ADAPTIVE WIRELESS NETWORKING FOR VIDEO STREAMING

WANG HUI

(B.Eng., HIT, China)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2015

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

WANG HUI 6 August 2015

ACKNOWLEDGMENTS

This thesis would not have been possible without the guidance and support of many people during my PhD study. It is now my great pleasure to take this opportunity to thank them.

First of all, I would like to express my sincere gratitude to my supervisors Prof. Wei Tsang Ooi and Prof. Mun Choon Chan. Prof.Ooi introduced me to the research field of streaming. He taught me to think thoroughly, communicate logically, and write clearly. I have greatly benefited from these invaluable skills. I also thank him for patiently reading through my papers and thesis, and for giving me insightful comments and suggestions. Prof.Chan has been greatly supportive in solving the research problems. He taught me to be innovative. I learned a lot from him when we are discussing ideas. He also encouraged me to push the boundaries and provided invaluable support. I am particularly grateful for the insightful advices and comments. It has been my great honor to be their student.

I would like to thank my thesis advisory committee Prof. Yong Chiang Tay, Prof. Richard T. B. Ma, and Prof. Ben Leong for their invaluable assistance, feedback, and patience at all stages of this thesis.

I want to express my sincere appreciation to my dear colleagues during my PhD

study, especially Wei Wang, Vu-Thanh Nguyen, Girisha Durrel De Silva, Padmanabha Venkatagiri Seshadri, Kartik Sankaran, Jingli Cai, Xiangfa Guo for their support and constructive comments to my research work. I am particularly thankful to my nine-year schoolmate and my best friend Nuo Xu for helping me through the hard times.

Last, but not least, I would like to thank my dear wife Mengmeng Yang for her understanding and love during the past eight years. She was always there cheering me up and stood by me through the good times and bad. Her support and encouragement was in the end what made this dissertation possible. Also, I am deeply and forever indebted to my parents for their love and support. None of my achievements would be possible without their love and encouragement.

CONTENTS

ABSTRACT

Although wireless technologies have evolved significantly over the past decades, the wireless bandwidth, however, is still insufficient to support the fast-growing mobile traffic, especially due to the increasing popularity of mobile video streaming applications. Indeed, streaming high-definition videos over wireless networks to a large number of heterogeneous users remains a challenging problem. In this thesis, we work towards designing effective solutions in improving the streaming quality and the network scalability by exploiting three essential adaptive techniques: *adaptive multicast rate*, *adaptive association control*, and *adaptive contention windows*.

First, we investigate *mixed resolutions tiling video* where tiles in a video frame can come from different resolution streams. This approach is flexible and effective in wireless multicast video streaming. Applying this scheme to adaptive wireless multicast, we have the following optimal adaptive multicast allocation problem: given the subset of tiles that each user requested, the link rate of each user, and the available time slots, at which resolution and at which link rate each tile should be sent, to maximize the overall video quality by all users. By applying dynamic programming, we design an efficient algorithm to optimally solve this problem.

Second, we explore the general multi-sessions multicast allocation problem by con-

sidering each tile as an individual video session, and each user can subscribe multiple sessions. Moreover, multiple access points are deployed to improve the network capacity. We present a joint user and rate allocation scheme for video multicast over multiple access points. This scheme intelligently determines user to access point association, the video resolution version (quality) to be delivered for each session, and the transmission link rate for each video version.

Last, we study the heterogeneous (or asymmetric) topologies that are present when neighboring access points are operating on the same channel. The saturated traffic introduced by video streaming could lead to severe unfairness in presence of these topologies. We build an