5.18 as:

$$A' = \begin{cases} \frac{A}{1+X} - \frac{XC_e}{1+X} & (R_p \geq A) \\ 2C_e - \frac{C_e}{1+X} & (R_p < A) \end{cases} \quad (5.18)$$

To study the errors caused by pre-dispersion and pre-compression, we compute the relative error  $E$  between the estimated available bandwidth and the real available bandwidth using Equation 5.15. Therefore, a positive and negative relative error  $E$  denote a over-estimation and under-estimation of the available bandwidth, respectively. The relative error in available bandwidth caused by pre-dispersion and pre-compression can be derived as Equation 5.19:

$$E = \begin{cases} \frac{X(C_e/A-1)}{1+X} & (R_p \geq A) \\ \frac{(1+2X)C_e/A}{1+X} - 1 & (R_p < A) \end{cases} \quad (5.19)$$

Figure 5.5 shows the theoretical relationship between the pre-dispersion and pre-compression ratio  $X$  and the relative error in available bandwidth  $E$  for the network with different amount of available bandwidth.

It clearly shows that pre-dispersion results in a lower estimated available bandwidth than the real available bandwidth. On the contrary, pre-compression results in a higher estimated available bandwidth. Moreover, as the available bandwidth decreases, the impact caused by pre-dispersion and pre-compression increases. In addition to the theoretical relationship, if the pre-dispersion reduces the packet train probing rate lower than the available bandwidth, there will be no further dispersion at the last hop. Therefore, Equation 5.16 cannot be used to compute the dispersion rate  $R$ . Instead, we have  $R = R_p$  and the relative error of available bandwidth can be computed based on Equation 5.8 and 5.15. The “No dispersion” curve depicts the converting point of the relative errors when the pre-dispersion rate is lower than the available bandwidth.

For streaming applications, the underestimation of the available bandwidth caused by pre-dispersion makes the streaming rate selection more conservative, which is helpful for avoiding performance problems such as bursty lost and rebuffer events. The overestimation

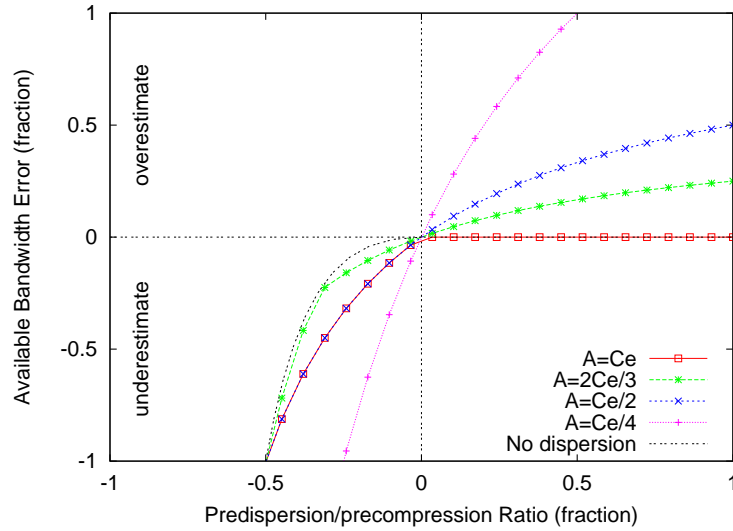


Figure 5.5: Relative Error Caused by Pre-dispersion/compression

caused by pre-compression impacts the streaming performance. A possible solution is to increase the estimation samples, thus reducing the errors caused by pre-compression. Be aware that the errors shown in Figure 5.5 represent the worst case such that all samples in the estimation are pre-dispersed/pre-compressed for the given ratio. We expect a lower relative error in practice because the WBest algorithm is based on an average of multiple samples.

### 5.3.6 Error Detection

Packet loss observed at the WBest receiver may be attributed to either wireless losses or congestion losses (queue overflow). The WBest error correction adjusts for wireless losses. However, while WBest controls the probing traffic sending rate to avoid queue overflow, large amounts of crossing traffic and contending traffic may still produce queue losses that can cause an over-estimate of available bandwidth. In most cases, one can assume that any queuing loss is due to a saturated wireless link with no available bandwidth. However, to guard against queue overflow at an upstream router, Loss Discrimination Algorithms (LDA), such as [169, 170] could be added to WBest to distinguish congestion loss or wireless loss.

Another potential source of estimation error comes from last hop probe packet compression. System factors, such as high CPU load at the wireless clients and user-level timestamps [110] may cause two or more packets to have very close arrival timestamps. This last hop compression can result in recorded arrival rates that are higher than the effective capacity. For example, our measurements show the minimum timestamp from the user level timer is about 2.3  $\mu$ s. This results in a dispersion rate over 5000 Mbps for a probe packet size of 1500 bytes. Thus, to reduce the error due to last hop compression, if the received timestamp yields a higher rate than the actual sending rate, WBest uses the actual sending rate instead of the dispersion rate to compute available bandwidth.

## 5.4 Experiments

### 5.4.1 Experiment Design

WBest is implemented<sup>3</sup> in Linux and evaluated by varying network conditions in an IEEE 802.11 wireless testbed. As shown in Figure 5.1, the wireless testbed consists of an application server that performs the estimation (wbserver), a traffic server (tgserver), a wireless AP and three clients (Client A, B and C). The AP in the testbed is a Cisco Air-AP1121G<sup>4</sup> with IEEE 802.11b/g mode. Both servers are PCs with P4 3.0 GHz CPUs and 512 MBytes RAM and the three clients are PCs with P4 2.8 GHz CPUs with 512 MBytes RAM. All the testbed PCs run SUSE<sup>5</sup> 9.3 Linux with kernel version 2.6.11. The servers connect to the AP with a wired 100 Mbps LAN, and the clients connect to the AP with IEEE 802.11b/g WLAN using Allnet<sup>6</sup> ALL0271 54 Mbps wireless PCI card with a prism GT chipset.<sup>7</sup>

Even though there are some recent wireless bandwidth estimation tools being proposed including [90, 127, 159, 160, 162], most of them cannot be included in a direct comparison

---

<sup>3</sup>WBest source code can be download from <http://perform.wpi.edu/tools>

<sup>4</sup><http://www.cisco.com/en/US/products/hw/wireless/ps4570/index.html>

<sup>5</sup><http://www.novell.com/linux/>

<sup>6</sup><http://www.allnet-usa.com/>

<sup>7</sup><http://www.conexant.com/products/entry.jsp?id=885>

with WBest. For example, some tools estimate only capacity [159, 160] or require third party capacity estimation [90]. Thus these tools are not directly comparable to WBest. Additionally, many tools are provided only via simulation [127, 162] or do not have source code available [161]<sup>8</sup>. Hence, these tools are not able to be evaluated through experiments. Therefore, for performance comparison, three popular and published independent tools that can estimate available bandwidth were selected: *IGI/PTR* v2.0 [117], *pathChirp* v2.4.1 [116] and *pathload* v1.3.2 [113].

For the experimental runs, the four tools are run sequentially to estimate the downstream available bandwidth from wbestserver to client A. While all the tools were setup using their default configuration, to provide a fair performance comparison, the following methodology was used to run and summarize the estimation results. Although *IGI/PTR* converges with two results, the PTR results are used as the author suggests. Since *pathload* converges with a range of available bandwidths, the median of the range is used for comparison. During the evaluation, some *pathload* runs never converge under particular wireless channel conditions. These runs were halted if they fail to converge in 100 seconds which is the upper limit of normal convergence time for *pathload*. Since *pathChirp* is designed as a continuous monitoring tool without an explicit convergence policy, convergence follows the author’s method described in [116]. In this method, the difference between the 90th and 10th percentiles of the estimations are computed and convergence is defined when the difference is less than 1/5 of the available bandwidth<sup>9</sup> (approximately 6 Mbps in our testbed).

As shown in Figure 5.6, each evaluation consists of back-to-back runs employing four bandwidth estimation tools and one downstream CBR flow, which is used to approximate the actual available bandwidth of the wireless network as discussed in Section 5.4.2. For all cases with crossing or contending traffic, the estimations start five seconds after the background traffic starts to let the system stabilize. Similarly, there is a five second delay between the end of one tool and the start of the next to allow background traffic to stabilize.

---

<sup>8</sup>Lately, DietTOPP source code is released after the evaluation of WBest.

<sup>9</sup>This ratio is computed from the evaluation setup in [116].

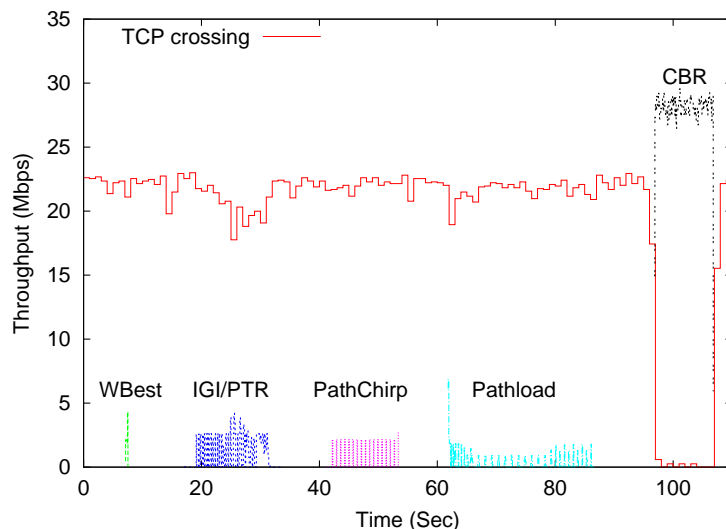


Figure 5.6: An Example of a Sequential Run of Bandwidth Estimation Tools.

Table 5.1 itemizes the fifteen experimental cases. The base configuration, case 0, has no contending or crossing traffic and no induced changes in the wireless network conditions. Cases 1-12 include a variety of crossing and/or contending traffic situations provided by UDP and TCP traffic generators residing on client B, client C and tgserver. The Multi-Generator Toolset (mgen) v4.2b6<sup>10</sup> and iperf v2.0.2<sup>11</sup> are used to generate UDP and TCP traffic, respectively. For case 13, wireless rate adaptation is induced by removing the antenna of a wireless client and reducing the wireless AP’s sending power and receiving antenna gain. With a client received signal strength indicator (RSSI) between -70 dbm and -74 dbm, the rate adaptation ranged from 1 to 48 Mbps. Figure 5.7 shows the actual rate adaptation measured with a wireless sniffer<sup>12</sup>. This rate adaptation case results in 8% of wireless layer retries for both the AP and the client. By enabling the client wireless network card power save mode (PSM) option, Case 14 provides an experiment to evaluate bandwidth estimation when PSM is used. The AP’s Data Beacon Rate (DTIM) is set to the default value of 2 such that a delivery traffic indication message is included in every other beacon to notify the PSM clients of waiting packets at the AP. The PSM option on

<sup>10</sup><http://pf.itd.nrl.navy.mil/mgen/>

<sup>11</sup><http://dast.nlanr.net/Projects/Iperf/>

<sup>12</sup><http://perform.wpi.edu/tools/>

the cards is disabled for all the other experimental cases.

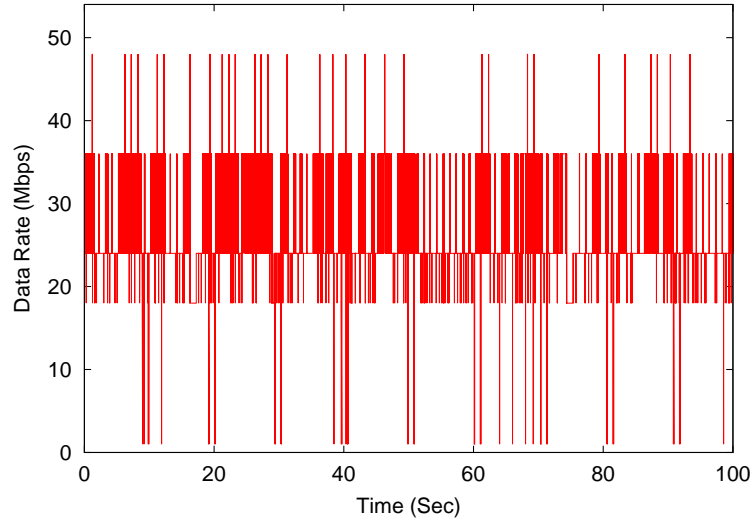


Figure 5.7: Rate Adaptation Behavior

Each of the fifteen cases were repeated 30 times with the median and quartiles reported for all runs. To ensure comparability across different runs, the RSSI range for all wireless clients is between -38 dbm and -42 dbm, and all clients were shown to have the same maximum throughput of about 29 Mbps. To mitigate interference from co-existing campus wireless networks, all experiments are run in our wireless streaming multimedia lab<sup>13</sup> which was painted with an additive<sup>14</sup> to reduce the radio transmissions going through the walls. Furthermore, all the experiments were conducted after midnight during the WPI summer break such that most of the campus wireless network was in an idle state.

The relationship between relative error and the number of pairs in step 1 of the WBest algorithm (estimate effective capacity) was explored by using Equation 5.15 to compute the error of the estimated effective capacity using different numbers of packet pairs and defining real effective capacity as the median of the 90 packet pair run. Figure 5.8 shows the relationship between the effective capacity error and the number of packet pairs sent for four typical wireless cases: idle, crossing traffic, contending traffic, and rate adaptation. As the number of packet pairs sent increases, the error decreases. Rate adaptation requires

<sup>13</sup><http://perform.wpi.edu/wsml/>

<sup>14</sup><http://www.forcefieldwireless.com/defendairadditive.html>

Table 5.1: Evaluation Cases for Experiments.

Case	Crossing Traffic	Contending Traffic
0	None	None
1	Client B: UDP 4.6 Mbps	None
2	None	Client B: UDP 4.6 Mbps
3	Client B: TCP	None
4	None	Client B: TCP
5	Client B: UDP 2.3 Mbps Client C: UDP 2.3 Mbps	None
6	None	Client B: UDP 2.3 Mbps Client C: UDP 2.3 Mbps
7	Client B: TCP Client C: TCP	None
8	None	Client B: TCP Client C: TCP
9	Client B: UDP 2.3 Mbps	Client C: UDP 2.3 Mbps
10	Client B: TCP	Client C: TCP
11	Client B: UDP 2.3 Mbps	Client C: TCP
12	Client B: TCP	Client C: UDP 2.3 Mbps
13	Case 0 with rate adaptation	
14	Case 0 with PSM enabled	

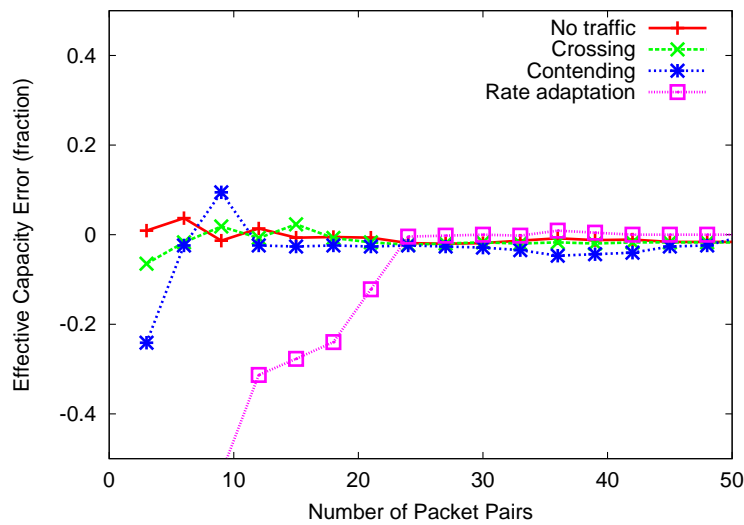


Figure 5.8: Analysis of Number of Packet Pairs

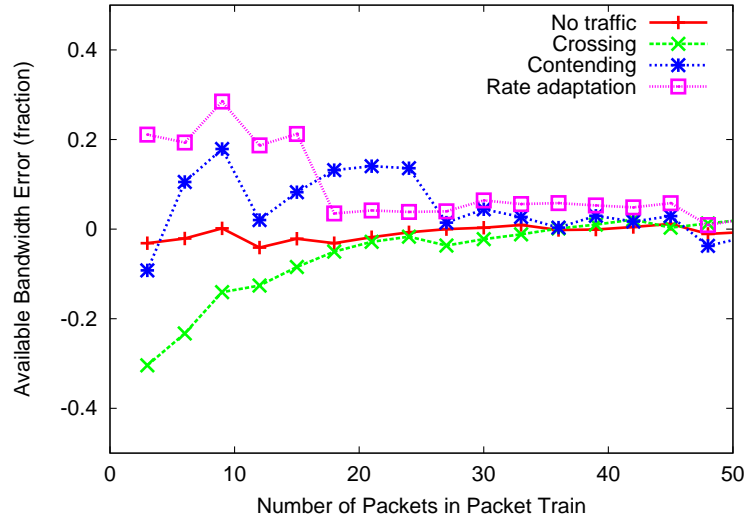


Figure 5.9: Analysis of Length of Packet Train

the highest number of packet pairs to produce reasonably accurate measurements. To provide accuracy for all these cases while reducing the impact on the available bandwidth estimations, 30 packet pairs were used in all the WBest evaluations. Similarly, based on Figure 5.9, 30 was chosen as the length of the packet train for step 2 of the WBest algorithm (estimate available bandwidth) for all the WBest experiments.

In addition to these fifteen evaluation cases described earlier, we also design evaluations to discover the impact of packet sizes of both WBest probing traffic and crossing/contending traffic. The packet size tests include two parts. First, a series of WBest sessions with different probing packet sizes under the idle channel (case 0) are executed to discover the relationship between WBest estimation and the CBR throughput with the same packet sizes. Next, a series of WBest sessions with the default packet size (1460 bytes) and different packet size crossing/contending traffic (case 0 and 1) are executed to discover the impact of packet sizes of crossing/contending traffic. For these packet size evaluations, we vary the packet sizes of crossing/contending traffic from 300 bytes to 1460 bytes in a step of 300 bytes. To ensure comparability across different packet sizes, we change the packet rate for each packet size to keep an equal amount of contending/crossing traffic of 4.6 Mbps. That is, for small packet sizes, the packet sending rate is increased to achieve the same

amount of traffic in term of bits per second. CBR throughput with the same packet size as the probing traffic is measured to compare with the estimation results. Similar to these fifteen cases, the test for each packet size are repeat 30 times and the median and quartiles reported for all runs.

### 5.4.2 Metrics for Performance Evaluation

We use three performance metrics, accuracy, convergence time and intrusiveness to evaluate the bandwidth estimation tools in wireless networks.

To evaluate estimation accuracy, the true available bandwidth of the wireless network under different configurations is needed – referred to here as the *ground truth*. In wired networks, the ground truth can be obtained from the network setup or measurement on the router. However, in wireless networks, it is difficult to get the actual ground truth due to the dynamic wireless network conditions. First, wireless network capacity is not constant due to the impact of rate adaptation, thus the available bandwidth cannot be simply computed as in wired networks by subtracting the cross traffic from the capacity. Moreover, the high and volatile overhead in wireless networks makes it difficult to infer the available bandwidth from the amount of crossing and contending traffic. This is because the overhead varies due to both the amount of contending traffic and the wireless conditions. For example, the same amount of contending traffic could result in a higher bandwidth reduction in a fading channel than in an ideal channel. Finally, even though the wireless AP could report its own channel utilization, this utilization information is insufficient to infer the available bandwidth of the wireless network. For instance, the available bandwidth in a wireless network is also impacted by the clients' environment, such as the co-channel interference from other wireless nodes that are out of the interference range of the wireless AP or by bit errors in frame transmissions. Therefore, it is difficult to obtain the true available bandwidth directly from network setup or by measuring utilization on the AP in wireless networks.

In our evaluation, the ground truth of the available bandwidth is approximated by the

downstream throughput of a single saturated CBR UDP flow with a packet size of the Maximum Transmission Unit (MTU) for each case tested. However, using CBR to approximate ground truth may include errors for some cases. For the ideal case and the cases with inelastic UDP crossing traffic, the CBR throughput provides a close approximation to the ground truth of the available bandwidth because there is no contending overhead. For the cases with UDP crossing and contending traffic in a non-saturated wireless network, the approximated ground truth could be slightly higher than the true available bandwidth because the CBR traffic may actively take bandwidth away from the contending traffic. However, the CBR throughput can still be considered as a close approximation of the ground truth for these cases because these errors caused by contending are relatively small. When there are multiple UDP contending sources that saturate the wireless network, the heavy contending effect could trigger rate adaptation and the CBR throughput could be lower than the true available bandwidth. Therefore the CBR throughput does not accurately represent the ground truth, and we mark the ground truth of these cases as unknown, such as case 6 where the CBR traffic is contending with two other UDP traffic sources. The setup of case 6 is analyzed in Section 5.4.3 in detail. Finally, for cases with TCP crossing and contending traffic, the CBR throughput does not represent the ground truth because TCP traffic reduces its sending rate in response to packet losses and delay due to the queuing and contending in wireless networks. Since the nature of TCP bulk transfer is to expand to use all available bandwidth in the network, we approximate the ground truth for these cases with TCP crossing/contending traffic as zero. In summary, the accuracy is evaluated by comparing the estimated available bandwidth to the ground truth, which is approximated by CBR throughput (except case 6, which is marked as unknown), or zero for the cases with TCP crossing/contending traffic.

Convergence time is the total time used by the tool to complete an estimation. IGI/PTR, pathload and WBest report their convergence times upon completion. The convergence time of pathChirp is measured based on the definition of convergence as discussed in Section 5.4.1. Even though WBest is not a converging-based tool, we use convergence time to

denote the total estimation time used to complete one WBest estimation.

Intrusiveness can be measured by the number of bytes or by the data rate relative to the available bandwidth. The measurement of number of bytes provides the total intrusiveness during the estimation sessions, while the measurement of data rate provides the relative bursty intrusiveness of the tools. Streaming applications are more sensitive to the total intrusiveness than the bursty intrusiveness because of the usage of a playout buffer, which can smooth bursty intrusiveness. Therefore, to gauge the applicability of the bandwidth estimation tools for streaming applications in wireless networks, we use the total number of bytes sent into the network as the measure of intrusiveness.

### 5.4.3 Discussion on Experiment Setup

This section provides additional information about evaluation case 6, 13 and 14. As discussed in Section 5.5, case 6 with 2 UDP contending traffic sources experiences the impact of rate adaptation. While it depends upon the implementation of the rate adaptation algorithm, it is a normal behavior that wireless connections reduce the data rate upon multiple transmission failures, which could be caused by either low signal strength, high BER or contention. The rate adaptation triggered by contending effects may reduce the wireless network performance, which is also demonstrated in research [156]. To describe the rate adaptation behaviors of the clients and AP, we collect and analyze data from the wireless AP log and the packets captured on the clients. Figure 5.10 and Figure 5.11 show that both the clients and AP are involved in rate adaptation in a typical run of case 6. The AP log also denotes that the AP has a high MAC layer retry rate of more than 20%. The data rate of the retried frames are shown in Figure 5.12, which denotes that the MAC layer retries involved in rate adaptation. Typical throughput measured at the clients also confirm the impact due to rate adaptation as shown in Figure 5.13. As shown in Figure 5.13, the throughput of the UDP crossing traffic decreases because of the rate adaptation and MAC layer retries caused by contending effects from the two UDP contending traffic, starting at 80 seconds for this session.

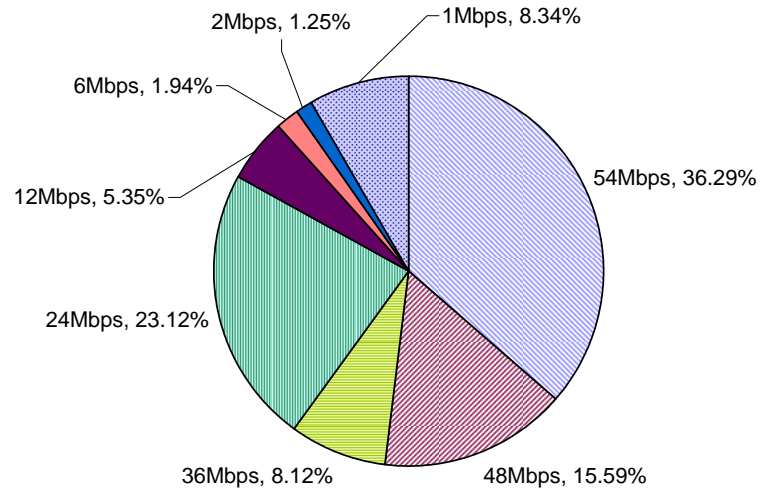


Figure 5.10: Typical Clients' Data Rate for Case 6 (Two Contending UDP Traffic)

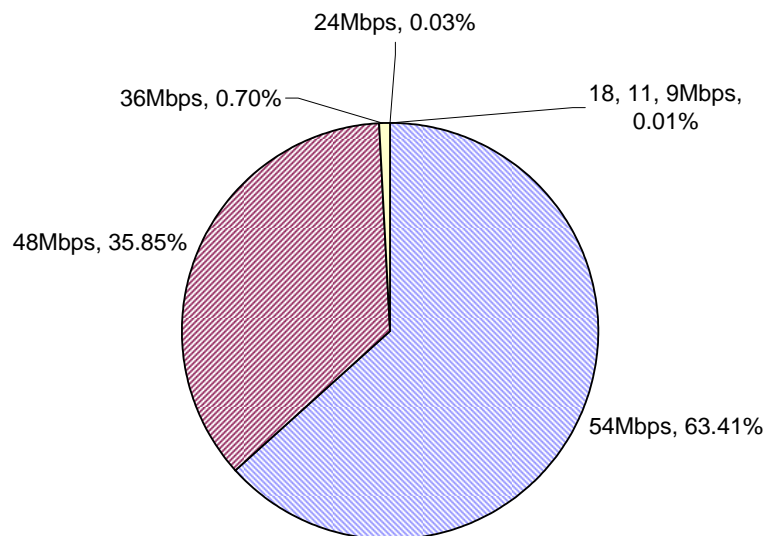


Figure 5.11: Typical AP's Data Rate for Case 6 (Two Contending UDP Traffic)

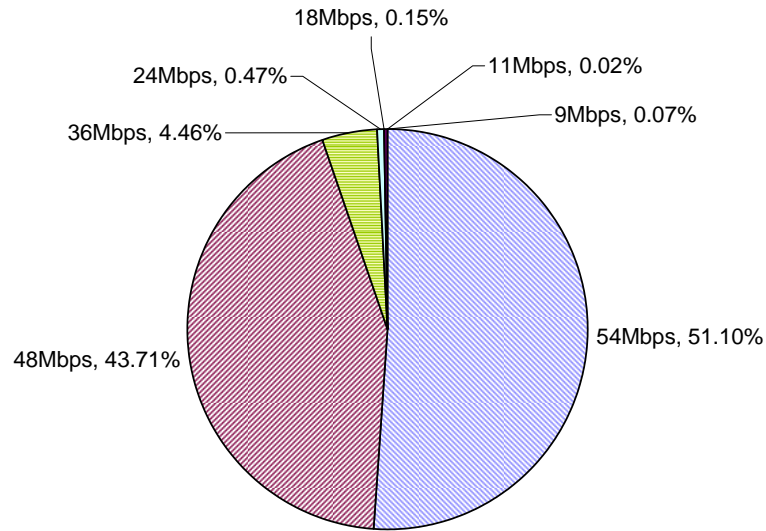


Figure 5.12: Typical AP Retries' Data Rate for Case 6 (Two Contending UDP Traffic)

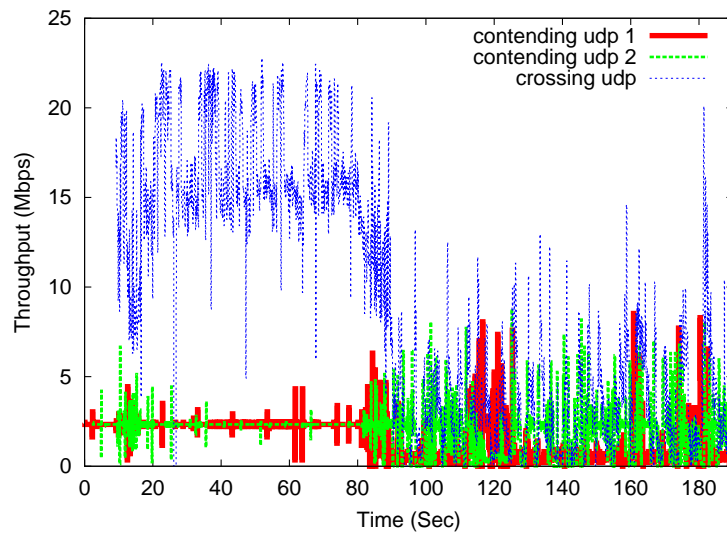


Figure 5.13: Typical Downstream UDP Throughput for Case 6 (Two UDP Contending Traffic)

Evaluation case 13 is designed to test WBest under rate adaptation conditions. As discussed in Section 5.4, the rate adaptation is observed by a wireless sniffer. However, to show the impact of rate adaptation on packet delay, which could impact accuracy of delay-based bandwidth estimation, we show the RTT measured by ping with 64 byte and 1460 byte packets in Figure 5.14. Fewer than 10% of 64 byte packets and 20% for 1460 byte packets have a large RTT in good wireless conditions. While for the bad conditions, in which the data rate is adapted to the channel condition, more than 30% of 64 byte packets and 40% of the 1460 byte packets have a large RTT. The delay changes under the rate adaptation condition could potentially reduce the accuracy of all bandwidth estimation tools. For example, the packet delay or dispersion may unexpectedly increase due to the rate adaptation, thus impacting the convergence decision of the delay or dispersion based tools. Moreover, as discussed in Sections 2.2 and 4.2, the rate adaptation behavior varies due to not only the vendor-specific adaptation algorithms, but also due to the environmental conditions, such as the room composition or indoor/outdoor setups. This could further increase the uncertainties for these convergence algorithms using delay or rate measurements. More estimation samples can reduce the impact of rate adaptation and results in a long term average of the available bandwidth, such as the large number of probing packets being used in pathload. WBest does not use the increases in delay to converge to available bandwidth. Instead, it statistically captures the average dispersion changes between the effective capacity and achievable throughput during the estimation. Therefore WBest reflects the available bandwidth during the measured period and suffers less from the impact of rate adaptations in wireless networks than do other bandwidth estimation techniques that converge. However, WBest may result in large variances between estimations due to the relative short sampling period and small number of samples.

Evaluation case 14 is designed to test WBest under power saving mode (PSM). The delay in PSM is impacted by both the AP queuing delay and the sleeping delay, which is the time that the packet is being buffered at the AP while the client is in sleeping mode. Figure 5.15 shows the RTT and loss rate changes as the amount traffic load increases. As

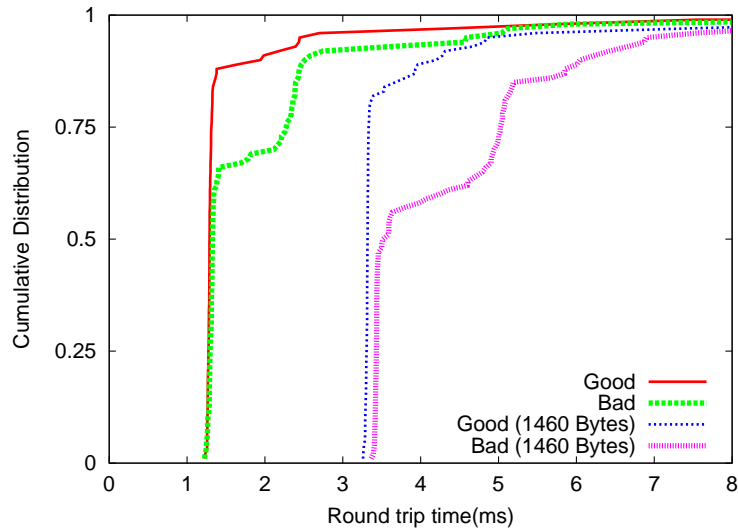


Figure 5.14: Ping Result for Case 13 (Wireless Rate Adaptation)

the amount of crossing traffic increases, the RTT is reduced because the more traffic in the network, the longer the client stays awake. This results in a reduced delay caused by the sleep mode. However, as the crossing traffic load increases to close to the effective capacity, the AP queue fills up, which yields a higher packet loss rate and an increased queuing delay. This shows that for PSM mode, a higher crossing traffic is not necessary to result in a higher delay. Therefore, PSM mode may potentially impact the bandwidth estimation tools based on the characterization of an increase in delay, such as pathload and PathChirp. WBest does not rely on the increase in delay to converge, and it always probes at the effective capacity to keep the client in awake mode. Therefore, WBest is not impacted by the PSM mode.

#### 5.4.4 Accurate Sending Rate Control in WBest

We implemented WBest in Linux and evaluated it in our IEEE 802.11 wireless testbed. The Linux system provides timers with millisecond (*sleep*) and microsecond (*usleep* and *select*) resolution timers. However, these timers may not satisfy the required resolution to control accurate sending rates. Therefore, we implement a busy-waiting timer using *gettimeofday* to provide a microsecond resolution timer for cases requiring a high sending rate.

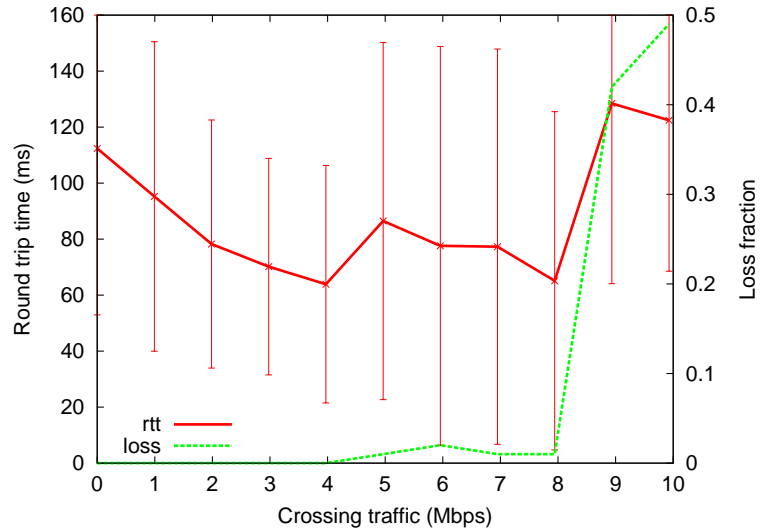


Figure 5.15: Ping Result for Case 14 (PSM Case)

Even though the busy-waiting method may increase the CPU usage of the server during measurement, the impact caused by this short measurement duration is not significant, especially when the sending hosts of WBest are usually on high performance, multiprocessor servers. The microsecond resolution timer together with the *select* functions provide a reasonable sending rate control for WBest. Figure 5.16 shows an evaluation of the sending rate control mechanism of WBest. Each data point depicts the average rate of 100 packets with packet sizes of 1460 bytes sending from wbestserver, which is a Pentium 4 3.0 GHz computer with 512 Mbytes RAM. The mean sending rate and the confidence interval show that the rate control mechanism works as expected in that the real sending rate equals the requested sending rate with a small confidence interval. Also, the CPU usage does not have a noticeable increase when WBest is sending at the rate of 35 Mbps on wbestserver, where 35 Mbps is about the maximum effective throughput of IEEE 802.11g working at 54 Mbps link data rate with a packet size of 1460 Bytes.

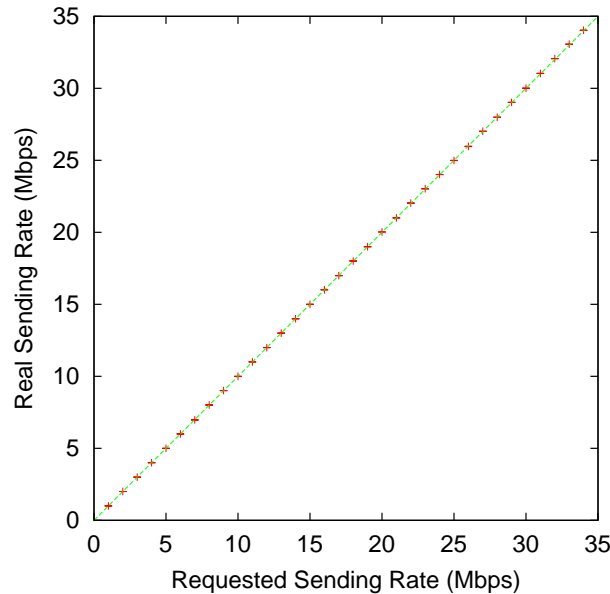


Figure 5.16: Evaluation of Sending Rate Control Mechanism in WBest

## 5.5 Analysis

### 5.5.1 Data Collected

For each of the fifteen test cases, Table 5.2 gives the median estimated available bandwidth for 30 evaluations runs of each of the four bandwidth estimation tools. The ‘ground truth’ column provides the true available bandwidth, approximated from the measured CBR UDP throughput with a packet size of 1500 bytes or set to zero if the specific test case includes a TCP bulk transfer as described in Section 5.4.

For Case 6, as discussed in Section 5.4.3, the UDP traffic from the two contending clients causes the AP and the clients to use rate adaptation even with good RSSI values. While it is normal for rate adaptation to be triggered by high contention for the wireless channel, the saturated CBR throughput of 9.29 MBps for case 6 does not represent ground truth because higher throughput can be obtained with a lower offered CBR rate, as could be the case with the bandwidth estimation tools. Thus, for case 6 the ground truth is marked as unknown. In general, for all other cases in Table 5.2, WBest provides the most accurate estimation of the available bandwidth compared to the other three bandwidth

estimation techniques.

In addition to the accuracy, the intrusiveness and convergence time is recorded for each test case. The intrusiveness is defined as the total bytes sent by each tool during estimation and the convergence time is the time spent by each tool to arrive at a final bandwidth estimation result in each estimation. Table 5.3 and Table 5.4 provide the median values for intrusiveness and convergence times over 30 runs for all fifteen test cases, respectively. WBest yields the lowest intrusiveness and convergence time in every case.

Table 5.2: Estimated Available Bandwidth (Median, in Mbps).

Case: Remark	IGI/PTR	PathChirp	Pathload	WBest	Ground truth
0: Idle channel	8.11	30.15	6.78	28.47	28.94
1: UDP crossing	8.74	28.89	6.81	23.24	24.39
2: UDP contending	10.06	27.59	6.91	15.76	20.52
3: TCP traffic	1.92	5.00	1.95	1.01	0.00
4: TCP traffic	1.12	14.50	1.69	0.00	0.00
5: UDP crossing	9.99	26.91	7.07	22.87	24.50
6: UDP contending	9.62	26.98	6.78	14.56	-
7: TCP traffic	1.48	5.00	1.10	0.00	0.00
8: TCP traffic	0.66	11.97	0.92	0.00	0.00
9: Multiple TCP/UDP	6.89	25.60	6.47	13.26	16.26
10: Multiple TCP/UDP	0.67	5.72	0.99	0.00	0.00
11: Multiple TCP/UDP	0.59	9.95	0.48	0.00	0.00
12: Multiple TCP/UDP	0.77	12.73	1.06	0.00	0.00
13: Rate Adaptation	5.18	16.79	5.99	13.99	15.26
14: PSM	3.62	10.82	0.87	8.36	8.19

The detailed results of each experiment cases are presented as box-and-whisker plots<sup>15</sup> as shown in Figure 5.17 to Figure 5.31. However, detailed analyses are based on the most representative categorization of these cases, namely: idle channel (case 0), UDP crossing traffic (cases 1 and 5), UDP contending traffic (cases 2 and 6), TCP crossing/contending traffic (cases 3, 4, 7, and 8), multiple crossing/contending sources and mixed protocols (cases 9 to 12), rate adaptation (case 13), and power saving mode (case 14). In addition

<sup>15</sup>In a box-and-whisker plot, the ends of the box are the upper and lower quartiles, the horizontal line inside the box is the median and the two lines (whiskers) outside the box extend to the 10 and 90%-tile of the observations.

Table 5.3: Intrusiveness (Median, in MBytes)

Case: Remark	IGI/PTR	PathChirp	Pathload	WBest
0: Idle channel	0.56	0.45	1.18	0.13
1: UDP crossing	0.56	0.45	1.55	0.13
2: UDP contending	0.47	0.45	1.53	0.13
3: TCP traffic	2.54	0.46	1.22	0.13
4: TCP traffic	1.51	0.45	0.86	0.13
5: UDP crossing	0.56	0.45	1.67	0.13
6: UDP contending	0.47	0.45	1.66	0.13
7: TCP traffic	3.11	0.46	0.95	0.13
8: TCP traffic	1.98	0.46	0.98	0.13
9: Multiple TCP/UDP	0.66	0.45	1.57	0.13
10: Multiple TCP/UDP	2.17	0.46	1.24	0.13
11: Multiple TCP/UDP	1.79	0.49	0.53	0.13
12: Multiple TCP/UDP	2.17	0.46	1.46	0.13
13: Rate Adaptation	0.66	0.45	1.66	0.13
14: PSM	0.38	0.53	1.02	0.13

Table 5.4: Convergence Time (Median, in seconds)

Case: Remark	IGI/PTR	PathChirp	Pathload	WBest
0: Idle channel	1.55	17.43	14.88	0.41
1: UDP crossing	1.42	17.58	20.22	0.42
2: UDP contending	1.29	17.62	17.04	0.42
3: TCP traffic	17.21	17.24	42.06	0.67
4: TCP traffic	7.86	17.22	32.16	0.44
5: UDP crossing	1.35	17.68	19.24	0.42
6: UDP contending	1.30	17.79	17.33	0.42
7: TCP traffic	26.69	18.41	53.90	0.70
8: TCP traffic	19.57	17.89	55.02	0.51
9: Multiple TCP/UDP	1.60	18.10	18.42	0.42
10: Multiple TCP/UDP	23.30	17.15	80.86	0.98
11: Multiple TCP/UDP	28.37	18.27	30.24	0.59
12: Multiple TCP/UDP	15.59	17.45	74.94	0.44
13: Rate Adaptation	1.86	17.48	23.73	0.42
14: PSM	5.43	17.64	84.94	1.03

to these case analysis, the impact of packet sizes, the estimated standard deviations in available bandwidth and the consistency are also analyzed in this section.

### 5.5.2 Case Analysis

#### Idle Channel (Case 0)

Figure 5.17 depicts the estimations, intrusiveness and convergence times for the idle channel (case 0). When the wireless channel is idle, the available bandwidth and the effective capacity are the same. The measured ground truth throughput shows the available bandwidth/effective capacity of 28.94 Mbps, close to the maximum throughput of 31.4 Mbps mentioned in the Cisco documentation on AP<sup>16</sup>. Figure 5.17 shows that *IGI/PTR* and *pathload* significantly under-estimate the available bandwidth. A possible reason is that the packet sizes used during probing these two tools are small. *IGI/PTR* uses a 500 byte packet and *pathload* uses a 200 byte packet. The overhead caused by the sizes of probing packets has been shown to be larger in wireless networks than in wired networks [90, 158, 161], so the maximum throughput will be lower for these smaller packet sizes. Since with a 500 byte or 200 byte packet, the maximum throughput of the wireless network is around 19.2 Mbps or 11.4 Mbps, respectively, even with the consideration of smaller packet sizes, *IGI/PTR* and *pathload* still significantly underestimate the available bandwidth. *PathChirp* and *WBest* get an available bandwidth estimate close to the ground truth. However, *pathChirp* tends to overestimate the available bandwidth with a large variance in the estimation. *Pathload* and *pathChirp* both have long convergence times, because both apply a search algorithm to adapt the probing rate during the estimations.

#### UDP Crossing Traffic (Cases 1 and 5)

Figure 5.18 depicts the estimations, intrusiveness and convergence times when there is one UDP crossing traffic source (case 1). *WBest* performs better than the other tools with low intrusiveness and convergence times and accurate estimated results. The under-estimation

<sup>16</sup>Cisco AVVID Wireless LAN Design. <http://www.cisco.com/en/US/netsol/>

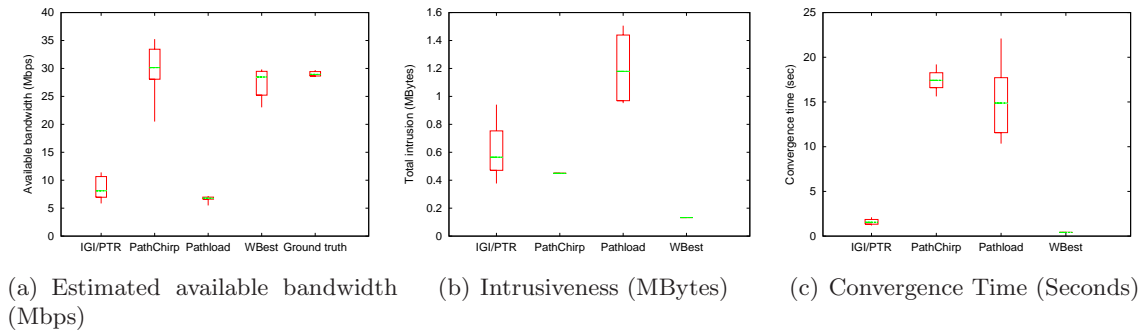


Figure 5.17: Summary Results for Evaluation Case 0 (Idle Channel).

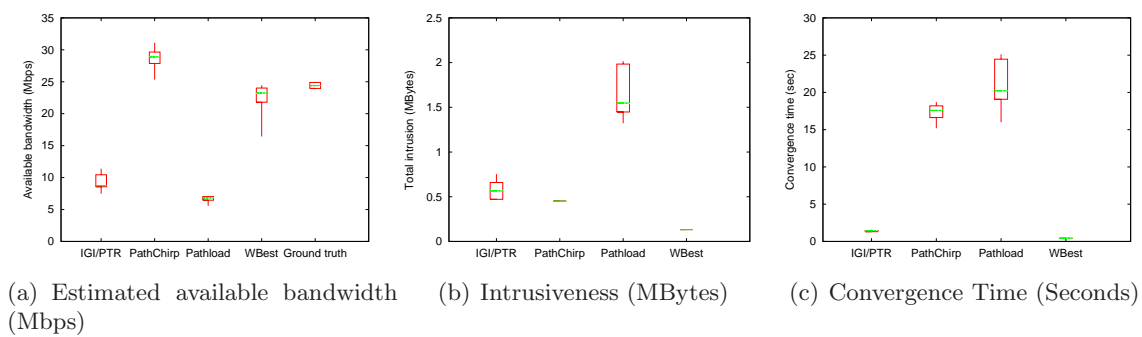


Figure 5.18: Summary Results for Evaluation Case 1 (One UDP Crossing Traffic of 4.6 Mbps).

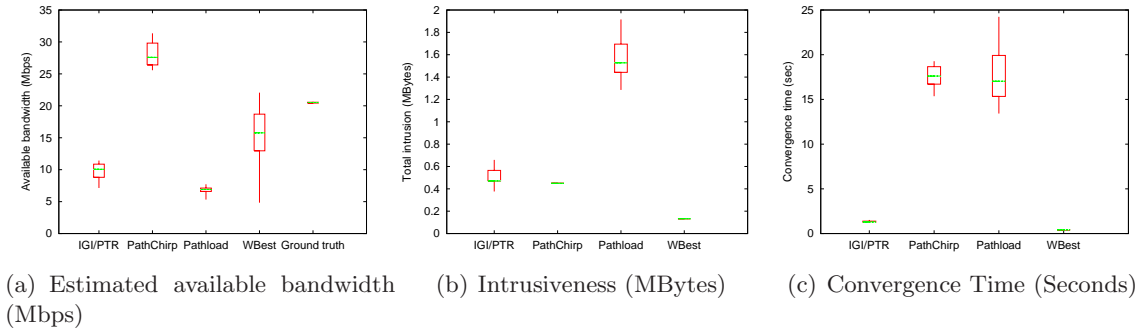


Figure 5.19: Summary Results for Evaluation Case 2 (One UDP Contending Traffic of 4.6 Mbps).

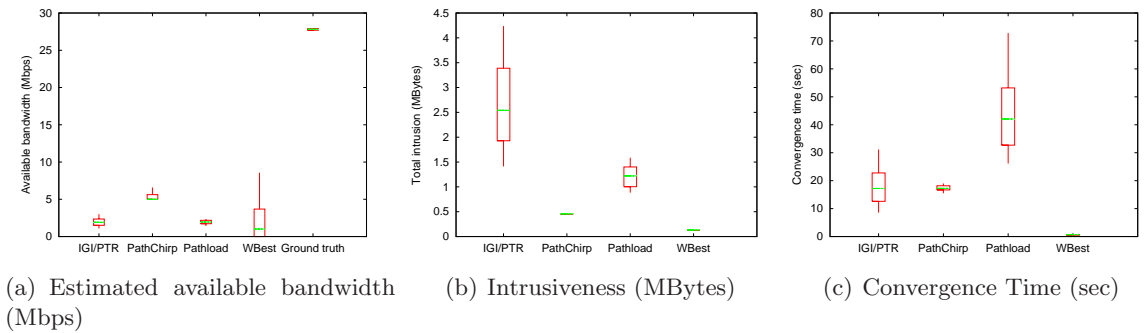


Figure 5.20: Summary Results for Evaluation Case 3 (One TCP Crossing Traffic)

caused by the smaller packet sizes used in *IGI/PTR* and *pathload* shows that they are insensitive to crossing traffic, as well. *Pathload*, in particular, has large intrusiveness and convergence times.

Figure 5.22 depicts the estimations, intrusiveness and convergence times when there are two UDP crossing traffic sources (case 5). These results are similar to the single UDP crossing traffic source. WBest performs better than other tools with multiple sources of UDP crossing traffic.

### UDP Contending Traffic (Cases 2 and 6)

Figure 5.19 shows results when there is one UDP contending traffic source (case 2). WBest still performs well in the presence of contending traffic, however the variance is larger than in the case of crossing traffic (case 1), because contending traffic increases the variance in

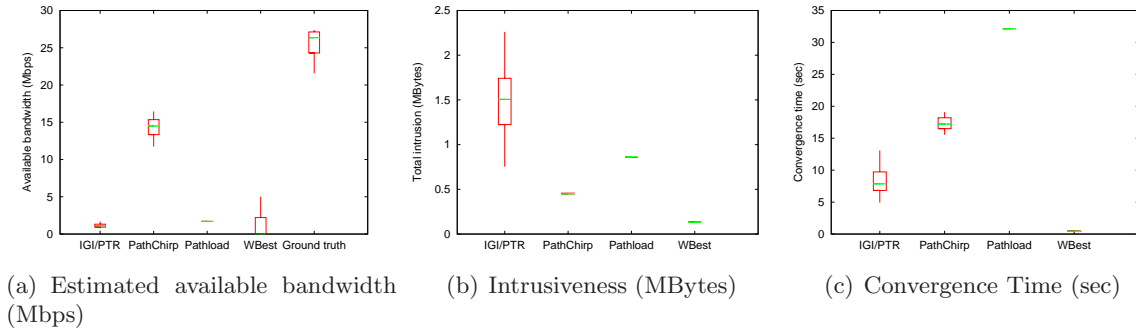


Figure 5.21: Summary Results for Evaluation Case 4 (One TCP Contending Traffic)

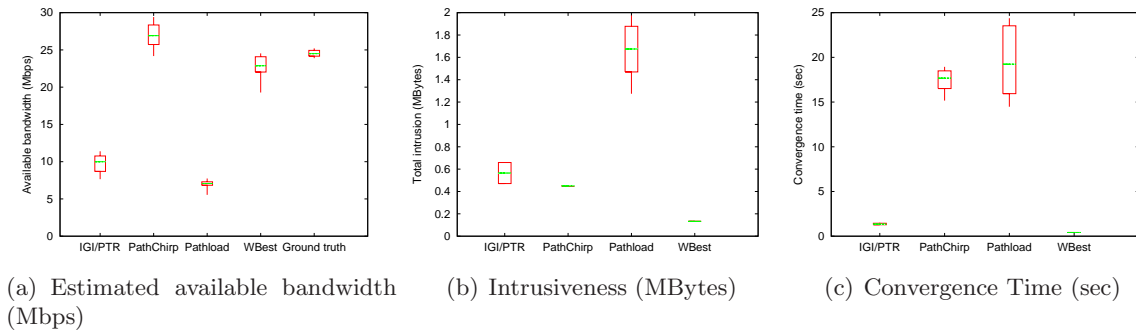


Figure 5.22: Summary Results for Evaluation Case 5 (Two UDP Crossing Traffic of 2.3 Mbps each)

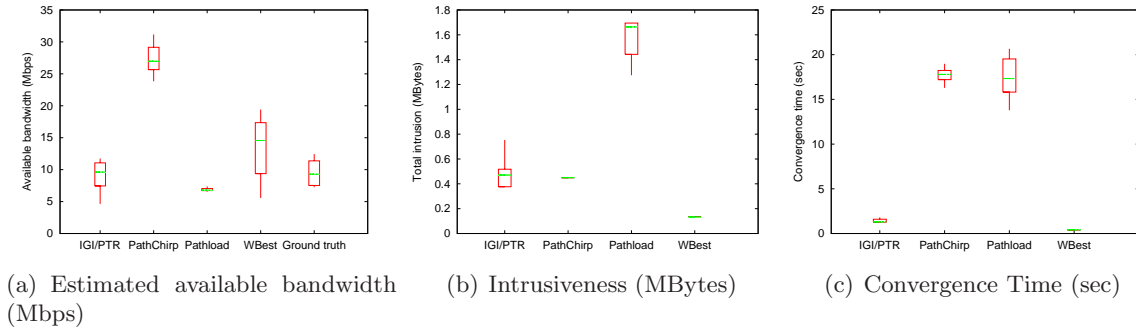


Figure 5.23: Summary Results for Evaluation Case 6 (Two UDP Contending Traffic of 2.3 Mbps each)

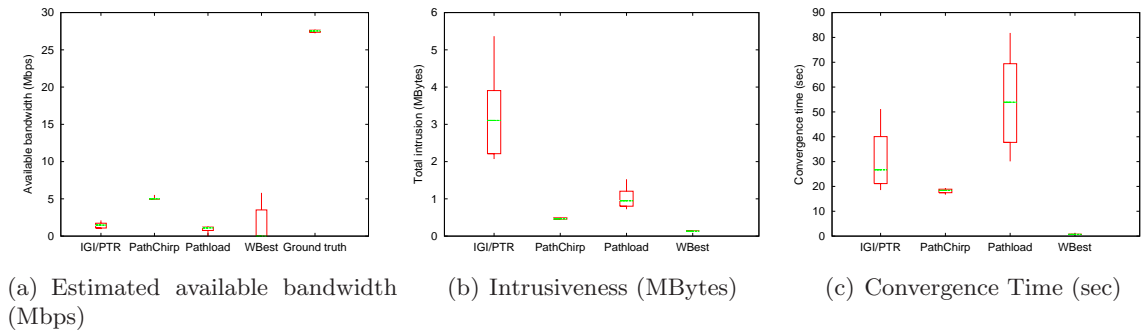


Figure 5.24: Summary Results for Evaluation Case 7 (Two TCP Crossing Traffic)

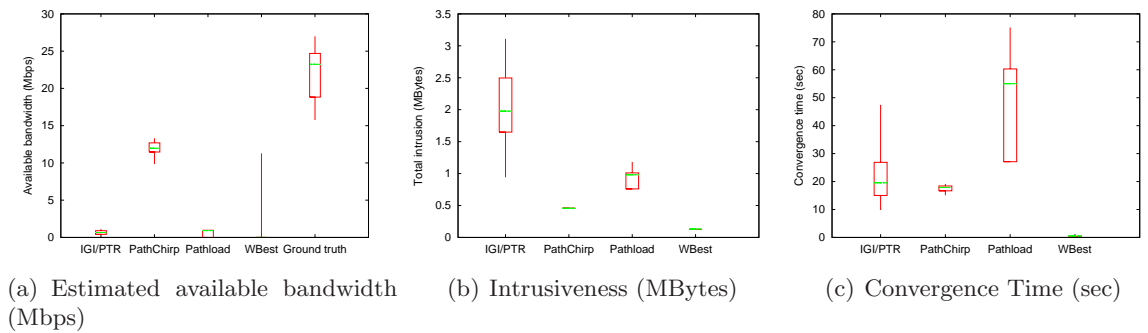


Figure 5.25: Summary Results for Evaluation Case 8 (Two TCP Contending Traffic)

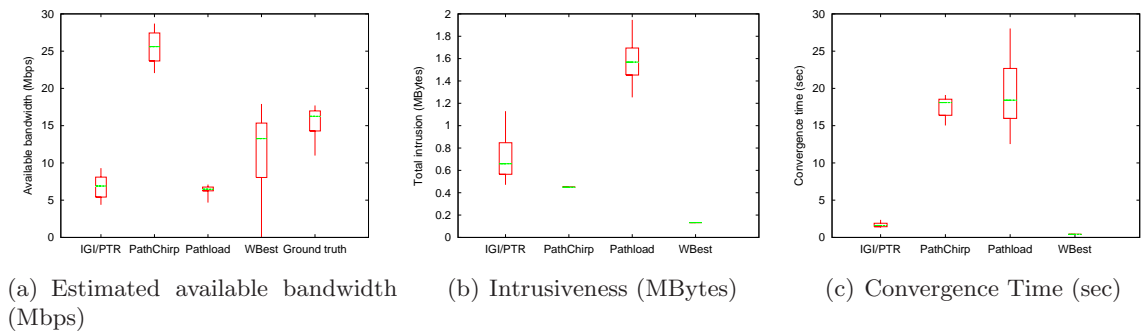


Figure 5.26: Summary Results for Evaluation Case 9 (One UDP Crossing and One UDP Contending Traffic of 2.3 Mbps Each)

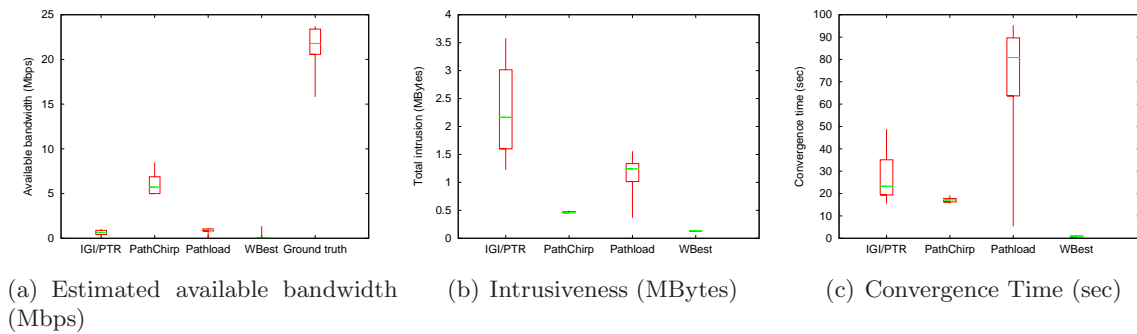


Figure 5.27: Summary Results for Evaluation Case 10 (One TCP Crossing and One TCP Contending Traffic)

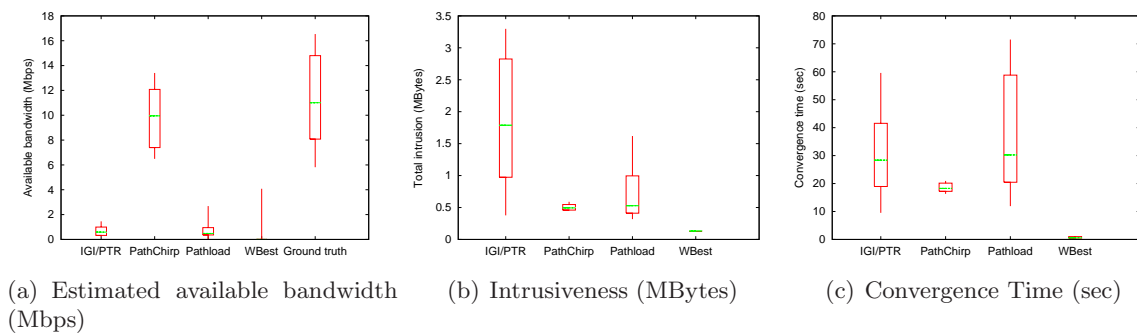


Figure 5.28: Summary Results for Evaluation Case 11 (One UDP Crossing of 2.3 Mbps and One TCP Contending Traffic)

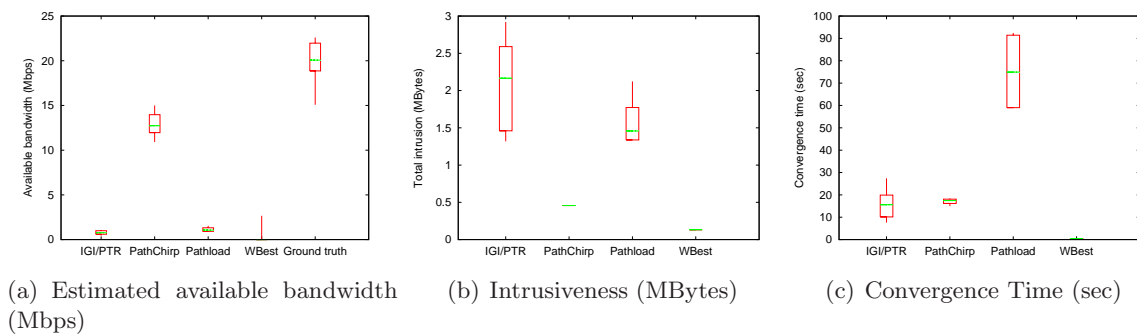


Figure 5.29: Summary Results for Evaluation Case 12 (One TCP Crossing and One UDP Contending Traffic of 2.3 Mbps)

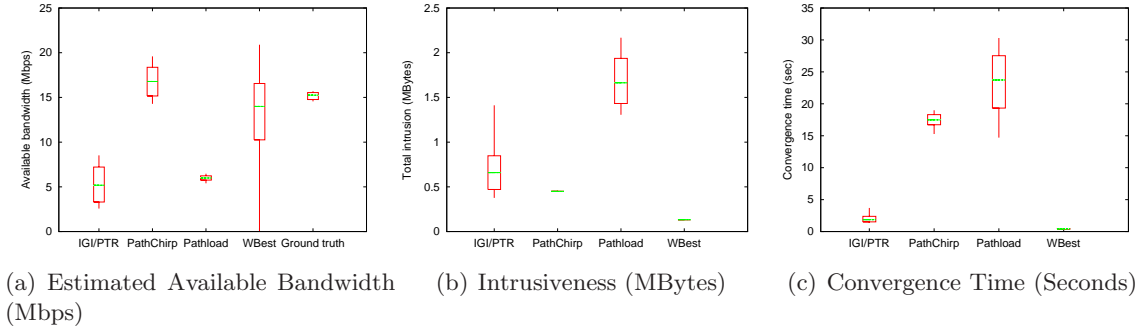


Figure 5.30: Summary Results for Evaluation Case 13 (Wireless Rate Adaptation, Range 1-48 Mbps).

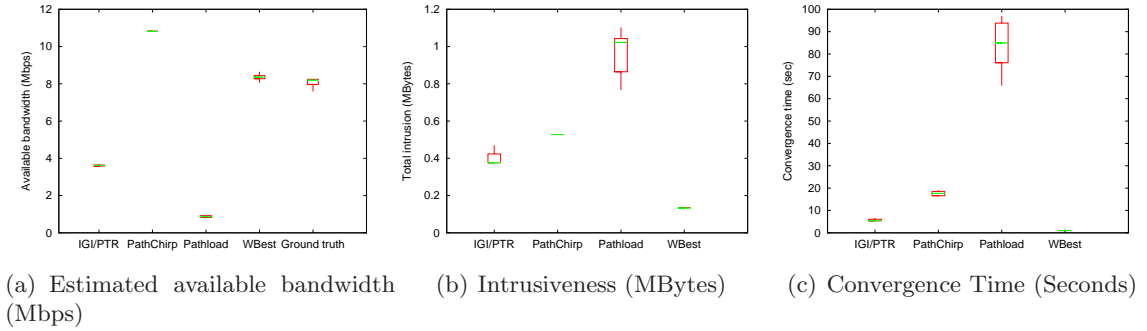


Figure 5.31: Summary Results for Evaluation Case 14 (Power Saving Mode).

delay in accessing the wireless channel. Since the sampling period is smaller in WBest than in other tools, the variance is amortized by other tools, such as *pathload*. Again, comparing case 2 with with case 0 and 1, *IGI/PTR* and *pathload* are not sensitive to contending traffic.

Figure 5.23 depicts the experiment results of two UDP contending traffic sources. Similar to the single UDP contending traffic case (case 2), WBest has a better performance than other tools. However, as discussed in Section 5.4.3 and 5.5.1, multiple UDP contending sources saturate the wireless networks. This causes unexpected rate adaptations and reduces the ground truth measurements. The results from Figure 5.19 and 5.23 show that WBest is sensitive to contending traffic and the number of contending sources, which confirms with the packet dispersion model discussed in Chapter 4.

**TCP Crossing/Contending Traffic (Cases 3, 4, 7, and 8)**

Figures 5.20, 5.21, 5.24, 5.25 show the results of experiment cases with single or multiple sources of TCP crossing/contending traffic. With TCP crossing/contending traffic sources, the available bandwidth is close to zero in theory because the TCP protocol is designed to use the maximum available bandwidth. *IGI/PTR*, *pathload* and *WBest* can successfully estimate the low available bandwidth that is close to zero, while *PathChirp* tends to strongly overestimate the available bandwidth. Moreover, compared to the simple cases, such as case 0, 1 and 2, *IGI/PTR* shows a 3 to 10 times increase in intrusiveness and convergence times. Similarly, *pathload* also shows a significant increase in convergence time. For all cases with TCP traffic, *WBest* performs the most accurate estimation with consistent intrusiveness and convergence times in all cases with TCP crossing/contending traffic.

**Complex TCP/UDP crossing/contending Traffic (Cases 9 to 12)**

Figures 5.26, 5.27, 5.28, 5.29 show the experiments with multiple crossing and contending traffics of both TCP and UDP protocols. Similar to the simple crossing/contending cases, *WBest* performs better than the other tools with low intrusiveness and convergence times and accurate estimated results. For these complex cases, both *IGI/PTR* and *pathload* show noticeably long convergence times that are as high as 28 seconds and 80 seconds, respectively.

**Wireless Rate Adaptation (Case 13)**

Figure 5.30 shows results for wireless rate adaptation (case 13), where the the packet transmission rate and channel access delay vary as in Figure 5.7. With wireless rate adaptation, all the bandwidth estimation tools produce a larger variance than when there is no rate adaptation. However, *pathload* has a lower variance than all other tools because of its large number of probing packets and searching algorithm. Since *WBest* does not use the increases in delay to converge to available bandwidth, it mitigates the impacts of rate adaptation on packet delay. Therefore, *WBest* reflects the accurate available band-

width during the measured period. Due to the short sampling period, WBest results in a relative large variance because it captures the oscillation in available bandwidth. The median of WBest estimation shows that WBest performs the most accurate estimations with consistent intrusiveness and convergence times in the rate adaptation conditions.

### Power saving mode (Case 14)

Figure 5.31 depicts the estimations, intrusiveness and convergence times of the power saving channel. Power saving mode buffers the probing packets at the AP during the client sleep period and sends them as a burst when the client wakes up, greatly impacting all the tools that rely on the delay for estimating bandwidth. For example, as shown in Figure 5.31(c), *pathload* results in much longer convergence times of up to 85 seconds due to variance caused by power saving mode. However, higher probing rates, such as the rates equal to the effective capacity used in WBest, prevent the client from going into power saving mode, thus result in a more accurate estimation. The available bandwidth in power saving mode is much lower than that in the idle case because of the overhead of power saving polling packets exchanged in the power management algorithms.

### Summary of Case Analysis

To provide summary analysis, the estimation error of each case is computed and the distributions of the error versus the convergence time and error versus intrusiveness are drawn in Figure 5.32 and 5.33, respectively. For these figures, on the x-axis, a negative error represents an under-estimation and a positive error represents an over-estimation; and on the y-axis, lower numbers are better. Therefore, good, fast estimates lie in the bottom center of these two figures.

*IGI/PTR* tends to greatly under-estimate the available bandwidth with UDP crossing or contending traffic and even with an idle channel. *IGI/PTR* has widely variable convergence times and intrusiveness, varying by a factor of 20 times for the different cases. *PathChirp* tends to over-estimate the available bandwidth in all cases. *PathChirp* has a consistent

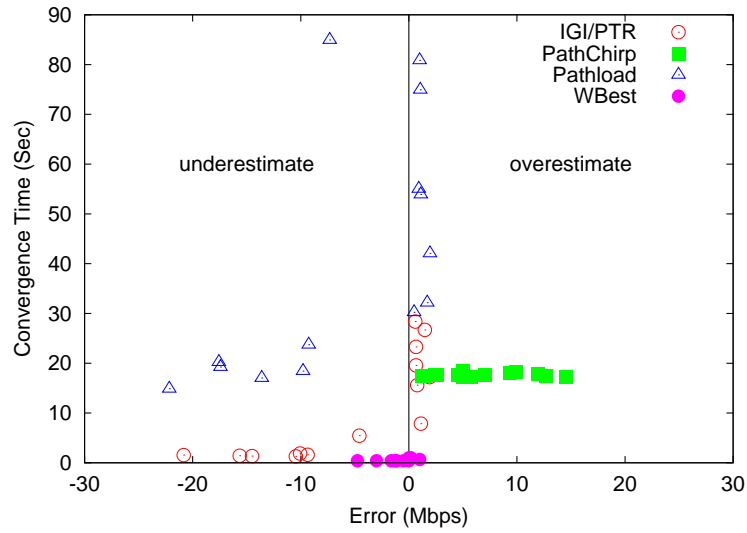


Figure 5.32: Summary of All Experiments – Convergence Time versus Error.

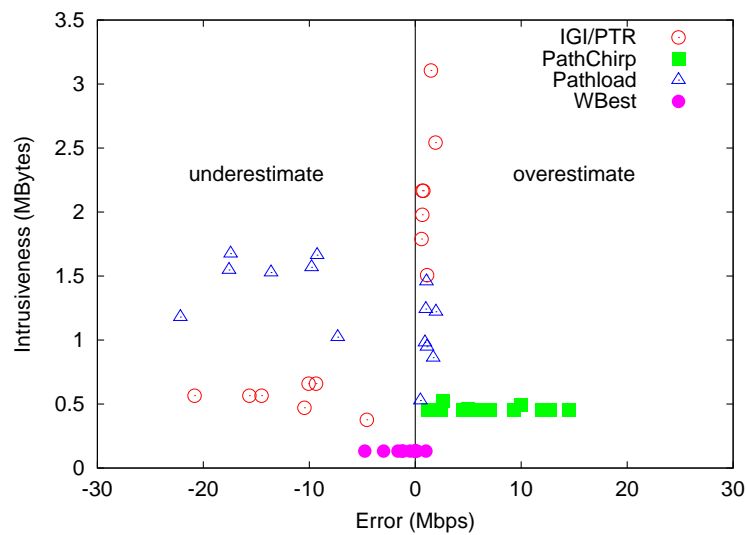


Figure 5.33: Summary of All Experiments – Intrusiveness versus Error.

convergence time of around 17 seconds and a consistent intrusiveness of about 400 KBytes. *Pathload* tends to greatly under-estimate the available bandwidth in most wireless traffic cases including: idle channel, UDP crossing or contending traffic, and rate adaptation. *Pathload* has the longest overall convergence time, taking up to 85 seconds in some cases and even fails to converge in 100 seconds for some crossing and contending cases. WBest generally provides the most accurate estimations compared with the other tools. In most cases, WBest converges in less than half a second, and has a nearly constant intrusiveness of 130 KBytes.

Table 5.5 summarizes the average and standard deviation of relative errors in available bandwidth estimation, intrusiveness, and convergence time for all cases evaluated for each tool. The errors are computed using Equation 5.15. For those cases with no available bandwidth, the errors are computed for the estimated crossing/contending effects, which is defined as  $C_e - A$ . The summary shows that compared with other tools, WBest reduce the average relative error by 82% to 86%, the intrusiveness by 70% to 90%, and the convergence time by 95% to 99%.

Table 5.5: Mean and Standard Deviation of Relative Error (ratio), Intrusiveness (MBytes) and Convergence Time (seconds) of All Evaluated Cases

Metrics	IGI/PTR		PathChirp		Pathload		WBest	
	mean	stdev	mean	stdev	mean	stdev	mean	stdev
Relative Error	0.32	0.30	0.28	0.16	0.37	0.35	0.05	0.07
Intrusiveness	1.31	0.92	0.46	0.02	1.27	0.35	0.13	0.00
Convergence Time	10.29	10.35	17.66	0.37	39.00	24.93	0.55	0.21

For wireless networks, the accuracy of *IGI/PTR*, *pathChirp* and *pathload* is poor because each approach relies on delay changes to measure available bandwidth. In wireless networks queuing delay is not the only source of changes in delay. Wireless contention, MAC layer retries and rate adaptation can all result in delay changes to different extents. These delay changes disturb the searching algorithm for these tools and yield inaccurate results and often increase the convergence times and intrusiveness. Moreover, with higher packet loss

rates in wireless networks, some estimation techniques discard probes impacted by loss to improve accuracy, but this also increases convergence time and intrusiveness.

WBest estimates the available bandwidth without using searching algorithms which means a low, consistent convergence times and intrusiveness. Furthermore, WBest does not depend on delay measurements to measure the available bandwidth. Instead, WBest measures the available bandwidth in terms of fraction of the effective capacity by measuring the relative changes in packet dispersion between two steps. This makes WBest robust even when packet dispersion is impacted by the wireless conditions. In addition, WBest provides a broader range of information on the network than other tools, providing effective capacity, achievable throughput and variance of available bandwidth.

### 5.5.3 Impacts of Packet Sizes

As discussed in the analysis of the idle channel (Case 0), packet size has a significant impact on the bandwidth measurements. We evaluate the impact of packet size for both the WBest probing traffic and the crossing/contending traffic. Figure 5.34 summarizes the results of case 0 with different probing packet sizes. With the configurable probing packet sizes, WBest can perform accurate available bandwidth estimation that is close to the ground truth with the same packet size. This suggests that WBest should be configured to use the same packet size as the application traffic to perform accurate estimations.

Figure 5.35 summarize the WBest estimations of case 1 and 2 with different packet sizes for the crossing/contending traffic. When estimating the wireless networks using WBest with default packet size (1460 bytes), the packet size of crossing/contending traffic can impact the estimations of available bandwidth. Even though the crossing/contending traffic has the same bit rate, the different packet sizes, thus the different packet rates, will impact the estimation result. With high packet rates, the packet based channel access increases the packet dispersion time, thus resulting in lower available bandwidth estimations. The ground truths measured by CBR throughput with the same packet size as the WBest probing traffic are also shown in Figure 5.35. However, the ground truths are less

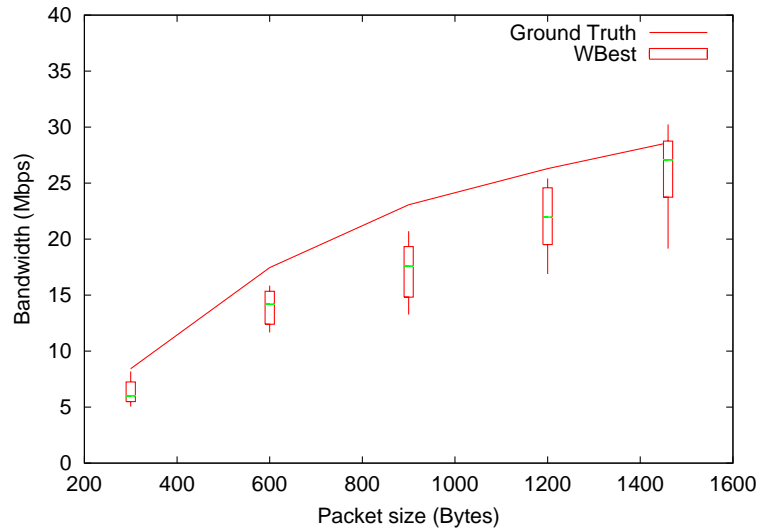


Figure 5.34: WBest Bandwidth Estimation versus Probing Packet Sizes

sensitive to the packet sizes of crossing/contending traffic than is WBest. This is because the probability-based fairness in IEEE 802.11 wireless networks benefits traffic with large packet sizes in terms of throughput. As the packet rate increases, the CBR traffic can keep a higher throughput by unfairly competing with the crossing/contending traffic with small packet sizes. Therefore, the CBR throughput does not represent the ground truth of available bandwidth, but instead represents the achievable throughput. In these experiments, WBest provides a more close-to-real estimation of the available bandwidth than the CBR throughput. As defined in Section 3.2, the available bandwidth estimated by WBest does not include the bandwidth that is actively taken from crossing/contending traffic. Since WBest takes the competing issues into consideration, WBest can be used to estimate the available bandwidth with crossing/contending traffics of different packet sizes, thus providing the available bandwidth that can be used by applications without hurting existing traffic.

#### 5.5.4 Consistency Analysis

The analysis for each case shows that WBest can consistently provide accurate, low intrusiveness and fast convergence time estimations for different wireless network conditions.

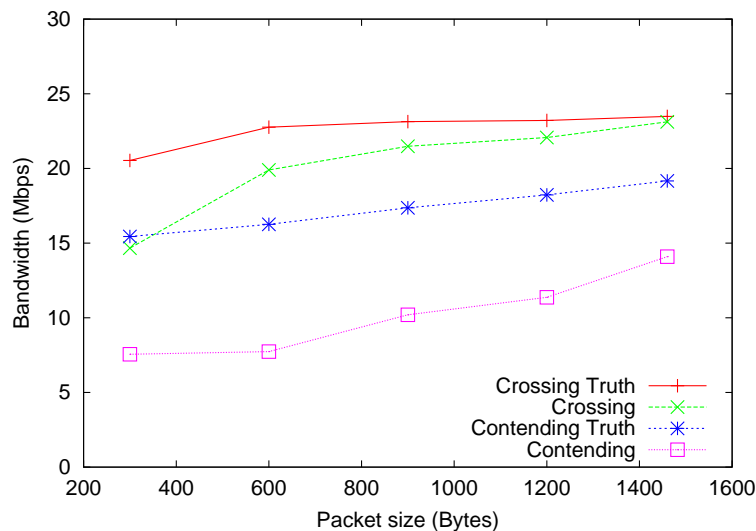


Figure 5.35: WBEST Bandwidth Estimation versus Crossing/Contending Packet Sizes

However, to further evaluate the robustness of WBEST, we analyze the consistency of WBEST estimations within each case. That is, we compute the error of each run relative to the median of all runs in that case, which is computed based on Equation 5.15. Figure 5.36 and 5.37 depict the cumulative distribution of the relative error in effective capacity and available bandwidth estimation for all cases evaluated. For the cases with no available bandwidth, the error is computed for estimated crossing/contending effects. Both the effective capacity and available bandwidth have consistent estimations according to the CDF shown in the figures.

### 5.5.5 Estimating the Standard Deviation in Available Bandwidth

The box-and-whisker plot of estimated standard deviations from all cases is shown in Figure 5.38. In addition, we plot the median of the measured standard deviations in available bandwidth in the same figure. The measured standard deviations are based on the available bandwidth calculated from all packet dispersion samples using Equation 5.8 for each case. As shown in Figure 5.38, WBEST estimates the standard deviation in available bandwidth close to the measured standard deviations. However, in the environments with high contention, such as case 11 with contending UDP traffic and crossing TCP traffic,

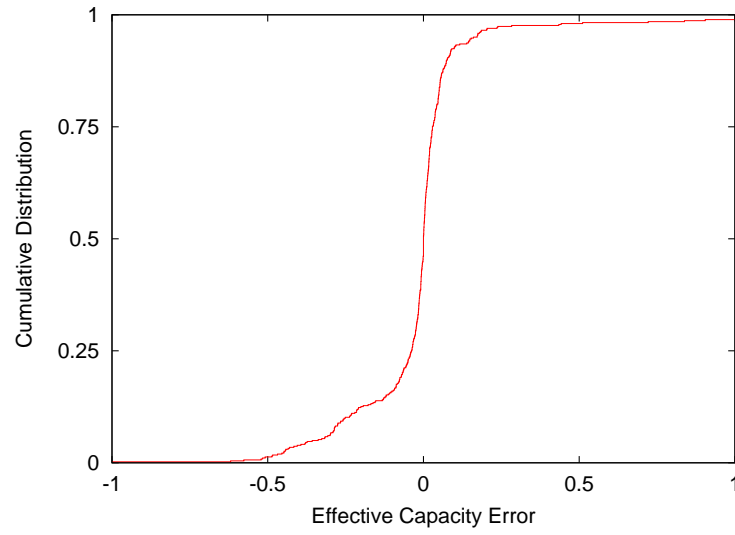


Figure 5.36: CDF of Effective Capacity Error

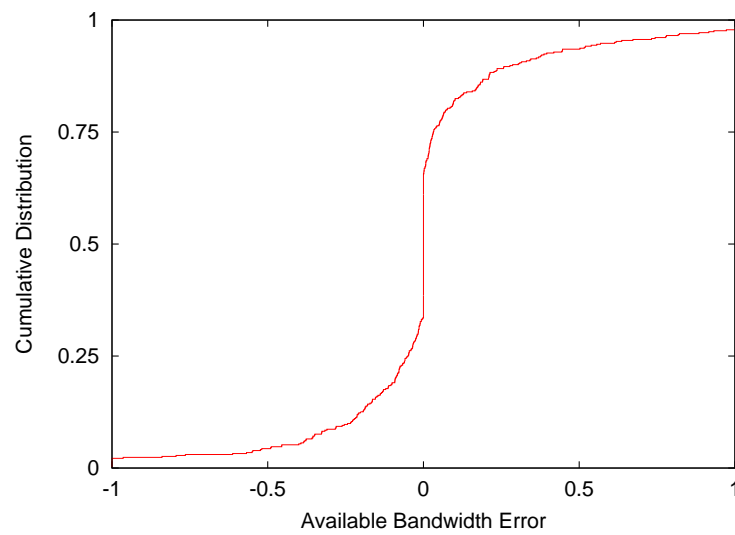


Figure 5.37: CDF of Available Bandwidth Error

WBest may overestimate the standard deviations of available bandwidth.

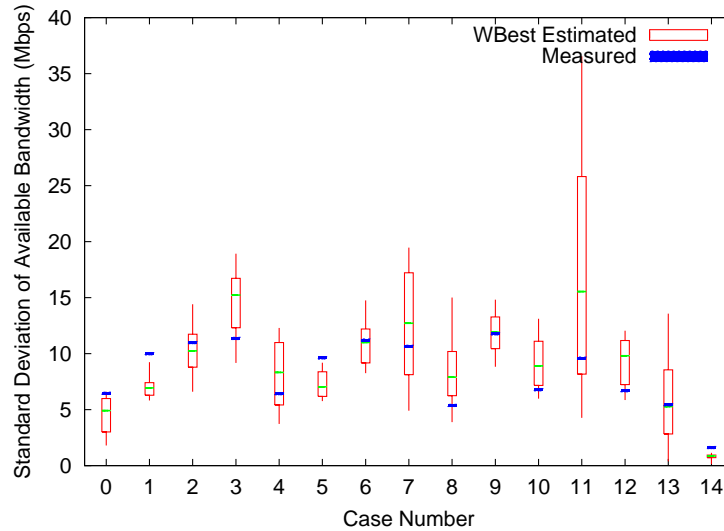


Figure 5.38: Estimated Standard Deviation in Available Bandwidth

## 5.6 Summary

This chapter presents WBest, a new bandwidth estimation tool for wireless networks, designed to provide accurate bandwidth estimation in a short amount of time and without excessively intruding on existing traffic. One advantage of WBest over existing tools is that WBest does not depend upon search algorithms to measure available bandwidth. Instead, WBest statistically measures the relative available fraction of the effective capacity, mitigating estimation delay and the impact of wireless channel errors. WBest is compared with other popular available bandwidth estimation tools in a wireless testbed under a variety of wireless and network conditions. The evaluations show that on average, WBest can effectively reduce the average relative error by 82% to 86%, the intrusiveness by 70% to 90%, and the convergence time by 95% to 99%. The following conclusions can be drawn:

1. Current bandwidth estimation tools are significantly impacted by wireless network conditions, such as contention from other traffic and rate adaptation. This results in inaccurate estimates and high and varying convergence times and intrusiveness. This

makes current tools generally impractical for applications running over a wireless link, such as streaming media, that require fast, accurate, non-intrusive bandwidth estimates.

2. WBest consistently provides fast available bandwidth estimation, with overall more accurate estimations and lower intrusiveness compared with other tools over all conditions evaluated.
3. WBest satisfies the requirement of a bandwidth estimation tool for wireless multimedia streaming applications, providing low convergence time, low intrusiveness and robustness. For example, a bandwidth estimation tool must provide a convergence time less than the initial buffer time, e.g. 5 seconds for Windows Media Player. WBest has consistent convergence times regardless of the network conditions because it estimate available bandwidth with a constant intrusiveness without changing the probing rate or discarding estimation during packet losses.

Chapter 6 applies WBest to multimedia streaming applications to improve the performance of streaming rate selection and buffer optimization in wireless networks.



## Chapter 6

# BROS: Buffer and Rate

# Optimization for Streaming

This chapter presents the Buffer and Rate Optimization for Streaming (BROS) algorithm to improve streaming performance. BROS uses Wireless Bandwidth Estimation Tool (WBest) as discussed in Chapter 5 and models the relationship between buffer size, streaming data rate and available bandwidth distribution. BROS optimizes the streaming data rate and initial buffer size, resulting in reduced frame losses and buffer underflow events, while keeping a small initial buffer delay. Section 6.1 gives a brief overview of the streaming rate selection and buffer optimization in wireless networks. Section 6.2 presents the buffer model based on the available bandwidth distribution. Section 6.3 discusses the BROS algorithm and related issues. Section 6.4 describes the experimental setup. Section 6.5 analyzes the experimental results. Finally, Section 6.6 provides conclusions and possible future work.

### 6.1 Overview

In best effort networks, streaming media applications use streaming rate selection and playout buffers to reduce degradations in performance caused by changes in the available

bandwidth along the path of the streaming flow. However, most streaming rate selection and buffer optimization algorithms are developed for wired networks and can perform poorly over wireless networks. As discussed in Chapter 3, wireless networks often provide rate adaptation, retransmissions, Forward Error Correction (FEC), and channel access control, and other behaviors that are unexpected in wired networks, leading to significant degradation in the effectiveness of streaming rate selection and playout buffer techniques.

Based on a Markov Chain model of the buffer size, streaming data rate and available bandwidth distribution, we develop the Buffer and Rate Optimization for Streaming (BROS) algorithm to concurrently optimize the streaming rate selection and the playout buffer size. BROS applies a low-cost bandwidth estimation approach, the Wireless Bandwidth Estimation Tool (WBest) presented in Chapter 5, at the application layer to provide bandwidth information for the bottleneck wireless network. BROS is incorporated into the Emulated Streaming (EmuS) client-server system in Linux and evaluated on an IEEE 802.11 wireless testbed over a variety of wireless conditions that include an idle channel, channels with contending and crossing traffic, and rate adaptation during poor connection conditions. The evaluation shows that BROS can effectively optimize the streaming rate and initial buffer size based on wireless network bandwidth conditions, and achieve better performance than static rate selection and static or jitter removal buffers. Analysis indicates that BROS can reduce buffer underflow probability by nearly 100%, frame lost rate by about 97% and the total buffer delay from 78% to 87%, compared with static and jitter removal approaches.

## 6.2 Model

### 6.2.1 Model Definitions and Assumptions

Figure 6.1 depicts a typical client-side playout buffer system with a buffer size of  $N$  frames, arrival rate  $\lambda$ , and playout rate  $\mu$ . Based on the buffer occupation, we create the Markov Model of  $N + 1$  states as shown in Figure 6.2. State  $N$  is defined as having a buffer with

$N$  frames, where state 0 is a buffer underflow. The matrix  $P$  in Equation 6.1 presents the transition probability of states  $i$  and  $j$ , where  $0 \leq i, j \leq N$ .

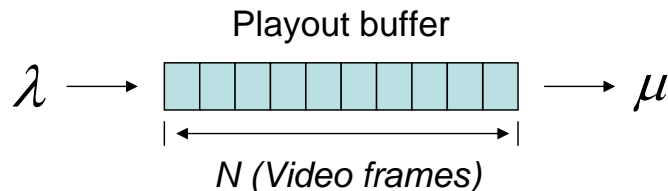


Figure 6.1: Buffer Model

$$P_{i,j} = [p_{i,j}] = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots & p_{0,N} \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\ p_{2,0} & p_{2,1} & p_{2,2} & \cdots & p_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{N,0} & p_{N,1} & p_{N,2} & \cdots & p_{N,N} \end{bmatrix}. \quad (6.1)$$

Most of the previously developed playout buffer models use a Poisson arrival process [60, 61, 73]. Poisson arrivals can be used as a lower bound on system performance when analyzing the buffer behavior [62]. However, for real streaming applications in wireless networks, the buffer space used to smooth interarrival time variance is small relative to the buffer space needed to smooth the variance in available bandwidth. In networks with a large available bandwidth variance, the expected arrival rate of the streaming packets is impacted by the available bandwidth. For example, if the available bandwidth is less than the streaming rate, the expected arrival rate at the playout buffer will also be less than the streaming rate. Therefore, the transition probability model is based on both the available bandwidth and the streaming rate.

To define the probability matrix, the following assumptions are made. First, as discussed in [64, 171], packet loss is modeled as a reduction in available bandwidth. Given a playout buffer of a few seconds, a lost packet will have multiple retransmission opportunities. For typical inter-packet loss rates of less than 20% [172], the probability that a packet is received after a few retransmission attempts is nearly one [64]. Thus, as shown

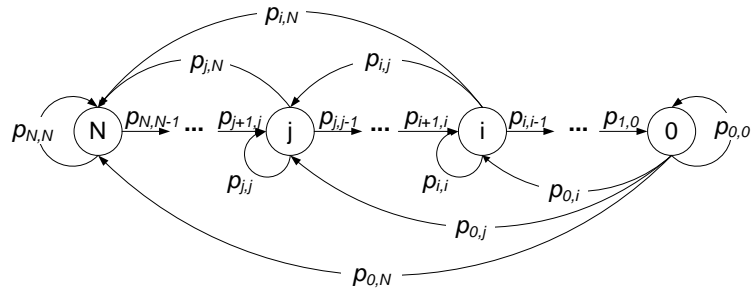


Figure 6.2: States of the Buffer Model

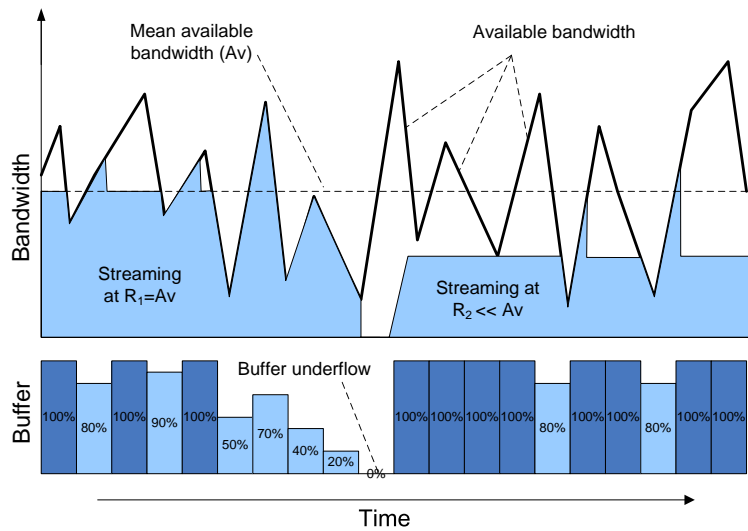


Figure 6.3: Streaming Rate and Available Bandwidth

in Figure 6.3, lost packets are treated as delayed due to insufficient bandwidth. When bandwidth does become available, delayed packets are sent in bursts at a rate equal to the available bandwidth until the buffer is filled again. Second, once a streaming rate is selected, one can model the multimedia content as a constant data rate  $R$  and a constant frame rate  $\mu$  for both streaming and decoding. While frame sizes do depend upon the encoding and type of encoded frame, such as I, B, and P frames in MPEG encoding, a constant frame size  $S = R/\mu$  is used to simplify our model. For a real streaming system, the constant frame size assumption can be accommodated by applying an additional buffer to smooth out the variable frame sizes.

Therefore, considering the constraint of the available bandwidth on the streaming traffic, the frame arrival rate  $\lambda_i$  in frame per seconds is:

$$\lambda_i = \begin{cases} A_i/S, & A_i \leq R; \\ A_i/S, & A_i > R, i < N; \\ \mu, & A_i > R, i = N. \end{cases} \quad (6.2)$$

where  $A_i$  is the available bandwidth in bits per seconds at state  $i$ . Assuming a constant arrival rate  $\lambda_i$  within each frame time  $t = 1/\mu$ , the expected number of frames  $n_i$  that arrive in each frame slot can be approximated by  $n_i = \lambda_i/\mu$ . This facilitates a model of the transition probability between states based on the distribution of the available bandwidth  $A$ . For example, given the Cumulative Distribution Function (CDF)  $F_A(\cdot)$  of the available bandwidth, the probability of  $1 < n_i \leq 2$  can be determined by  $F_A(A_i = 2R) - F_A(A_i = R)$ , which is the transition probability from state  $i$  to  $i + 1$ , where  $A_i > R$ ,  $i < N$ . Similarly, the transition probability matrix can be defined for the buffer model, henceforth referred to as the full model:

$$p_{i,j} = \begin{cases} F_A[(j-i+1)R] - F_A[(j-i)R], & 0 \leq i < j < N, \\ & j-i \leq A_i/R; \\ 1 - F_A[(j-i)R], & 0 \leq i < j, j = N, \\ & j-i \leq A_i/R; \\ 1 - \sum_{j=0, j \neq i}^N p_{i,j}, & 0 \leq i \leq N, j = i; \\ F_A(R), & 0 < i \leq N, \\ & j = i-1; \\ 0, & \textit{elsewhere}. \end{cases} \quad (6.3)$$

The steady state probability distribution of the buffer occupancies,  $\Pi = [\pi_0, \pi_1, \dots, \pi_N]$  can be directly computed by solving the stationary equation  $\Pi = \Pi P$  and  $\sum_{k=0}^N \pi_k = 1$ . The probability of 0 buffer occupancy is the expected buffer underflow probability. However, the full model characterization does not lend itself to a closed form solution, thus it is cumbersome to use this model for real systems due to the required massive matrix computations.

As shown in the right side of Figure 6.3, lowering the streaming rate below the average available bandwidth reduces the buffer requirement to avoid buffer underflow. Therefore, by focusing on situations where the multimedia streaming rate is close to the average available bandwidth, the full model can be simplified to yield a closed form solution for the buffer underflow probability. When  $A_i/R \approx 1$ , then  $p_{i,i+1} \gg p_{i,i+2}, \dots, p_{i,N}$ , the transition matrix  $P$  can be further reduced to a simplified buffer model:

$$p_{i,j} = \begin{cases} 1 - F_A(R), & 0 \leq i < N, j = i+1 \\ & j-i \leq A_i/R; \\ 1 - \sum_{j=0, j \neq i}^N p_{i,j}, & 0 \leq i \leq N, j = i; \\ F_A(R), & 0 < i \leq N, j = i-1; \\ 0, & \textit{elsewhere}. \end{cases} \quad (6.4)$$

Therefore, the closed form solution for the simplified buffer model becomes:

$$\pi_i = \gamma^i \pi_0 \quad (6.5)$$

where  $\gamma = (1 - F_A(R))/F_A(R)$ . The buffer underflow probability  $\pi_0$  for a given buffer size of  $N$  frames is:

$$\pi_0 = \frac{1 - \gamma}{1 - \gamma^{1+N}}, \quad \gamma \neq 1. \quad (6.6)$$

A streaming system with  $\gamma \leq 1$  means streaming at a rate greater than the average available bandwidth and usually results in a high buffer underflow probability. Therefore, the streaming rate selection algorithm selects the initial streaming rate such that  $\gamma > 1$ . Moreover, if a streaming application demands an upper bound on the buffer underflow in terms of  $\pi_0$ ,  $\gamma$  can be computed from  $\pi_0$  using Equation 6.6, and the streaming data rate can be looked up from the inverse CDF of the available bandwidth  $F_A^{-1}(\cdot)$  by:

$$R = F_A^{-1}\left(\frac{1}{\gamma + 1}\right). \quad (6.7)$$

As discussed in [64], the mean time between buffer underflows (MTBBU) can be used as a measure of performance for the buffer underflow event. Each discrete frame slot can be treated as an independent Bernoulli trial with the outcome being either underflow, or no underflow with the probability  $\pi_0$ . Thus, the MTBBU,  $M_U$ , is distributed geometrically over the succession of frame slots as:

$$M_U = \left(\frac{1}{\pi_0}\right) \cdot \left(\frac{1}{\mu \cdot 60}\right) \quad (6.8)$$

where  $\mu$  is the playout rate in frames per second and 60 is the number of seconds in a minute. Given  $M_U$  and the CDF of the available bandwidth, Equation 6.9 is the required buffer size  $N$  in frames:

$$N = \left\lceil \frac{\log(1 + (\gamma - 1)(M_U \cdot \mu \cdot 60))}{\log \gamma} - 1 \right\rceil \quad (6.9)$$

In practice, the minimum client side buffer,  $N'$ , may also include an extra buffer space required for video decoding or playback. For example, an extra buffer  $B_{min}$  may be needed for handling VBR video or encoding dependencies. This research assumes  $B_{min} = 1$ , which means only the frame that is currently being played out is considered. Equation 6.10 shows the expression that can be used to predict the minimum buffer size in practice:

$$N' = N + B_{min} = N + 1 \quad (6.10)$$

### 6.2.2 Model Validations

To study the impact of the simplifications to the full buffer model, outputs from both the full model and simplified model are compared. Figure 6.4 depicts the MTBBU of the full model, which is computed using numerical matrix computations based on Equation 6.3, and the MTBBU of the simplified model based on Equation 6.4. The CDF of the available bandwidth is based on the trace of 900 samples of bandwidth estimations using WBest in a wireless testbed with dynamic rate adaptation. In all cases, the simplified model is close to the full model, especially when the buffer size is small or the streaming rate is close to the median available bandwidth. Moreover, for the  $F_A(R) = 0.475$  cases, the simplified model has a lower MTBBU than does the full model due the approximation of  $p_{i,i+1} \gg p_{i,i+2}, \dots, p_{i,N}$ . However, this small error encourages a conservative estimate for selecting a playout buffer size (e.g., to get the same amount of MTBBU, the simplified model demands a slightly larger buffer size). The simplified model greatly reduces the amount of matrix computation required over the full model.<sup>1</sup>

Our model is further validated with the EmuS system in an IEEE 802.11 wireless testbed using the setup as discussed in Section 6.4. A 10-minute multilayer video is streamed with a one second fixed buffer size at different streaming rates, and the MTBBUs (in minutes) are recorded. The wireless testbed is configured with 802.11b and a poor reception sig-

---

<sup>1</sup>Our numerical solution computation takes about 1.5 seconds to solve the full buffer model for 300 frames.

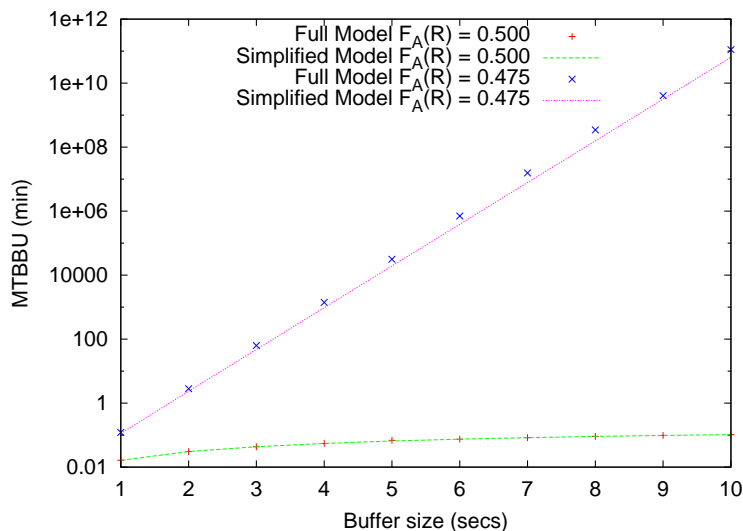


Figure 6.4: Comparison of Full and Simplified Buffer Models

nal strength of -89 dBm that causes MAC layer retransmissions and rate adaptation. The available bandwidth is monitored by WBest before each run and the available bandwidth is around 1.12 Mbps. A wireless sniffer is used to capture the rate adaptation of the wireless network, as shown in Figure 6.5. The streaming rate,  $R$ , as a CDF of available bandwidth,  $F_A(R)$ , is computed and the relationship between  $F_A(R)$  and MTBBU is depicted in Figure 6.6. The MTBBU for the one second buffer is also computed based on the simplified buffer model.

As seen in Figure 6.6, the measured results fit well with our buffer model, with the measured and modeled curves having the same shape at nearly the same places. The measured curve is slightly shifted to the left, which shows that for the same streaming rate, the wireless uncertainties, such as bursty loss, result in a small MTBBU, and thus a higher buffer underflow probability than in the model. Moreover, the measured MTBBU for high streaming rates, such as for rates with  $F_A(R) > 0.5$ , is slightly larger than the modeled MTBBU. This is because in the streaming client-server system implemented, when the buffer underflows, the rebuffer mechanism stops playback of the video until the buffer is filled again, which effectively increases the buffer size for each buffer underflow and intentionally raises the MTBBU for periods after the underflow.

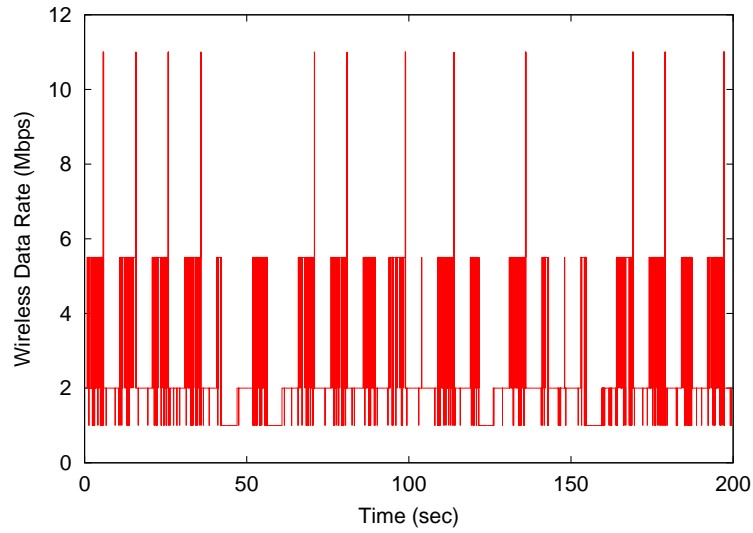


Figure 6.5: Typical Rate Adaptation

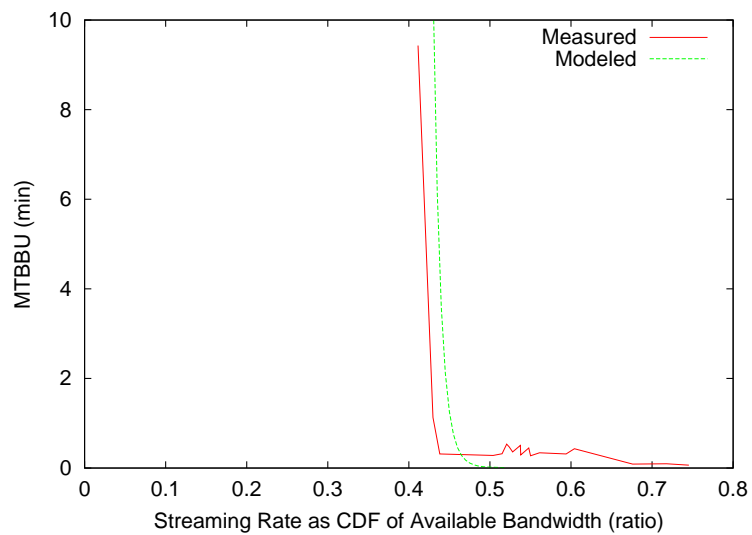


Figure 6.6: MTBBU and the CDF of the Streaming Rate

### 6.2.3 Model Results

The streaming buffer model can be used to estimate the minimum buffer required to achieve a MTBBU with a given rate and known network conditions, such as the CDF of the available bandwidth  $F_A(\cdot)$ . Figure 6.7 shows the minimum buffer size required for various selections of the streaming rate as a fraction of the available bandwidth. The higher the streaming rate, the larger the buffer needed to achieve a desired MTBBU. Moreover, as the streaming rate gets closer to the median of the available bandwidth, the benefits to MTBBU for an increasing buffer size gets smaller.

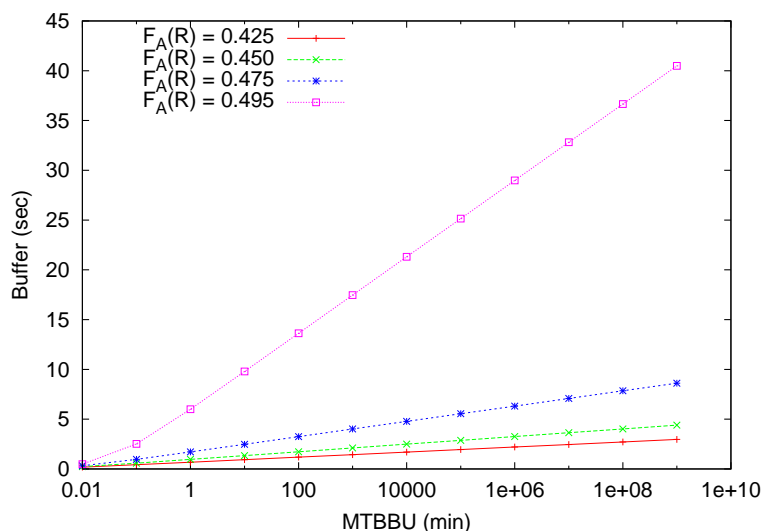


Figure 6.7: Required Buffer Size versus MTBBU

The buffer model can also be used to explore the relationship between MTBBU, buffer size and streaming rate from a different aspect. Figure 6.8 depicts the relationship of MTBBU and streaming rate for 4 different fixed sized buffers: 1, 3, 5 and 7 seconds. For a giving buffer size, reducing the streaming rate decreases the MTBBU, thus providing fewer buffer underflow events. For a modest buffer buffer size (i.e. 5 seconds), a small decrease in the streaming rate selected results in a greatly increased MTBBU. Similarly, for a small buffer size (i.e. 1 second), a small decrease in the streaming rate has much less effect on the MTBBU.

Figure 6.9 shows the relationship between buffer size and streaming rate for a given

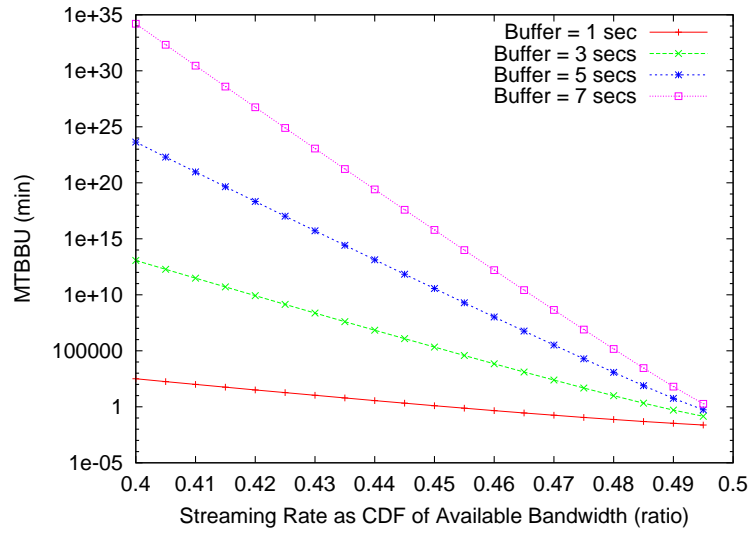


Figure 6.8: MTBBU versus Streaming Rate

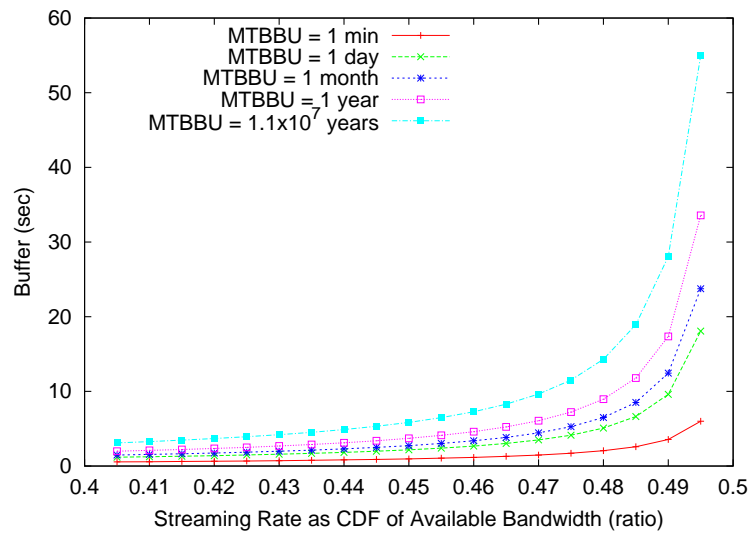


Figure 6.9: Buffer Size versus Streaming Rate

MTBBU, a value that could be provided by the content service provider. The higher MTBBU values are more sensitive to the buffer size as the streaming rate selected is increased, with dramatic increases in required buffer size as the streaming rate approaches the median (0.5, on the far right) of the available bandwidth. The ‘knee’ for the curves occur when the streaming rate selected is around 0.485 of the available bandwidth. Streaming rates higher than this show a marked increase in the buffer size requirements. However, also note at the far left of the graph that streaming rates only slightly lower, around 0.4 of the available bandwidth, can have a marked decrease in the MTBBU for only a slight (around 1 second) increase in the buffer size.

## 6.3 Buffer and Rate Optimization for Streaming (BROS)

### 6.3.1 Algorithm

With the buffer model discussed in Section 6.2, the maximum  $F_A(R)$  can be estimated given a maximum tolerable buffer size to ensure a target MTBBU, thus allowing selection of an appropriate maximum streaming rate by lookup from the inverse CDF of the available bandwidth,  $R = F_A^{-1}(\cdot)$ . The minimum buffer required with given streaming rate and CDF of available bandwidth can also be estimated to ensure a target MTBBU. As discussed in Chapter 5, WBest can be used to estimate available bandwidth distribution  $F_{\hat{A}}(\cdot)$  in wireless networks. Combining WBest and the buffer model provides Algorithm 6.1 with streaming rate selection and buffer optimization using the bandwidth estimation information.

The BROS algorithm has three stages. At line 1, the first stage uses WBest to estimate the average and variance of the available bandwidth, which can be used to infer the distribution of available bandwidth of the wireless network. Starting at line 2, the second stage first selects the maximum affordable streaming rate,  $Th_R$ , given the available bandwidth, target MTBBU and tolerable client buffer size. The selection of  $Th_R$  uses Equations 6.6-6.10 and looks up the streaming rate from the inverse CDF of the available bandwidth. The

**Algorithm 6.1** Buffer and Rate Optimization for Streaming (BROS).

---

**Require:** Client tolerable delay:  $N_{max} > 0$   
**Require:** Target MTBBU:  $M_U > 0$   
 // Perform WBest Bandwidth Estimation  
 1: CDF of Available Bandwidth:  $F_{\hat{A}}(\cdot) \leftarrow WBest$   
 // Select streaming rate ( $R$ )  
 2: Rate Threshold:  $Th_R \leftarrow (N_{max}, F_{\hat{A}}(\cdot), M_U)$   
 3: **for**  $i = 0, MAX$  **do**  
 4:   **if**  $R_i < Th_R \leq R_{i+1}$  **then**  
 5:      $R \leftarrow R_i$   
 6:   **end if**  
 7: **end for**  
 8: **if**  $Th_R \leq R_0$  **then**  
 9:   Stop {Not enough bandwidth for streaming}  
 10: **else**  
 11:   // Optimize buffer ( $N'$ )  
 12:    $N' \leftarrow N(M_U, F_{\hat{A}}(R_i)) + 1$   
 13: **end if**

---

second stage proceeds by selecting the highest encoded streaming rate  $R_i$  less than  $Th_R$ , as shown in line 5. Where  $i$  denotes the encoded level between 0 and the maximum available encoding level. If there is no encoded streaming rate less than  $Th_R$ , the algorithm exits.<sup>2</sup> From line 10 to line 13, the third stage optimizes the buffer size based on the selected rate  $R$ , the MTBBU, and the distribution of the available bandwidth using Equations 6.6-6.10, which are summarized as  $N' = N(M_U, F_{\hat{A}}(R_i)) + 1$  as shown in line 12.

BROS is run at the streaming server. The streaming client typically provides the maximum tolerable buffer size and target MTBBU when the initial connection to the server is made. WBest is run between server and client, with the client returning the results to the server. When the server selects the streaming rate and buffer size, this information is conveyed back to the client and then streaming commences. The client buffers the streaming data until the playout buffer is full and then starts playout.

---

<sup>2</sup>Note, this case also occurs when there is no available bandwidth.

### 6.3.2 Approximate CDF of Available Bandwidth

Even though WBest can provide the mean and standard deviation of the available bandwidth estimates, it is difficult to get a closed form equation for the CDF of the available bandwidth. To simplify the algorithm and reduce the time of recording (and transmitting) multiple bandwidth estimate samples that make up the CDF, the estimated mean  $\mu_A$  and standard deviation  $\sigma_A$  of the available bandwidth are used to approximate the CDF of the available bandwidth using a normal distribution. Therefore, the approximated CDF of the available bandwidth is  $F_{\hat{A}}(\cdot) = Normal(\mu_A, \sigma_A^2)$ . To validate this approximation with measurements, four different setups in our IEEE 802.11 wireless testbed are used: an idle channel, a channel with 5 Mbps downstream crossing traffic, a channel with 5 Mbps upstream contending traffic, and channel with link rate adaptations. For each setup, WBest is repeated 30 times and the CDF for the available bandwidth is recorded. The normal CDFs are then generated based on the median of the mean and standard deviation of the estimated available bandwidth from the 30 WBest runs. The CDF of the available bandwidth and the approximated available bandwidth CDF using the normal distribution are compared by the relative error of the 25th and 75th percentile values. The relative error is computed as  $(F_A^{-1}(x) - F_{\hat{A}}^{-1}(x))/F_A^{-1}(x)$ , where  $F_A^{-1}(x)$  and  $F_{\hat{A}}^{-1}(x)$  are the inverse CDFs of the available bandwidth and approximated available bandwidth, and  $x$  is the CDF value, such as 0.25 and 0.75. Thus, a positive relative error denotes an under approximation, while a negative one denotes an over approximation.

As shown in Table 6.1, the results confirm that using a normal distribution can closely approximate the available bandwidth. Moreover, the normal distribution tends to have a lower value than the available bandwidth distribution for the same CDF value in the region from zero to the median. Since BROS sensibly only selects a streaming rate lower than the median available bandwidth, this implies that using a normal distribution instead of the sampled available bandwidth distribution results in a conservative estimation of the rate selected, which is helpful to avoid buffer underflow.

Figure 6.10 shows a closer look at the effectiveness of using a normal distribution to

Table 6.1: Relative Error of Approximating Available Bandwidth using Normal Distribution

Setup	25th percentile	75th percentile
Idle channel	< 0.01	-0.07
Contending traffic	0.16	-0.23
Crossing traffic	0.14	-0.17
Rate adaptation	0.05	-0.01

approximate the available bandwidth for the case of rate adaptation. Even though rate adaptation has link capacity changes in fixed intervals, the interactions with the MAC layer retries makes the available bandwidth closely follow a normal distribution. In other words, under conditions of bad signal strength, it may take multiple retries to transmit at a higher data rate, while taking fewer retries to transmit at a lower data rate, thus “smoothing” the fixed capacity steps to follow a normal distribution.

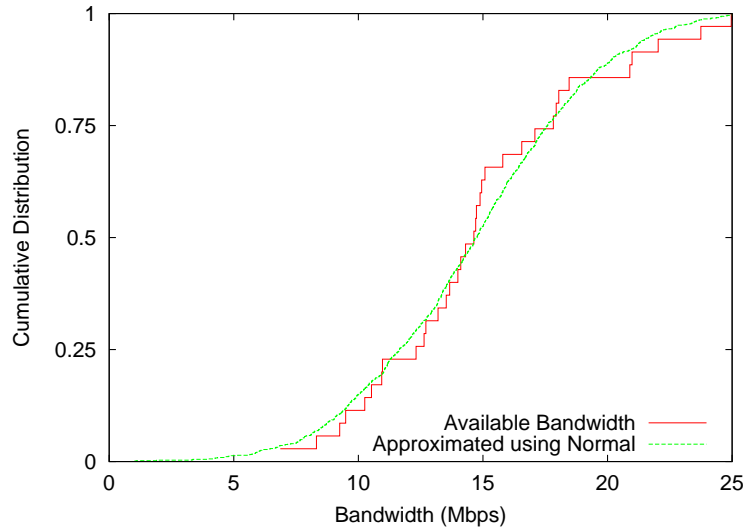


Figure 6.10: Approximating Available Bandwidth using Normal Distribution

## 6.4 Evaluation

### 6.4.1 Wireless Testbed

The Buffer and Rate Optimization for Streaming (BROS) algorithm is implemented and evaluated using an IEEE 802.11 wireless testbed. As shown in Figure 6.11, the wireless testbed consists of a streaming server (wbestserver) that performs bandwidth estimation and streams video to a client, a traffic server (tgenserver) that generates crossing and contending traffic, a wireless AP and three clients (Client A, B and C). The AP in the testbed is a Cisco Air-AP1121G<sup>3</sup> supporting IEEE 802.11b/g. Both servers are PCs with P4 3.0 GHz CPUs and 512 MBytes RAM and the three clients are PCs with P4 2.8 GHz CPUs and 512 MBytes RAM. All the testbed PCs run SUSE<sup>4</sup> 9.3 Linux with kernel version 2.6.11. The servers connect to the AP with a wired 100 Mbps LAN, and the clients connect to the AP with a IEEE 802.11b/g WLAN using Allnet<sup>5</sup> ALL0271 54 Mbps wireless PCI card with a prism GT chipset.<sup>6</sup>

As labeled in Figure 6.11, Client A is configured as the streaming client to receive streaming traffic (1) from the streaming server. Client B, Client C and the traffic server generate crossing traffic (2) or contending traffic (3) using the Multi-Generator Toolset (mgen) v4.2b6<sup>7</sup> over UDP. Client B or C is also configured as a wireless sniffer<sup>8</sup> to monitor the traffic in the testbed when not being used as a traffic generator.

### 6.4.2 Emulated Streaming (EmuS) Client/Server System

To evaluate BROS, we develop an emulated streaming client-server system, called *Emulated Streaming (EmuS)*, with initial buffer and rate selection features. EmuS has features common of many commercial streaming systems, including multiple encoded video layers, configurable initial buffer size, and an RTSP-like communication mechanism. EmuS uses

<sup>3</sup><http://www.cisco.com/en/US/products/hw/wireless/ps4570/index.html>

<sup>4</sup><http://www.novell.com/linux/>

<sup>5</sup><http://www.allnet-usa.com/>

<sup>6</sup><http://www.conexant.com/products/entry.jsp?id=885>

<sup>7</sup><http://pf.itd.nrl.navy.mil/mgen/>

<sup>8</sup><http://perform.wpi.edu/wsniffer/>

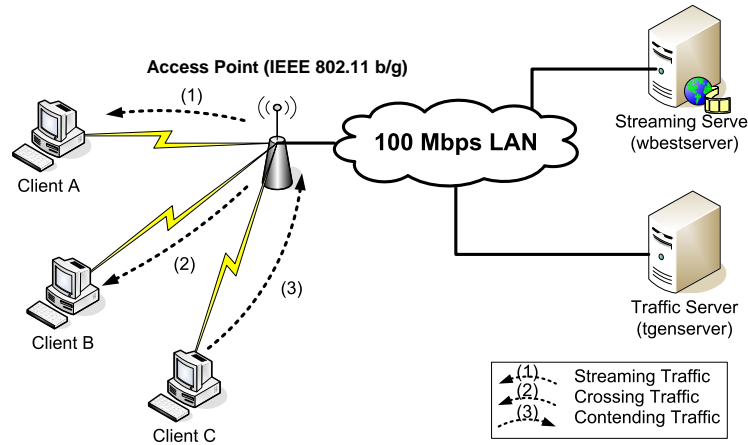


Figure 6.11: Network Path with Last Mile Wireless Network.

two channels, a control channel running over TCP and a data channel running over UDP. Even without a real media codec, EmuS provides framing and packetization functionalities to emulate a real streaming system. EmuS uses CBR encoding with equal sized frames and a fixed size smoothing buffer. While actual video is typically VBR, smoothing buffers are often used to enable decoding and avoid packet bursts. The client detects frame losses based on out-of-order sequence numbered frames and sends NACK messages back to the server asking for retransmission until the frame is too late to be played out. The client also applies frame repair, which repeats the last good frame for each lost frame, to mitigate the impact of frames loss. When buffer underflow occurs, the client stops playback and starts re-buffering until the buffer has filled, then resumes playback. The client records statistical information, such as buffer underflow, frame rate, frame loss, retransmission, etc.

Figure 6.12 and Figure 6.13 depict the actual buffer occupancy, and arrival/playout framerate, respectively, recorded for a typical EmuS streaming session, shown for a 50 second time slice starting about 2.5 minutes after the session starts. Note that the buffer occupancy fluctuates in Figure 6.12 in response to fluctuations in the available bandwidth, evidenced by the fluctuations in the arrival frame rate in Figure 6.13. As long as the buffer remains above 0, the frame playout rate in Figure 6.13 remains at 30 frames per second. However, at time 220 seconds, the buffer drops to zero (a buffer underflow event) and frame

playout stops until the buffer is filled at about time 227. At time 232, the buffer again drops to zero and frame playout stops until the buffer fills at about time 239.

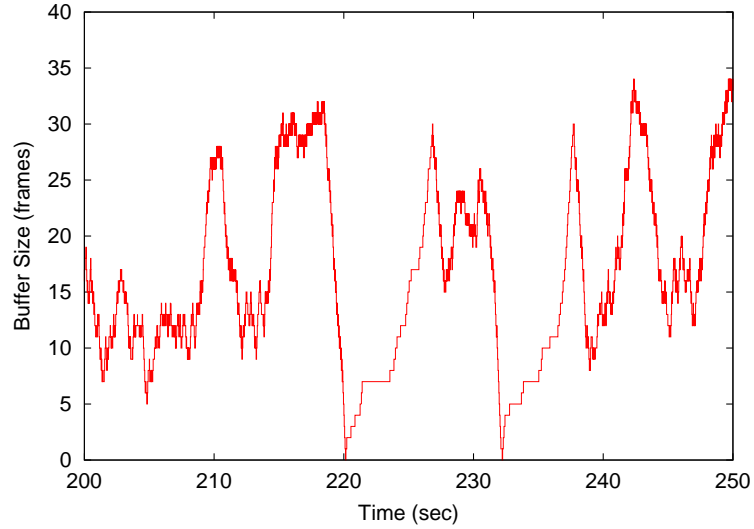


Figure 6.12: Typical Buffer Size Fluctuation

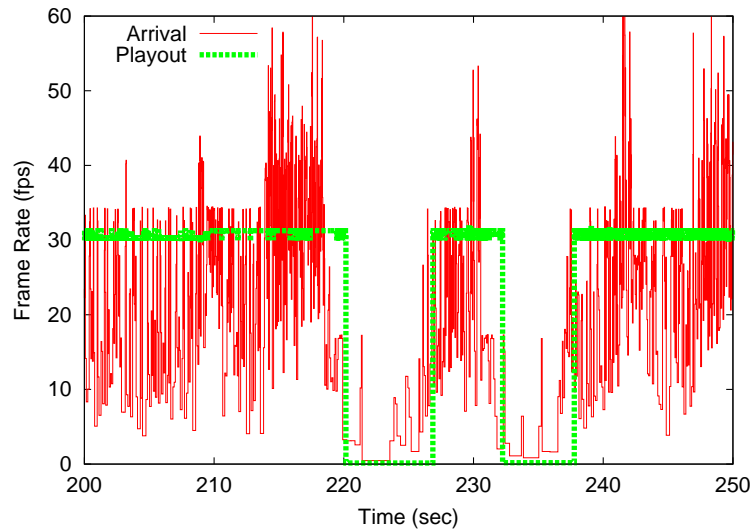


Figure 6.13: Frame Arrival and Playout

As discussed in Section 6.3, a proper MTBBU value needs to be indicated in order to compute the buffer size based on streaming rate and vice versa. As discussed for Figure 6.9, as long as the streaming rate is kept under 0.485 of the available bandwidth, a modest increase in buffer size can result in a significantly increased MTBBU. In light of this, for

all experiments, an extremely low target MTBBU of  $\pi_0 = 1 \times 10^{-16}$  is used based on the underflow probability in Equation 6.8. In practice, for future experiments, both BROS (and EmuS) allow configuration of the targeted MTBBU to fit the application requirements.

### 6.4.3 Experiment Design

Four test cases are designed to evaluate BROS running on EmuS in different wireless network setups. BROS is compared with a fixed buffer size, fixed streaming rate, and jitter-removal buffer algorithms.

The encoded layers of the streaming media used in our experiments follows the default multilayer encoding profile for Windows Media Encoder,<sup>9</sup> which has 10 encoding layer rates from 28 Kbps to 1.1 Mbps, and 4 encoding rates for higher layers at 2.1, 3.6, 5.1 and 6.8 Mbps. The video media length is 2 minutes, which is the median length of streaming video available on the Web [173]. The wireless testbed is set to 802.11b with an average available bandwidth of 6.4 Mbps as measured by WBest. Four different test cases are setup: an idle channel (with good reception quality), a channel with crossing traffic, a channel with contending traffic and a channel with rate adaptations (with poor reception quality). The setup details are listed in Table 6.2. For each test case, the streaming traffic is sent downstream from the wbestserver to Client A. The rate adaptation case is setup by manually reducing both the transmission power of AP and Client A to generate rate adaptation, as in Figure 6.5, along with the accompanying MAC layer retries. Table 6.2 provides a summary view of the network conditions in that case, with the medians of the mean and standard deviation of the available bandwidth of all test runs for each case.

For each test case, the streaming sessions listed in Table 6.3 are run for 30 times in sequence. First, the sessions with BROS are run with a maximum tolerable buffer size of 5 seconds. Second, the sessions with a 5 second fixed buffer size are run, first with a streaming rate higher than the mean available bandwidth (H), then with a streaming rate close to the mean available bandwidth (M), and lastly with a streaming rate less than the

---

<sup>9</sup><http://www.microsoft.com/windows/windowsmedia/default.mspx>

Table 6.2: Evaluation Setups

Case	Configuration
1. Idle channel	No other traffic $RSSI > -38dBm$ Available Bandwidth: $mean = 6.4Mbps, stdev = 0.4Mbps$
2. Contending traffic	1.2 Mbps UDP: Client C to tgenserver $RSSI > -38dBm$ Available Bandwidth: $mean = 4.7Mbps, stdev = 2.3Mbps$
3. Crossing traffic	1.2 Mbps UDP: tgenserver to Client B $RSSI > -38dBm$ Available Bandwidth: $mean = 5.2Mbps, stdev = 1.9Mbps$
4. Rate adaptation	No other traffic, $-89dBm \leq RSSI \leq -85dBm$ Available Bandwidth: $mean = 3.5Mbps, stdev = 1.6Mbps$

mean available bandwidth (L). Third, traces for the fixed buffer tests (H, M, L) are used to determine the performance of the jitter removal buffer algorithm.

Table 6.3: Sessions for Each Case

Sessions		Case 1,2,3		Case 4	
		Rate(Mbps)	Buf(sec)	Rate(Mbps)	Buf(sec)
BROS		Auto	Auto	Auto	Auto
Fixed buffer	H	6.8	5	5.1	5
	M	5.1	5	3.6	5
	L	3.6	5	2.1	5
Jitter buffer	H	6.8	$B_{jit}$	5.1	$B_{jit}$
	M	5.1	$B_{jit}$	3.6	$B_{jit}$
	L	3.6	$B_{jit}$	2.1	$B_{jit}$

The jitter removal buffer  $B_{jit}$  is computed using an approach similar to that discussed in [69]. Most jitter removal buffer algorithms propose variable sampling and averaging algorithms to estimate the network jitter. However, since the frame arrival traces for the entire session are used, no jitter prediction is actually needed. Instead, jitter is measured as the difference between the 95th percentile and 5th percentile of frame inter-arrival times, denoted by  $D_{.95} - D_{.05}$ . Therefore, the buffer size for jitter removal  $B_{jit}$  is computed as:

$$B_{jit} = \frac{1}{\mu} \left[ \frac{D_{.95} - D_{.05}}{1/\mu} + 1 \right] \quad (6.11)$$

where  $\mu$  is the frame rate. With the traces of frame arrival data gathered from the fixed buffer sessions, the jitter removal buffer algorithm is applied to decide the buffer size needed to remove the jitter for each session. Then the session is trace driven using the frame arrival data with  $B_{jit}$ , recording the same set of statistics as for the measured experiments, buffer underflow events, frame losses, and playout delay.

## 6.5 Results

BROS is evaluated with four different network setup cases, each having 120 two-minute video sessions and 90 trace-based sessions. The total data collected includes more than 16 hours of streaming, recording buffer underflow events, frame arrival and playout, frame loss and buffer delay. BROS is evaluated by comparing the rate selection, buffer optimization results, and the related streaming quality metrics [174]: buffer underflow events, frame losses and buffer delay.

Figure 6.14 depicts the rate selection results for the four test cases. The average streaming rate selected for the 120 sessions for each case is shown along with the standard deviation. With the input of bandwidth estimation information from WBest (shown in Table 6.2), the BROS algorithm correctly selects the maximum available encoding rate that was lower than the available bandwidth. Moreover, BROS also takes into consideration the variance of the available bandwidth. For example, for the crossing traffic case, even though the median available bandwidth is higher than the encoding rate of 5.1 Mbps, on average, BROS selects the next lower encoding rate, 3.6 Mbps because the variance in the available bandwidth causes the rate of 5.1 Mbps to have a higher buffer underflow probability.

Figure 6.15 depicts the buffer optimization results, in comparison with the size of the jitter removal buffer (the fixed sized buffer can also be compared as always having a 5 second buffer size on the y-axis). With BROS, the initial buffer size is greatly reduced from fixed size, five second buffer to about one or two seconds. The jitter removal buffers are substantially smaller, consistently less than 0.5 seconds even for the largest jitter removal

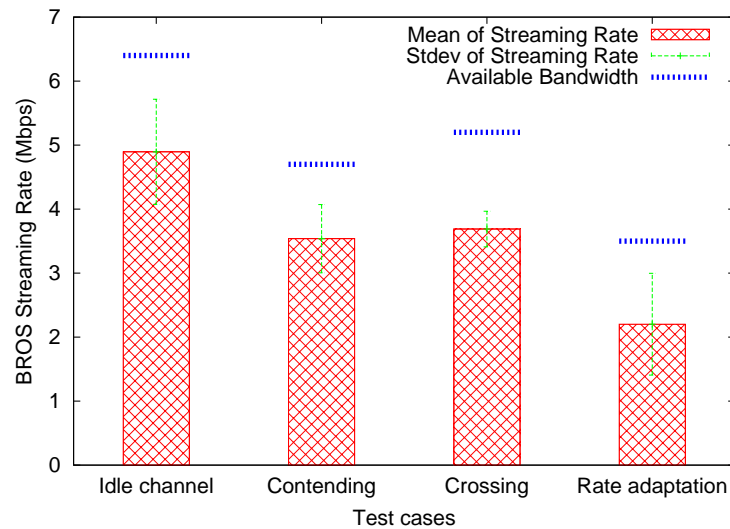


Figure 6.14: BROS Streaming Rate

buffer. While a small buffer is attractive for reducing the startup delay for playout, the quality is also impacted by the resulting impact on frame loss rate (see below). In addition, BROS adjusts the buffer based on the variance in the available bandwidth (see Table 6.2). Comparing the contending and crossing cases, even though the available bandwidth and the selected streaming are close to each other, the buffer sizes BROS chooses are different. The contending traffic case has a higher average buffer size than the crossing traffic case because of the higher variance in available bandwidth.

Figure 6.16-6.19 depict the results for the streaming quality metrics for each case, showing the averages over all sessions of the fraction of buffer underflows, frame loss rate, and buffer delay. By combining the optimization of the streaming rate selection and the initial buffer size, BROS performs better than the manual rate selection and jitter buffer approaches in terms of buffer underflow, frame loss, and buffer delay. As a quick visual summary, lower is better for all metrics depicted and BROS is the lowest or nearly the lowest for all graphs depicted.

The jitter buffer sessions always have a high fraction of buffer underflow events because of the small buffer sizes chosen. The fixed buffer size sessions with high (H) and medium (M) streaming rates also have high buffer underflow fractions since the streaming rates are

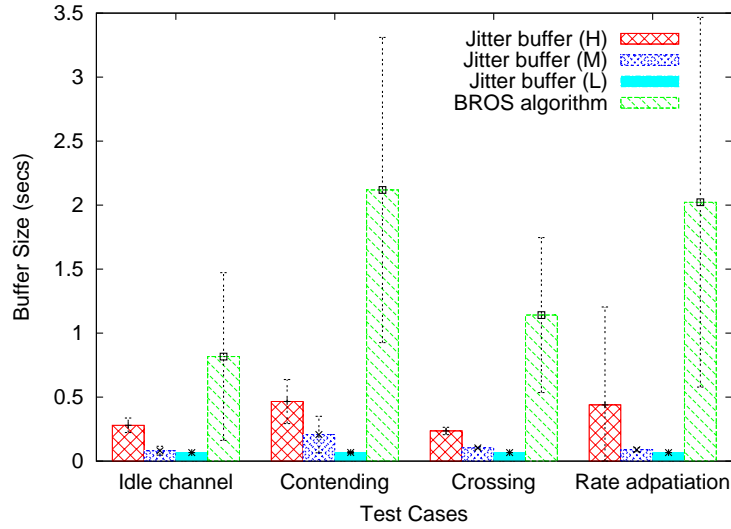


Figure 6.15: BROS Buffer Size

generally too high to be consistently sustained for the duration of the video session. The fixed buffer size sessions with high (H) streaming rates may have a lower buffer underflow fraction than these with medium (M) streaming rates. This is because the sessions with high (H) streaming rates take an extreme long time to fill up the playout buffer due to the high frame loss rate, thus resulting a short actual playout time and low buffer underflow fraction. While the buffer underflow fractions for the low streaming rate sessions (L) with fixed buffer size have a buffer underflow fraction close to that of BROS, BROS has a buffer delay reduced by more than 50%.

Moreover, BROS has much lower frame losses than all the sessions with high (H) and medium (M) streaming rate with both fixed size or jitter buffers. While the low (L) streaming rate sessions with fixed or jitter buffer have frame loss rates similar to BROS, BROS has a lower initial delay than the fixed buffer size sessions and a lower buffer underflow than the jitter buffer sessions.

In general, BROS significantly reduces the average buffer delay for the fixed buffer size and jitter buffers, for both the initial buffer delay and total buffer delay (the total buffer delay includes the initial buffer delay and the delay caused by rebuffer events). For instance, even though the jitter buffer sessions with high (H) and medium (M) streaming

rates usually have a small initial delay due to the small buffer size, the total buffer delays are high because of the large number of buffer underflow events in each session. The jitter buffer sessions with low (L) streaming rate has the lowest buffer delay. However, the buffer underflow fraction is unacceptable for most streaming media applications.

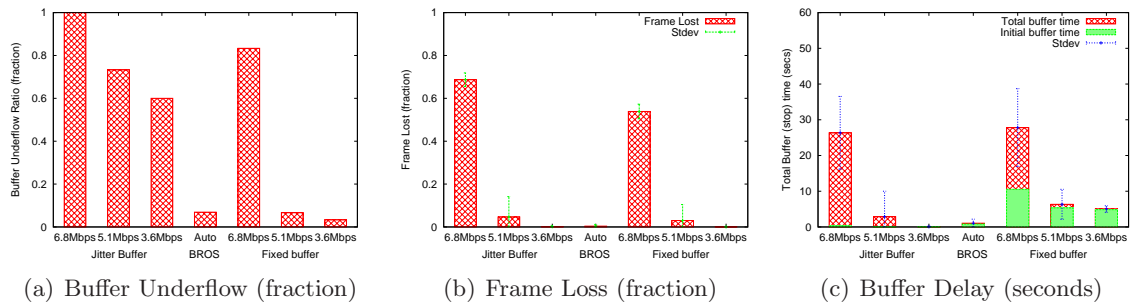


Figure 6.16: Summary Results for Evaluation Case 1 (Idle Channel)

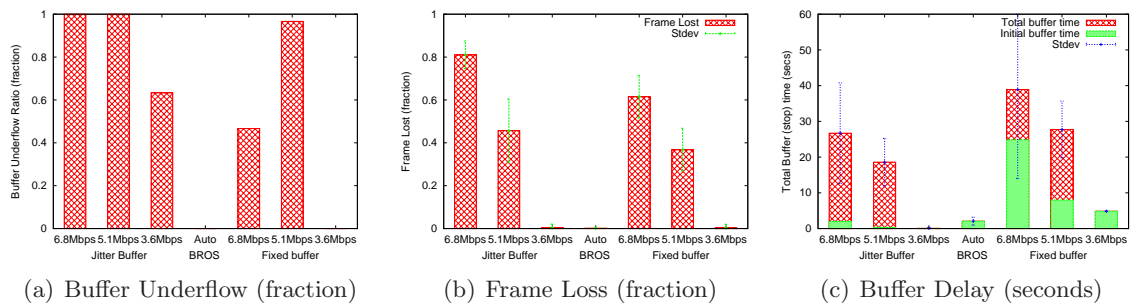


Figure 6.17: Summary Results for Evaluation Case 2 (Contending Traffic)

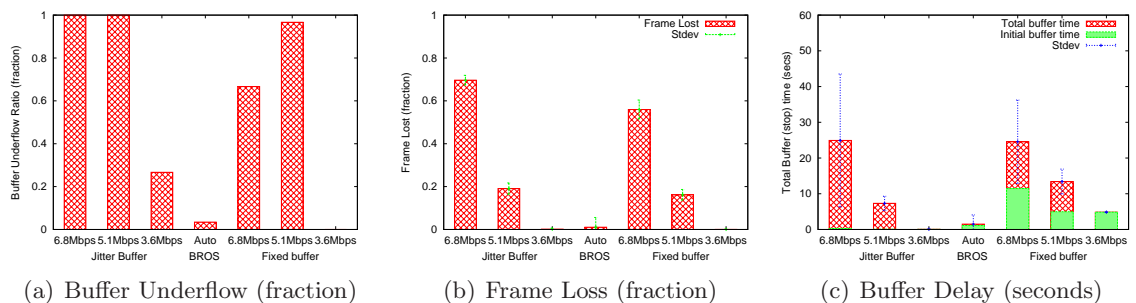


Figure 6.18: Summary Results for Evaluation Case 3 (Crossing Traffic)

Tables 6.4, 6.5 and 6.6 summarize the streaming quality results by comparing the averages for all the sessions in each case in terms of the buffer underflow fraction, frame

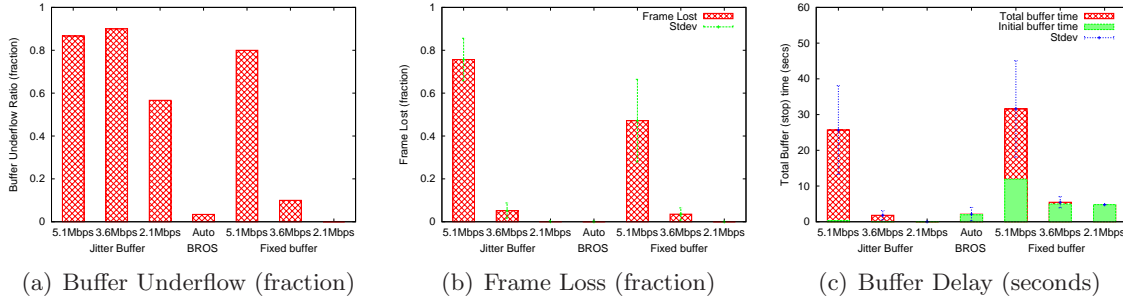


Figure 6.19: Summary Results for Evaluation Case 4 (Rate Adaptation)

loss percent, and total buffer delay. BROS shows the best overall performance for the quality metrics.

Table 6.4: Buffer Underflow (fraction)

		Case 1	Case 2	Case 3	Case 4
BROS		0.07	0	0.03	0.03
Fixed buffer	H	0.83	0.47	0.67	0.80
	M	0.07	1.00	1.00	0.10
	L	0.03	0	0	0
Jitter buffer	H	1.00	1.00	1.00	0.87
	M	0.73	1.00	1.00	0.90
	L	0.60	0.63	0.26	0.56

Table 6.5: Frame Loss Rate (percent)

		Case 1	Case 2	Case 3	Case 4
BROS		0.39%	0.16%	1.06%	0.05%
Fixed buffer	H	53.9%	61.5%	56.0%	47.2%
	M	3.00%	36.8%	16.2%	3.57%
	L	0.04%	0.32%	0.03%	0.01%
Jitter buffer	H	68.7%	81.0%	69.7%	75.7%
	M	4.66%	45.7%	19.1%	5.20%
	L	0.14%	0.37%	0.14%	0.09%

BROS significantly reduces the buffer underflow event, frame loss, and buffer delay by selecting the proper initial streaming rate and buffer size that corresponds with the available bandwidth information. For example, comparing the BROS algorithm with the medium (M) streaming rate for fixed and jitter removal buffer, BROS effectively reduces

Table 6.6: Buffer Delay (seconds)

		Case 1		Case 2		Case 3		Case 4	
		Init	Sum	Init	Sum	Init	Sum	Init	Sum
BROS		0.82	1.06	2.06	2.06	1.13	1.50	1.95	2.12
Fixed buffer	H	10.6	27.8	24.9	38.9	11.5	24.6	11.9	31.6
	M	5.33	6.34	7.97	27.7	4.97	13.4	4.97	5.47
	L	4.85	5.02	4.85	4.85	4.84	4.84	4.79	4.79
Jitter buffer	H	0.31	26.4	1.99	26.7	0.26	24.9	0.32	25.7
	M	0.05	2.89	0.31	18.6	0.07	7.31	0.05	1.81
	L	0.05	0.06	0.03	0.08	0.03	0.04	0.03	0.03

the buffer underflow fraction by 99% and 100%, the frame lost rates by 97% and 98%, and the total buffer delay by 87% and 78%, respectively.

## 6.6 Summary

This chapter proposes BROS, an algorithm designed to select the proper streaming rate and initial buffer size based on the available bandwidth estimations using WBest to reduce the buffer underflow events, buffer delay, and improve the frame loss rate for multimedia streaming application over wireless networks. A core contribution is a model of the client side initial buffer size for streaming multimedia applications, as the function of streaming rate and the distribution of available bandwidth in the wireless networks. One advantage of the buffer model over existing jitter or Poisson arrival models is that it considers changes in available bandwidth, which typically have a larger impact on streaming performance than does inter-arrival jitter. For evaluation, BROS is implemented in a streaming system (EmuS), and compared with approaches with a fixed streaming rate and buffer sizes, and jitter removal buffers, in a wireless testbed with a variety of network setups. The following conclusions can be drawn:

1. Existing buffer models that consider only the impact of jitter are not adequate to remove the affects of changes in available bandwidth in IEEE 802.11 networks. Moreover, the assumption of constant average arrival rate in previous research is not appli-

cable in an environment with considerable changes in available bandwidth, such as for IEEE 802.11 wireless networks. This make the current buffer models under-predict the buffer size needed to avoid buffer underflow in wireless networks.

2. Performance of multimedia streaming applications is significantly impacted by the initial streaming rate and buffer size. With BROS, the streaming rate and buffer size can be optimized according to the current available bandwidth conditions, thus resulting in better performance.
3. BROS uses WBest algorithm to estimate the mean and standard deviations of available bandwidth to infer the available bandwidth distribution. However, BROS is flexible enough to be used with other bandwidth estimation tools that provide similar bandwidth information. This makes it easy to improve the bandwidth estimation techniques independently of the BROS algorithm.

Overall BROS can significantly reduce buffer underflows, frame losses and buffer delays by optimizing the initial streaming rate and buffer size.

## Chapter 7

# Future Work

This chapter presents some possible future work that can be extended from this dissertation.

- Even though WBest is developed specially for IEEE 802.11b/g infrastructure networks, WBest can be extended to other types of wireless networks such as WWANs using CDMA or GPRS techniques. For instance, the ARQ and FEC approaches in the GPRS Radio Link Control (RLC) layer adapt the channel capacity and available bandwidth and may cause streaming applications to suffer from performance degradation. WBest would need to be adapted, such as modifying the number of packet pairs or the length of the packet train, based on further analytical research and empirical study before it can be applied to these types of wireless networks.
- WBest is designed for IEEE 802.11b/g Distributed Coordination Function (DCF) wireless networks. WBest can be further extended to either the optional Point Coordination Function (PCF), which is a centralized MAC protocol that uses a point coordinator to determine which node has the right to transmit, or to the developing IEEE 802.11e wireless network standard. In the PCF model or IEEE 802.11e standard, the AP queue does not follow a strict FCFS policy. Therefore the packet dispersion behavior may be different from that of DCF in IEEE 802.11b/g wireless networks. WBest would need to be adapted based on related modeling and study of non-FCFS behavior in these types of wireless networks.

- The number of packet pairs and the length of packet train is empirically decided by AP queue sizes and measurements. One possible future work may include designing the algorithm to optimize the number of packet pairs and length of the packet train based on the network conditions. For example, the packet train length in the second step of WBest can be decided by the effective capacity measured from the first step, or the expected accuracy, convergence time or intrusiveness requested by the application.
- WBest and BROS are designed for pre-recorded streaming multimedia applications, which usually tolerant a long playout delay. However, WBest and BROS can be further tuned to reduce the cost in terms of convergence time and intrusiveness, thus make it applicable to interactive streaming applications, such as for video conference or Internet TV service, where users may switch streams frequently. The tuning would include the number of packet pairs and the length of the packet train of WBest, and startup parameters of BROS, such as the client tolerable buffer delay and the target Mean Time Between Buffer Underflow (MTBBU).
- Currently, WBest does not explicitly report loss rate and delay information. WBest can be improved to report detailed information about the loss and delay measured by WBest. For example, with a Loss Discrimination Algorithm (LDA), WBest can have the capability of reporting wired and wireless loss rate to the application, allowing the application to employ more effective repair techniques.
- In addition to the improvement of WBest with loss and delay measurements, BROS can be further improved to optimize application layer media repair approaches, such as Forward Error Correction (FEC) and streaming packet retransmissions. For example, the available bandwidth information can be used to decide the level of FEC or number of streaming packet retries. For the streaming sessions that are constrained by available bandwidth, limiting the amount of FEC and retransmission traffic can reduce the packet delay or loss caused by media repair procedures.

- 
- Currently, BROS suggests not to stream when WBest reports no available bandwidth in the wireless network. However, since the wireless network is a contention domain, streaming applications can still get the throughput of a fair share of the effective capacity. Therefore, instead of suggesting not to stream, BROS could be further improved to decide the streaming rate inferred as a function of achievable throughput, or the fair share of the effective capacity, perhaps informed by cross-layer knowledge of the number of contending nodes in the wireless network.
  - Strong interference or mobility may still impact the available bandwidth during streaming sessions, thus resulting in unexpected buffer underflow events. Therefore, to further improve streaming rate selection and buffer optimization for the entire streaming sessions in an unsteady environment, BROS can be applied periodically during playback or at each rebuffer event. One possible area of future work would be to use the streaming multimedia data to estimate available bandwidth during the session. This can reduce the traffic overhead caused by WBest probing.



## Chapter 8

# Conclusions

This dissertation presents an application layer solution for improving streaming multimedia application performance in IEEE 802.11 wireless networks by using enhanced bandwidth estimation techniques. The solution includes two parts: 1) a new Wireless Bandwidth estimation tool (WBest) designed for fast, non-intrusive, accurate estimation of available bandwidth in IEEE 802.11 networks, which can be used by streaming multimedia applications to improve the performance in wireless networks; 2) a Buffer and Rate Optimization for Streaming (BROS) algorithm using WBest to guide the streaming rate selection and initial buffer optimization. With WBest and BROS, the performance of streaming multimedia applications in wireless networks can be significantly improved in terms of frame loss, rebuffer events and buffer delay. This chapter summarizes the major contributions and draws conclusions from the dissertation research.

Packet dispersion techniques have been commonly used to estimate bandwidth in wired networks. However, current packet dispersion techniques were developed for wired network environments and can provide inaccurate results in wireless networks due to the variability in wireless capacity over short time scales. To enhance wired bandwidth estimation techniques, Chapter 4 presents the in-depth study of the packet dispersion techniques in IEEE 802.11 wireless networks. We develop an analytical model to investigate packet dispersion behavior in wireless networks. The packet dispersion model is validated using

both an extended NS-2 simulator that includes 802.11 MAC layer rate adaptation and wireless 802.11b testbed measurements. Additionally, mean and variance of packet dispersion in IEEE 802.11 wireless networks is analyzed while considering the impact of channel conditions such as packet size, link rate, bit error rate and RTS/CTS.

Based on the packet dispersion model and analysis, Chapter 5 presents a new Wireless Bandwidth estimation tool (WBest) designed for fast, non-intrusive, accurate estimation of available bandwidth in IEEE 802.11 networks. WBest applies a two-step algorithm: 1) a packet pair technique to estimate the effective capacity of the wireless networks; 2) a packet train technique to estimate the achievable throughput and report the inferred available bandwidth. Using an analytic model, the possible error sources are explored and WBest parameters are optimized given the tradeoffs of accuracy, intrusiveness and convergence time. The advantage of WBest is that it does not depend upon search algorithms to detect the available bandwidth but instead, statistically detects the available fraction of the effective capacity, mitigating estimation delay and the impact of random wireless channel errors. WBest is compared with other popular available bandwidth estimation tools in a wireless testbed under a variety of wireless and network conditions. The evaluation shows that current bandwidth estimation tools are significantly impacted by wireless network conditions, such as contention from other traffic and rate adaptation. On the other hand, WBest consistently provides fast available bandwidth estimation, with overall more accurate estimations and lower intrusiveness over all conditions evaluated. Moreover, WBest provides a broad range of bandwidth information for the wireless networks, such as the effective capacity, available bandwidth, achievable throughput and variance of available bandwidth and achievable throughput. Thus, WBest demonstrates the potential for improving the performance of applications that need bandwidth estimation, such as multimedia streaming, on wireless networks.

To use the bandwidth related information provided by WBest, Chapter 6 develops a new buffer model to investigate the relationship of buffer size, streaming data rate and available bandwidth distribution. One advantage of our buffer model over existing jitter

---

or Poisson arrival models is that it takes available bandwidth changes into consideration, which usually have greater impact on streaming performance than fluctuation of inter-arrival times. Based on this new buffer model, Chapter 6 presents the Buffer and Rate Optimization for Streaming (BROS) algorithm to improve streaming multimedia application performance, such as the frame loss, buffer underflow events, and initial delay in wireless networks. BROS optimizes the streaming data rate and initial buffer size to reduce frame losses, buffer underflow events, with minimized initial buffer delay. BROS is implemented in an emulated streaming system, called Emulated Streaming (EmuS), and evaluated in an IEEE 802.11 wireless testbed with various wireless conditions. The evaluation shows that BROS can effectively select the best streaming rate and optimize the initial buffer size based on wireless network bandwidth conditions, thus achieving lower frame loss rate, fewer buffer underflow events and lower initial delay than static rate selection, static buffer sizing, and jitter removal buffers.

Based on the summary of this dissertation, the following conclusion can be drawn:

- Packet dispersion measures the *effective capacity* and the *achievable throughput* of a wireless network instead of the capacity as in a wired network. Effective capacity, defined as a function of packet size and time, represents the ability of a wireless network to forward data over a given time period. Achievable throughput is the maximum throughput that a node can achieve when contending with other existing traffic on a wireless network.
- Wireless channel conditions, such as packet sizes, link rate, Bit Error Rate (BER) and RTS/CTS impact the bandwidth estimation results and the variance of the results. The packet size and link rate have positive correlations with both the bandwidth estimations and variances of the estimations. The BER of the channel has a negative correlation with the bandwidth estimations and a positive correlation with variance of the estimations. RTS/CTS reduces estimated bandwidth and the variance of the estimations.

- Current bandwidth estimation tools are significantly impacted by wireless network conditions, such as contention from other traffic and rate adaptation. This results in inaccurate estimates and high and varying convergence times and intrusiveness. This makes current tools generally impractical for applications running over a wireless link, such as streaming media, that require fast, accurate, non-intrusive bandwidth estimates.
- WBest consistently provides fast available bandwidth estimation, with overall more accurate estimations and lower intrusiveness over all conditions evaluated. To provide effective bandwidth related information, WBest should be configured to use the same packet size as the application.
- Existing streaming buffer models that consider only the impact of jitter are not adequate to remove the effects of changes in available bandwidth in IEEE 802.11 networks. Moreover, the assumption of constant average arrival rate in previous research is not applicable in environments with considerable changes in available bandwidth, such as for IEEE 802.11 wireless networks. This makes the current buffer models under-predict the buffer needed to avoid buffer underflow events in wireless networks.
- By taking the available bandwidth fluctuation of wireless networks into consideration, our streaming buffer model can effectively predict the minimum buffer required for a given streaming rate, or decide the optimal streaming rate with a given streaming buffer size to mitigate buffer underflow events under a variety of wireless network conditions. Performance of multimedia streaming applications is significantly impacted by the initial streaming rate and buffer size. With BROS, the streaming rate and buffer size can be optimized according to the current available bandwidth conditions, thus resulting in improved frame loss rate, buffer underflow events, and buffer delay.

In conclusion, this dissertation presents an application layer solution for improving streaming multimedia application performance in IEEE 802.11 wireless networks by using

---

enhanced bandwidth estimation techniques. Using analytical models, simulations and network measurements, this dissertation shows that our application layer solution, consisting of WBest and BROS, can effectively improve the streaming multimedia performance in terms of frame losses, rebuffer events and buffer delay under a variety of wireless network conditions.



# Bibliography

- [1] Bill Birney, “Reducing Start-up Latency with Windows Media 9 Series,” Feb. 2004, Microsoft Document. Online at:  
<http://www.microsoft.com/windows/windowsmedia/howto/articles/reducingstartuplatency.aspx>. 2, 15, 32, 39
- [2] James Nichols, Mark Claypool, Robert Kinicki, and Mingzhe Li, “Measurements of the Congestion Responsiveness of Windows Streaming Media,” in *Proceedings of the 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Kinsale, County Cork, Ireland, June 2004. 2, 14, 15, 19
- [3] Paul Bocheck, Andrew Campbell, Shih-Fu Chang, and Raymond Lio, “Utility-based Network Adaptation for MPEG-4 Systems,” in *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Basking Ridge, NJ, USA, June 1999. 2
- [4] Avanish Tripathi and Mark Claypool, “Improving Multimedia Streaming with Content-Aware Video Scaling,” in *Workshop on Intelligent Multimedia Computing and Networking (IMMCN)*, Durham, North Carolina, USA, Mar. 2002. 2
- [5] Kang Li, Charles Krasic, Jonathan Walpole, Molly H. Shor, and Calton Pu, “The Minimal Buffering Requirements of Congestion Controlled Interactive Multimedia Applications,” in *IDMS '01: Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems*. 2001, pp. 181–192, Springer-Verlag. 2, 32, 33
- [6] Tianbo Kuang and Carey Williamson, “RealMedia Streaming Performance on an IEEE 802.11b Wireless LAN,” in *Proceedings of IASTED Wireless and Optical Communications (WOC)*, July 2002, pp. 306–311. 2, 35, 63
- [7] Tianbo Kuang and Carey L. Williamson, “Hierarchical Analysis of RealMedia Streaming Traffic on an IEEE 802.11b Wireless LAN,” *Computer Communications*, vol. 27, no. 6, pp. 538–548, Aug. 2004. 2, 35, 37
- [8] Feng Li, Jae Chung, Mingzhe Li, Huahui Wu, Mark Claypool, and Robert Kinicki, “Application, Network and Link Layer Measurements of Streaming Video over a Wireless Campus Network,” in *Proceedings of the 6th Passive and Active Measurement Workshop (PAM)*, Boston, Massachusetts, USA, Apr. 2005. 2, 3, 36, 37, 62

- [9] Guangwei Bai and Carey Williamson, “The Effects of Mobility on Wireless Media Streaming Performance,” in *Proceedings of Wireless Networks and Emerging Technologies (WNET)*, July 2004, pp. 596–601. [2](#), [36](#), [37](#), [63](#)
- [10] Mingzhe Li, Feng Li, Mark Claypool, and Robert Kinicki, “Weather Forecasting - Predicting Performance for Streaming Video over Wireless LANs,” in *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Skamania, Washington, USA, June 2005. [2](#), [37](#), [98](#)
- [11] IEEE Computer Society LAN MAN Standard Committee, “IEEE 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” Standard, Aug. 1999. [2](#), [25](#), [67](#), [74](#), [78](#)
- [12] Manthos Kazantzidis and Mario Gerla, “The Impact of Link Layer Assisted Multimedia Adaptation in Wireless Network,” in *ITRE2003*, Aug. 2003, pp. 326–330. [3](#), [38](#)
- [13] Mingzhe Li, Emmanuel Agu, Mark Claypool, and Robert Kinicki, “Performance Enhancement of TFRC in Wireless Networks,” in *Proceedings of the 10th International Conference on Distributed Multimedia Systems (DMS)*, San Francisco, California, USA, Sept. 2004. [3](#), [20](#), [38](#)
- [14] Amoolya Singh, Almudena Konrad, and Anthony D. Joseph, “Performance Evaluation of UDP Lite for Cellular Video,” in *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Port Jefferson, New York, USA, 2001, pp. 117–124, ACM Press. [3](#)
- [15] M. Chen and A. Zakhor, “Rate Control for Streaming Video over Wireless,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Hongkong, China, Mar. 2004. [3](#), [38](#)
- [16] Santhana Krishnamachari, Mihaela VanderSchaar, Sunghyun Choi, and X. Xu, “Video Streaming over Wireless LANs: A Cross-layer Approach,” in *Proceedings of International Packet Video Workshop (PV)*, Nantes, France, Apr. 2003. [3](#), [39](#)
- [17] R. Kapoor, M. Cesana, and M. Gerla, “Link Layer Support for Streaming MPEG Video over Wireless Links,” in *Proceedings of the 12th International Conference on Computer Communications and Networks, Computer Communications and Networks (ICCCN)*, Dallas, TX, USA, Oct. 2003, pp. 477–482. [3](#), [39](#)
- [18] J. Chesterfield, R. Chakravorty, J. Crowcroft, P. Rodriguez, and S. Banerjee, “Experiences With Multimedia Streaming over 2.5G and 3G Networks,” in *Proceedings of BROADNETS*, San Jose, California, USA, Oct. 2004. [3](#)
- [19] Julian Chesterfield, Rajiv Chakravorty, Suman Banerjee, Pablo Rodriguez, Ian Pratt, and Jon Crowcroft, “Transport Level Optimisations for Streaming Media over Wide-area Wireless Networks,” *ACM MONET Journal*, June 2004. [3](#)

- 
- [20] Gavin Holland, Nitin H. Vaidya, and Paramvir Bahl, “A Rate-adaptive MAC Protocol for Multi-Hop Wireless Networks,” in *MobiCom '01: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 236–251. vi, 8, 28, 29, 60, 67
- [21] Mingzhe Li, Mark Claypool, Robert Kinicki, and James Nichols, “Characteristics of Streaming Media Stored on the Web,” Tech. Rep. WPI-CS-TR-03-18, CS Department, Worcester Polytechnic Institute, May 2003. 12
- [22] Microsoft Corporation, “Windows Media Services 9 Series, How Protocol Rollover Works,” 2003, Microsoft Document. Online at: <http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/wmsserver/howprotocolrolloverworks.asp>. 12
- [23] Bill Birney, “Microsoft Corporation - Reducing Broadcast Delay,” Apr. 2003, Microsoft Document. Online at: <http://www.microsoft.com/windows/windowsmedia/howto/articles/BroadcastDelay.aspx>. 13, 14
- [24] Mingzhe Li, Mark Claypool, and Robert Kinicki, “MediaPlayer versus RealPlayer – A Comparison of Network Turbulence,” in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*, Marseille, France, Nov. 2002, pp. 131 – 136. 14
- [25] Inc. Apple Computer, “QuickTime Streaming End-to-end Solutions for Live Broadcasting and On-demand Streaming of Digital Media.,” 2003, Online at: [http://images.apple.com/server/pdfs/L31754A\\_QTStreaming\\_TB\\_final.pdf](http://images.apple.com/server/pdfs/L31754A_QTStreaming_TB_final.pdf). 14
- [26] Bill Birney, “Microsoft Corporation - Intelligent Streaming,” Oct. 2000, Microsoft Document. Online at: <http://msdn.microsoft.com/library/en-us/dnwmt/html/intstreaming.asp>. 14, 15
- [27] Jae Chung, Mark Claypool, and Yali Zhu, “Measurement of the Congestion Responsiveness of RealPlayer Streaming Video Over UDP,” in *Proceedings of the Packet Video Workshop (PV)*, Nantes, France, Apr. 2003. 15, 19
- [28] Luca Delgrossi, Christian Halstrick, Dietmar Hehmann, Ralf Guido Herrtwich, Oliver Krone, Jochen Sandvoss, and Carsten Vogt, “Media Scaling for Audiovisual Communication with the Heidelberg Transport System,” in *MULTIMEDIA '93: Proceedings of the First ACM International Conference on Multimedia*, Anaheim, California, USA, 1993, pp. 99–104. 16, 18
- [29] H. Katata, N. Ito, and H. Kusao, “Temporal-scalable Coding Based on Image Content,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 52–59, Feb. 1997. 16
- [30] C. Podilchuk, N. Jayant, and N. Farvardin, “Three Dimensional Subband Coding of Video,” *IEEE Transactions on Image Processing*, vol. 4, pp. 125–139, Feb. 1995. 16

- [31] D. Taubman and A. Zakhor, "Multirate 3-D Subband Coding of Video," *IEEE Transactions on Image Processing*, vol. 3, pp. 572–588, Sept. 1994. [16](#)
- [32] G. J. Conklin and S. S. Hemami, "A Comparison of Temporal Scalability Techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 909–919, Sept. 1999. [16](#)
- [33] T. Naveen and J. W. Woods, "Motion Compensated Multiresolution Transmission of High Definition Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, pp. 29–41, Feb. 1994. [16](#)
- [34] U. Benzler, "Spatial Scalable Video Coding Using a Combined Subband-DCP Approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1080–1087, Oct. 2000. [16](#)
- [35] N. Feamster, D. Bansal, and H. Balakrishnan, "On the Interactions between Layered Quality Adaptation and Congestion Control for Streaming Video," in *Proceedings of International Packet Video Workshop (PV)*, Kyongju, Korea, Apr. 2001. [16](#), [17](#)
- [36] Reza Rejaie, Mark Handley, and Deborah Estrin, "Quality Adaptation for Congestion Controlled Video Playback over the Internet," in *SIGCOMM '99: Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Cambridge, MA, Aug. 1999, pp. 189–200. [16](#), [17](#)
- [37] M. Miyabayashi, N. Wakamiya, M. Murata, and H. Miyahara, "MPEG-TFRCP: Video Transfer with TCP-Friendly Rate Control Protocol," in *Proceedings of IEEE International Conference on Communications (ICC)*, St. Petersburg, Russia, 2001, pp. 137–141. [17](#)
- [38] M. Domanski, A. Luczak, and S. Mackowiak, "Spatial-temporal Scalability for MPEG Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1088–1093, Oct. 2000. [17](#)
- [39] T. Talley and K. Jeffay, "Two-dimensional Scaling Techniques for Adaptive, Rate-based Transmission Control of Live Audio and Video Streams," in *MULTIMEDIA '94: Proceedings of the Second ACM International Conference on Multimedia*, San Francisco, California, USA, 1994, pp. 247–254. [17](#)
- [40] Sally Floyd, Mark Handley, Jitendra Padhye, and Jorg Widmer, "Equation-Based Congestion Control for Unicast Applications," in *Proceedings of ACM SIGCOMM Conference*, Stockholm, Sweden, Aug. 2000, pp. 43 – 56. [17](#)
- [41] Nadeem Aboobaker, David Chanady, Mario Gerla, and M. Y. Sanadidi, "Streaming Media Congestion Control Using Bandwidth Estimation," in *MMNS '02: Proceedings of the IFIP/IEEE International Conference on Management of Multimedia Networks and Services*. 2002, pp. 89–100, Springer-Verlag. [18](#)
- [42] Ren Wang, Massimo Valla, M. Y. Sanadidi, and Mario Gerla, "Adaptive Bandwidth Share Estimation in TCP Westwood," in *Proceedings of IEEE Globecom*, Taipei, Taiwan, R.O.C., Nov. 2002. [18](#)

- 
- [43] Philippe de Cuetos and Keith W. Ross, “Adaptive Rate Control for Streaming Stored Rine-grained Scalable Video,” in *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Miami, Florida, USA, 2002, pp. 3–12, ACM Press. 18
- [44] Cheng Huang, Philip A. Chou, and Anders Klemets, “Optimal Coding Rate Control for Scalable Streaming Media,” in *Proceedings of International Packet Video Workshop (PV)*, Irvine, CA, Dec. 2004. 18
- [45] Wu-chi Feng, “On the Efficacy of Quality, Frame Rate, and Buffer Management for Video Streaming Across Best-effort Networks,” *Journal of High Speed Networks*, vol. 11, pp. 199–214, Nov. 2002. 18
- [46] Silvia Hollfelder, Florian Schmidt, Matthias Hemmje, and Karl Aberer, “Transparent Integration of Continuous Media Support into a Multimedia DBMS,” in *Issues and Applications of Database Technology*, 1998, pp. 192–199. 19
- [47] Zhenghua Fu, Petros Zerfos, Haiyun Luo, Songwu Lu, Lixia Zhang, and Mario Gerla, “The Impact of Multihop Wireless Channel on TCP Throughput and Loss,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, Mar. 2003. 20
- [48] Zhiheng Wang, Sujata Banerjee, and Sugih Jamin, “Studying Streaming Video quality: from an Application Point of View,” in *Proceedings of the 11th ACM International Conference on Multimedia*, Berkeley, CA, USA, 2003, pp. 327–330, ACM Press. 21
- [49] T. Kim and M. Ammar, “Optimal Quality Adaptation for MPEG-4 Fine-Grained Scalable Video,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco CA, USA, Apr. 2003. 21
- [50] M. Masry and S. Hemami, “An Analysis of Subjective Quality in Low Bit Rate Video,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Thesaloniki, Greece, 2001, pp. 465–468. 21
- [51] Microsoft Corporation, “Windows Media Player 10 SDK,” 2004, Microsoft Document. Online at:  
[http://www.msdn.microsoft.com/library/en-us/wmplay10/mmp\\_sdk/windowsmedia\\_player10sdk.asp](http://www.msdn.microsoft.com/library/en-us/wmplay10/mmp_sdk/windowsmedia_player10sdk.asp). 21
- [52] Kaveh Pahlavan and Prashant Krishnamurthy, *Principles of Wireless Networks - A Unified Approach*, chapter 2-4, Prentice Hall PTR, 2002. 22
- [53] Intel Corporation, “White Paper: The New Era in Communications,” 2004, Online at: <http://www.intel.com/netcomms/bbw/302026.htm>. 24
- [54] Marcelo M. Carvalho and J. J. Garcia-Luna-Aceves, “Delay Analysis of IEEE 802.11 in Single-Hop Networks,” in *ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols*, Washington, DC, USA, 2003, p. 146, IEEE Computer Society. 26, 58, 60, 75

- [55] Ivaylo Haratcherev, Jacco Taal, Koen Langendoen, Reginald Legendijk, and Henk Sips, “Automatic IEEE 802.11 Rate Control for Streaming Applications: Research Articles,” *Wireless Communications and Mobile Computing*, vol. 5, no. 4, pp. 421–437, 2005. [29](#), [67](#)
- [56] A. Kamerman and L. Monteban, “WaveLAN II: A Highperformance Wireless LAN for Unlicensed Band,” in *Bell Labs Technical Journal*, 1997. [29](#), [60](#), [67](#)
- [57] Stephen Jacobs and Alexandros Eleftheriadis, “Providing Video Services over Networks without Quality of Service Guarantees,” in *Proceedings of WWW Consortium Workshop on Real-Time Multimedia and the Web*, Sophia Antipolis, France, Oct. 1996. [31](#)
- [58] Wai-tian Tan, Weidong Cui, and John Apostolopoulos, “Playback-Buffer Equalization for Streaming Media using Stateless Transport Prioritization (Abstract),” in *Proceedings of International Packet Video Workshop (PV)*, Nantes, France, Apr. 2003. [32](#)
- [59] Nagasuresh Seelam, Pankaj Sethi, and Wu-chi Feng, “A Hysteresis Based Approach for Quality, Frame Rate, and Buffer Management for Video Streaming Using TCP,” in *MMNS '01: Proceedings of the IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, 2001, pp. 1–15. [32](#)
- [60] M.C. Yuang, S.T. Liang, Yu.G. Chen, and C.L. Shen, “Dynamic Video Playout Smoothing Method for Multimedia Applications,” in *Proceedings of IEEE International Conference on Communications (ICC)*, Dallas, TX, USA, June 1996, vol. 3, pp. 1365–1369. [32](#), [34](#), [151](#)
- [61] Bernd Girod, Niko Färber, and Eckehard G. Steinbach, “Adaptive Playout for Low Latency Video Streaming,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Thessaloniki, Greece, 2001, pp. 962–965. [32](#), [34](#), [151](#)
- [62] Nikolaos Laoutaris and Ioannis Stavrakakis, “Adaptive Playout Strategies for Packet Video Receivers with Finitebuffer Capacity,” in *Proceedings of IEEE International Conference on Communications (ICC)*, Helsinki, Finland, 2001. [32](#), [34](#), [151](#)
- [63] M. Kalman, S.T. Eckehard, and B. Girod, “Adaptive Playout for Real-time Media Streaming,” in *IEEE International Symposium on Circuits and Systems*, 2002, vol. 1. [32](#)
- [64] Mark Kalman, Eckehard G. Steinbach, and Bernd Girod, “Adaptive Media Playout for Low-delay Video Streaming over Error-prone Channels,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 841–851, 2004. [32](#), [34](#), [151](#), [155](#)
- [65] E. Steinbach, N. Faerber, and B. Girod, “Adaptive Playout for Low-Latency Video Streaming,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Thessaloniki, Greece, Oct. 2001, pp. 962–965. [32](#)

- 
- [66] Y. Li, A. Markopoulou, N. Bambos, and J. Apostolopoulos, “Joint Power/Playout Control Schemes for Media Streaming over Wireless Links,” in *Proceedings of International Packet Video Workshop (PV)*, Irvine, CA, USA, Dec. 2004. [32](#), [39](#)
- [67] W. Feng and J. Rexford, “Performance Evaluation of Smoothing Algorithms for the Transmission of Pre-recorded Video,” *IEEE Transactions on Multimedia*, vol. 1, no. 3, Sept. 1999. [32](#)
- [68] Roger Zimmermann, Kun Fu, Cyrus Shahabi, Didi Shu-Yuen Yao, and Hong Zhu, “Yima: Design and Evaluation of a Streaming Media System for Residential Broadband Services,” in *DBTel '01: Proceedings of the VLDB International Workshop on Databases in Telecommunications II*, Roma, Italy, 2001, pp. 116–125, Springer-Verlag. [33](#)
- [69] P. Mundur, A. Sood, and R. Simon, “Network Delay Jitter and Client Buffer Requirements in Distributed Video-on-Demand Systems,” Tech. Rep., Department of Computer Science at George Mason University, Fairfax, VA 22030. [34](#), [169](#)
- [70] Ramachandran Ramjee, Jim Kurose, Don Towsley, and Henning Schulzrinne, “Adaptive Playout Mechanism for Packetized Audio Applications in Wide-area Networks,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Los Alamitos, CA, USA, June 1994, pp. 680–688. [34](#)
- [71] Sue B. Moon, Jim Kurose, and Don Towsley, “Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms,” *Multimedia Syst.*, vol. 6, no. 1, pp. 17–28, 1998. [34](#)
- [72] Kouhei Fujimoto, Shingo Ata, and Masayuki Murata, “Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications,” in *IEEE GLOBECOM*, Nov. 2002, vol. 3, pp. 2451–2457. [34](#)
- [73] Maria C. Yuang, Po L. Tien, and Shih T. Liang, “Intelligent Video Smoother for Multimedia Communications,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 2, pp. 136–146, 1997. [34](#), [151](#)
- [74] Wu-chi Feng and Stuart Sechrest, “Smoothing and Buffering for Delivery of Pre-recorded Compressed Video,” in *Proceedings of IST/SPIE Multimedia Networking and Computing 1995*, San Jose, CA, Feb. 1995, pp. 234–242. [34](#)
- [75] James D. Salehi, Zhi-Li Zhang, James F. Kurose, and Don Towsley, “Supporting Stored Video: Reducing Rate Variability and End-to-end Resource Requirements Through Optimal Smoothing,” in *SIGMETRICS '96: Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Philadelphia, Pennsylvania, USA, 1996, pp. 222–231, ACM Press. [34](#)
- [76] Roger Zimmermann, Cyrus Shahabi, Kun Fu, and Mehrdad Jahangiri, “A Multi-threshold Online Smoothing Technique for Variable Rate Multimedia Streams,” *Multimedia Tools and Applications*, vol. 28, no. 1, pp. 23–49, 2006. [34](#)

- [77] Hrvoje Jenkac, Thomas Stockhammer, and Gabriel Kuhn, “On Video Streaming over Variable Bit-Rate and Wireless Channels,” in *Proceedings of International Packet Video Workshop (PV)*, Nantes, France, Apr. 2003. 35
- [78] P. Ikkurthy and M. Labrador, “Characterization of MPEG-4 Traffic over IEEE 802.11b Wireless LANs,” in *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, Tampa, Florida, USA, Nov. 2002, pp. 421–430. 36
- [79] Song Cen, Pamela C. Cosman, and Geoffrey M. Voelker, “End-to-end Differentiation of Congestion and Wireless Losses,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 703–717, 2003. 37
- [80] Minghua Chen and Avideh Zakhor, “Rate Control for Streaming Video over Wireless,” *Wireless Communications, IEEE*, vol. 12, no. 4, pp. 32–41, aug 2005. 38
- [81] Guang Yang, Mario Gerla, and M. Y. Sanadidi, “Adaptive Video Streaming in Presence of Wireless Errors,” in *MMNS '04: Proceedings of the IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, San Diego, CA, 2004. 38
- [82] Chi-Yuan Hsu, A. Ortega, and M. Khansari, “Rate Control for Robust Video Transmission over Burst-error Wireless Channels,” *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 5, pp. 756–773, may 1999. 38
- [83] Qiong Li and Mihaela van der Schaar, “Providing Adaptive QoS to Layered Video over Wireless Local Area Networks through Real-time Retry Limit Adaptation,” *Multimedia, IEEE Transactions on*, vol. 6, no. 2, pp. 278–290, Apr. 2004. 38, 39
- [84] Igor Kozintsev and Jeff Mc. Veigh, “Improving Last-Hop Multicast Streaming Video over 802.11,” in *Proceedings of BROADNETS*, San Jose, California, USA, Oct. 2004. 38
- [85] Fan Yang, Qian Zhang, Wenwu Zhu, and Ya-Qin Zhang, “End-to-end TCP-friendly Streaming Protocol and Bit Allocation for Scalable Video over Wireless Internet,” *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 4, pp. 777–790, May 2004. 38
- [86] Peter van Beek, Sachin Deshpande, Hao Pan, and Ibrahim Sezan, “Adaptive streaming of high-quality video over wireless LANs,” in *Visual Communications and Image Processing 2004*, San Jose, CA, USA, 2004, pp. 647–660. 39
- [87] Mehmet U. Demircin and Peter van Beek, “Bandwidth Estimation and Robust Video Streaming over 802.11E Wireless LANs ,” in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, July 2005. 39
- [88] Peter van Beek and Mehmet U. Demircin, “Delay-constrained Rate Adaptation for Robust Video Transmission over Home Networks,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Genoa, Italy, Sept. 2005. 39

- 
- [89] Mingzhe Li, Mark Claypool, and Robert Kinicki, “Wbest: a bandwidth estimation tool for ieee 802.11 wireless networks,” Tech. Rep. WPI-CS-TR-06-14, Computer Science Department at Worcester Polytechnic Institute, July 2006. 39
- [90] Karthik Lakshminarayanan, Venkata N. Padmanabhan, and Jitendra Padhye, “Bandwidth Estimation in Broadband Access networks,” in *IMC '04: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, Taormina, Sicily, Italy, Oct. 2004, pp. 314–321. 39, 66, 94, 95, 114, 115, 131
- [91] L. Angrisani, A. Botta, A. Pescape, and M. Vadursi, “Measuring Wireless Links Capacity,” *Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 1–5, Jan. 2006. 39, 94
- [92] R.S. Prasad, M. Murray, C. Dovrolis, and K.C. Claffy, “Bandwidth Estimation: Metrics, Measurement Techniques, and Tools,” *IEEE Network*, November-December 2003. 40, 41, 42, 43, 45, 46, 47, 48, 52, 108
- [93] M. Mathis and M. Allman, “A Framework for Defining Empirical Bulk Transfer Capacity Metrics,” July 2001, IETF RFC 3148. 43
- [94] M. Allman, V.Paxson, and W.Stevens, “TCP Congestion Control,” Apr. 1999, IETF RFC 2581. 43
- [95] G. Jin and B. Tierney, “Netest: A Tool to Measure Maximum Burst Size, Available Bandwidth and Achievable Throughput,” in *Proceedings of the 2003 International Conference on Information Technology*, Newark, New Jersey, Aug. 2003. 43, 44
- [96] Steven M. Bellovin, “A Best-Case Network Performance Model,” Tech. Rep., ATT Research, Feb. 1992. 45
- [97] Van Jacobson, “pathchar: A Tool to Infer Characteristics of Internet Paths,” Apr. 1997, Online at: <ftp://ftp.ee.lbl.gov/pathchar/>. 45, 53, 55
- [98] Allen B. Downey, “Using Pathchar to Estimate Internet Link Characteristics,” in *SIGCOMM '99: Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Cambridge, MA, USA, Sept. 1999, pp. 222–223. 45, 53, 55
- [99] Bruce Mah, “pchar: a Tool for Measuring Internet Path Characteristics,” Feb. 1999, Online at: <http://www.kitchenlab.org/www/bmah/Software/pchar/>. 45, 53, 55
- [100] A. Pasztor and D. Veitch, “Active Probing using Packet Quartets,” in *IMW '02: Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, Nov. 2002. 45
- [101] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, “SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments,” Aug. 2001, Online at: <http://sprobe.cs.washington.edu>. 47, 53, 55

## BIBLIOGRAPHY

---

- [102] Ravi Prasad, Constantinos Dovrolis, and Bruce Mah, “The Effect of Layer-2 Store-and-Forward Devices on Per-Hop Capacity Estimation,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, Mar. 2003. 47
- [103] V. Jacobson, “Congestion Avoidance and Control,” in *SIGCOMM '88: Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Stanford, CA, USA, Aug. 1988. 47
- [104] Srinivasan Keshav, “A Control-Theoretic Approach to Flow Control,” in *SIGCOMM '91: Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Sept. 1991. 47
- [105] Jean-Chrysotome Bolot, “End-to-end Packet Delay and Loss Behavior in the Internet,” in *SIGCOMM '93: Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, San Francisco, CA, USA, Sept. 1993, pp. 289–298. 47
- [106] Robert L. Carter and Mark E. Crovella, “Measuring Bottleneck Link Speed in Packet-Switched Networks,” *Performance Evaluation*, vol. 27, no. 8, pp. 297–318, Oct. 1996. 47, 48, 53, 55, 70
- [107] Kevin Lai and Mary Baker, “Measuring Bandwidth,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, New York, NY, USA, Apr. 1999, pp. 235–245. 47
- [108] Kevin Lai and Mary Baker, “Measuring Link Bandwidths Using A Deterministic Model of Packet Delay,” in *SIGCOMM '00: Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Stockholm, Sweden, Aug. 2000, pp. 283–294. 47, 53, 55
- [109] Constantinos Dovrolis, Parmesh Ramanathan, and David Moore, “What do Packet Dispersion Techniques Measure?,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, Alaska, USA, Apr. 2001, pp. 905–914. 47, 48, 95
- [110] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore, “Packet-Dispersion Techniques and a Capacity-Estimation Methodology,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 963–977, Dec. 2004. 47, 53, 55, 70, 101, 108, 114
- [111] Vern. Paxson, “End-to-End Internet Packet Dynamics,” *IEEE/ACM Transaction on Networking*, vol. 7, no. 3, pp. 277–292, June 1999. 48
- [112] A. Pasztor and D. Veitch, “The Packet Size Dependence of Packet Pair Like Methods,” in *Proceedings of IEEE/IFIP International Workshop on Quality of Service (IWQoS) 2002*, Miami Beach, FL, USA, May 2002. 48

- 
- [113] Manish Jain and Constantinos Dovrolis, “End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput,” *IEEE/ACM Transactions in Networking*, , no. 295-308, Aug. 2003. 49, 50, 53, 55, 95, 96, 115
- [114] Bob Melander, Mats Bjorkman, and Per Gunningberg, “A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks,” in *IEEE GLOBECOM—Global Internet Symposium*, San Francisco, CA, USA, Nov. 2000, pp. 415–420. 49
- [115] Bob Melander, Mats Bjorkman, and Per Gunningberg, “Regression-Based Available Bandwidth Measurements,” in *International Symposium on Performance Evaluation of Computer and Telecommunications Systems, 2002*, July 2002. 49, 53, 55, 95
- [116] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, “pathChirp: Efficient Available Bandwidth Estimation for Network Paths,” in *Proceedings of Passive and Active Measurements (PAM) Workshop*, La Jolla, CA, USA, Apr. 2003. 49, 53, 55, 96, 115
- [117] Ningning Hu and Peter Steenkiste, “Evaluation and Characterization of Available Bandwidth Probing Techniques,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, Aug. 2003. 49, 50, 53, 54, 55, 96, 97, 115
- [118] Aditya Akella, Srinivasan Seshan, and Anees Shaikh, “An Empirical Evaluation of Wide-Area Internet Bottlenecks,” in *IMC '03: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, Miami Beach, FL, USA, 2003. 49, 53
- [119] K. Harfoush, A. Bestavros, and J. Byers, “Measuring Bottleneck Bandwidth of Targeted Path Segments,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, Apr. 2003. 49, 53
- [120] Ningning Hu, Erran L. Li, Zhuoqing Morley Mao, Peter Steenkiste, and Jia Wang, “Locating Internet Bottlenecks: Algorithms, Measurements, and Implications,” in *SIGCOMM '04: Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2004, pp. 41–54. 49, 53
- [121] Jacob Strauss, Dina Katabi, and Frans Kaashoek, “A Measurement Study of Available Bandwidth Estimation Tools,” in *IMC '03: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, Miami Beach, FL, USA, 2003, pp. 39–44. 49, 50, 53, 54, 55, 95
- [122] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, “Multifractal Cross-Traffic Estimation,” in *ITC Conference on IP Traffic, Modeling and Management*, Monterey, CA, September 2000. 50, 53, 55
- [123] Guanghui He and Jennifer C. Hou, “On Exploiting Long Range Dependence of Network Traffic in Measuring Cross Traffic on an End-to-end Basis,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, Mar. 2003. 50

- [124] M. Mathis and J. Mahdavi, “Diagnosing Internet Congestion with a Transport Layer Performance Tool,” in *Proceedings of INET '96*, Montreal, Canada, June 1996. [51](#), [53](#), [55](#)
- [125] Mark Allman, “Measuring End-to-end Bulk Transfer Capacity,” in *IMW '01: Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, USA, Nov. 2001, pp. 139–143. [51](#), [53](#), [55](#)
- [126] Wu Xiuchao and A. L. Ananda, “Link Characteristics Estimation For IEEE 802.11 DCF Based WLAN,” in *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 302–309. [51](#), [55](#)
- [127] Samarth Shah, “Available Bandwidth Estimation in IEEE 802.11-based Wireless Networks,” in *Finale Report of Bandwidth Estimation ISMA Workshop*, San Diego, CA, USA, Dec. 2003. [51](#), [55](#), [84](#), [94](#), [95](#), [114](#), [115](#)
- [128] Yegyalakshmi Easwaran and Miguel A. Labrador, “Evaluation and Application of Available Bandwidth Estimation Techniques to Improve TCP Performance,” in *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 268–275. [54](#)
- [129] Manish Jain and Constantinos Dovrolis, “Ten Fallacies and Pitfalls in End-to-End Available Bandwidth Estimation ,” in *IMC '04: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, Sicily Italy, Oct. 2004. [55](#)
- [130] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE Journal on Selected Areas in Communications, Wireless series*, vol. 18, no. 3, pp. 535–547, Mar. 2000. [56](#), [58](#), [60](#), [72](#), [74](#)
- [131] Haitao Wu, Yong Peng, Keping Long, Shiduan Cheng, and Jian Ma, “Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, New York, USA, June 2002, pp. 599–607. [58](#), [59](#)
- [132] P. Chatzimisios, V. Vitsas, and A. C. Boucouvalas, “Throughput and Delay Analysis of IEEE 802.11 Protocol,” in *Proceedings of IEEE International Workshop on Network Appliances, (IWNA) 2002*, Liverpool, UK, 2002, pp. 168–174. [58](#)
- [133] P. Chatzimisios, V. Vitsas, and A. C. Boucouvalas, “IEEE 802.11: Packet Delay - A Finite Retry Limit Analysis ,” in *Proceedings of IEEE Globecom 2003*, San Francisco, USA, 2003, pp. 950–954. [58](#), [59](#)
- [134] Periklis Chatzimisios, Anthony C. Boucouvalas, and Vasileios Vitsas, “Performance Analysis of IEEE 802.11 DCF in Presence of Transmission Errors,” in *Proceedings of IEEE International Conference on Communications (ICC)*, Paris, France, June 2004, vol. 27, pp. 3854–3858. [58](#), [59](#), [60](#), [73](#), [74](#)
- [135] Federico Cali, Marco Conti, and Enrico Gregori, “IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement..,” in *Proceedings of IEEE Conference on*

- 
- Computer Communications (INFOCOM)*, San Francisco, CA, USA, Mar. 1998, pp. 142–149. 59
- [136] Federico Cali, Marco Conti, and Enrico Gregori, “Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 785–799, 2000. 59
- [137] Federico Cali, Marco Conti, and Enrico Gregori, “IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1774–1786, Sept. 2000. 59
- [138] Y. Xiao and J. Rosdahl, “Throughput and Delay Limits of IEEE 802.11,” *IEEE Communications Letters*, vol. 6, no. 8, pp. 355–357, Aug. 2002. 60
- [139] Y. Xiao and J. Rosdahl, “Performance Analysis and Enhancement for the Current and Future IEEE 802.11 MAC Protocols,” *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, special issue on Wireless Home Networks, vol. 7, no. 2, pp. 6–19, Apr. 2003. 60
- [140] Jangeun Jun, Pushkin Peddabachagari, and Mihail Sichitiu, “Theoretical Maximum Throughput of IEEE 802.11 and its Applications,” in *Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications (NCA’03)*, Cambridge, MA, USA, Apr. 2003, pp. 249–256. 60
- [141] University of California Berkeley, “The Network Simulator - ns-2,” Online at: <http://www.isi.edu/nsnam/ns/>. 60
- [142] Javier del Prado and Sunghyun Choi, “Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement,” in *Proceedings of IEEE International Conference on Communications (ICC)*, Anchorage, Alaska, May 2003. 61
- [143] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, “Opportunistic Media Access for Multirate Ad Hoc Networks,” in *MobiCom ’02: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, Atlanta, Georgia, Sept. 2002. 61, 67
- [144] Mingzhe Li, “An Extension of Rate-Adaptive MAC Protocol for NS2 Simulator,” 2004, Online at: [http://www.cs.wpi.edu/lmz/web/docs/cs525m\\_report.pdf](http://www.cs.wpi.edu/lmz/web/docs/cs525m_report.pdf). 61
- [145] Anand Balachandran, Geoffrey M. Voelker, Paramvir Bahl, and P. Venkat Rangan, “Characterizing User Behavior and Network Performance in a Public Wireless LAN,” in *SIGMETRICS ’02: Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Marina Del Rey, California, 2002, pp. 195–205. 62
- [146] Diane Tang and Mary Baker, “Analysis of a Metropolitan-area Wireless Network,” in *MobiCom ’99: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, Washington, USA, 1999, pp. 13–23. 62

## BIBLIOGRAPHY

---

- [147] Diane Tang and Mary Baker, “Analysis of a Local-area Wireless Network,” in *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, Massachusetts, USA, 2000, pp. 1–10. [62](#)
- [148] David Kotz and Kobby Essien, “Analysis of a Campus-wide Wireless Network,” in *MobiCom '02: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, Atlanta, Georgia, USA, 2002, pp. 107–118. [62](#)
- [149] Camden C. Ho, Krishna N. Ramachandran, Kevin C. Almeroth, and Elizabeth M. Belding-Royer, “A Scalable Framework for Wireless Network Monitoring,” in *WMASH '04: Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, Philadelphia, PA, USA, 2004, pp. 93–101. [62](#)
- [150] A. Balachandran and G. Voelker, “WRAPI – Real-time Monitoring and Control of an 802.11 Wireless LAN,” Tech. Rep., CS at UCSD, 2004. [62](#)
- [151] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris, “Link-level Measurements from An 802.11b Mesh Network,” in *SIGCOMM '04: Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Portland, Oregon, USA, 2004, pp. 121–132. [63](#)
- [152] Jihwang Yeo, Moustafa Youssef, and Ashok Agrawala, “A Framework for Wireless LAN Monitoring and Its Applications,” in *ACM Workshop on Wireless Security (WiSe 2004) in conjunction with ACM MobiCom 2004*, Philadelphia, PA, USA, Oct. 2004. [63](#)
- [153] G. Judd and P. Steenkiste, “Repeatable and Realistic Wireless Experimentation through Physical Emulation,” in *Proceedings of HotNets-II*, Nov. 2003. [63](#)
- [154] Tony Sun, Guang Yang, Ling-Jyh Chen, M. Y. Sanadidi, and Mario Gerla, “A Measurement Study of Path Capacity in 802.11b based Wireless Networks,” in *The Wireless Traffic Measurements and Modeling Workshop (WiTMeMo'05, in conjunction with MobiSys'05)*, Seattle, USA, 2005. [66](#), [94](#)
- [155] Sourav Pal, Sumantra Kundu, Kalyan Basu, and Sajal Das, “Performance Evaluation of IEEE 802.11 Multi-Rate Control Algorithms using Heterogenous Traffic and Real Hardware,” in *Proceedings of Passive and Active Measurements (PAM) Workshop*, Adelaide, Australia, Mar. 2006. [67](#)
- [156] Amit P. Jardosh, Krishna N. Ramachandran, Kevin C. Almeroth, and Elizabeth M. Belding-Royer, “Understanding Congestion in IEEE 802.11b Wireless Networks,” in *IMC '05: Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, Berkeley, CA, 2005. [67](#), [122](#)
- [157] Ratish J. Punnoose, Pavel V. Nikitin, and Daniel D. Stancil., “Efficient Simulation of Ricean Fading within a Packet Simulator,” in *Proceedings Vehicular Technology Conference*, 2000. [69](#)

- 
- [158] Mingzhe Li, Mark Claypool, and Robert Kinicki, “Modeling and simulating packet dispersion in wireless 802.11 networks,” Tech. Rep. WPI-CS-TR-06-03, Computer Science Department at Worcester Polytechnic Institute, Mar. 2006. [81](#), [103](#), [106](#), [107](#), [108](#), [110](#), [131](#)
- [159] Tony Sun, Ling-Jyh Chen, Guang Yang, M. Y. Sanadidi, and Mario Gerla, “Sen-Probe: Path Capacity Estimation in Wireless Sensor Networks,” in *The Third International Workshop on Measurement, Modelling, and Performance Analysis of reless Sensor Networks(SenMetrics’05, in conjunction with MobiQuitous’05)*, San Diego, USA, 2005. [94](#), [114](#), [115](#)
- [160] Ling-Jyh Chen, Tony Sun, Guang Yang, M. Y. Sanadidi, and Mario Gerla, “Ad Hoc Probe: Path Capacity Probing in Wireless Ad Hoc Networks,” in *WICON ’05: Proceedings of the First International Conference on Wireless Internet*, Washington, DC, USA, 2005, pp. 156–163. [94](#), [114](#), [115](#)
- [161] Andreas Johnsson, Bob Melander, and Mats Björkman, “Bandwidth Measurement in Wireless Networks,” in *Mediterranean Ad Hoc Networking Workshop*, Porquerolles, France, June 2005. [95](#), [115](#), [131](#)
- [162] H. K. Lee, V. Hall, K. H. Yum, K. I. Kim, and E. J. Kim, “Bandwidth Estimation in Wireless LANs for Multimedia Streaming Services,” in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, Canada, July 2006. [95](#), [114](#), [115](#)
- [163] Mingzhe Li, Mark Claypool, and Robert Kinicki, “Packet dispersion in IEEE 802.11 wireless networks,” in *Proceedings Performance and Management of Wireless and Mobile Networks (P2MNet)*, Tampa, FL, USA, Nov. 2006. [96](#)
- [164] M. Zink, O. Kunzel, J. Schmitt, and R. Steinmetz, “Subjective Impression of Variations in Layer Encoded Videos,” in *Proceedings of the 11th International Workshop on Quality of Service (IWQoS)*, Monterey, CA, USA, June 2003. [97](#)
- [165] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP Throughput: A Simple Model and Its Empirical Validation,” in *Proceedings of ACM SIGCOMM*, Vancouver, British Columbia, Canada, 1998. [98](#)
- [166] Li Lao, Constantine Dovrolis, and M. Y. Sanadidi, “The Probe Gap Model Can Underestimate the Available Bandwidth of Multihop Paths,” *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 29–34, 2006. [101](#)
- [167] Mark Claypool, Robert Kinicki, Feng Li, Mingzhe Li, Rui Lu, and Huahui Wu, “Measuring the Queue Sizes of IEEE 802.11 Wireless Access Points,” Tech. Rep., CS Department at Worcester Polytechnic Institute, May 2006. [103](#), [109](#)
- [168] Alessio Botta, Salvatore D’Antonio, Antonio Pescap&#233;, and Giorgio Ventre Ven-  
tre, “BET: A Hybrid Bandwidth Estimation Tool,” in *ICPADS ’05: Proceedings of the 11th International Conference on Parallel and Distributed Systems - Workshops*, Washington, DC, USA, 2005, pp. 520–524. [103](#)

## BIBLIOGRAPHY

---

- [169] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda, “Achieving Moderate Fairness for UDP Flows by Path-status Classification,” in *LCN '00: Proceedings of the 25th Annual IEEE International Conference on Local Computer Networks*, Washington, DC, USA, 2000, p. 252, IEEE Computer Society. 113
- [170] Saad Biaz and Nitin H. Vaidya, “Distinguishing Congestion Losses from Wireless Transmission Losses: A Negative Result,” *iccn*, vol. 00, pp. 722, 1998. 113
- [171] S. Lin and D. J. Costello Jr., “Automatic-repeat-request Error-control Schemes,” *IEEE Commun. Mag.*, vol. 22, pp. 5–17, Dec. 1984. 151
- [172] G. J. Conklin, G. S. Greenbaum, K. O. Lillevold, A. F. Lippman, and Y. A. Reznik, “Video Coding for Streaming Media Delivery on the Internet,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 269–281, Mar. 2001. 151
- [173] Mingzhe Li, Mark Claypool, Robert Kinicki, and James Nichols, “Characteristics of Streaming Media Stored on the Web,” *ACM Transactions on Internet Technology (TOIT)*, vol. 5, no. 4, Nov. 2005. 168
- [174] Z. Wang, S. Banerjee, and S. Jamin, “Studying Streaming Video Quality: From an Application Point of View,” in *Proceedings of ACM Multimedia*, Berkeley, California, USA, Nov. 2003, pp. 327–330. 170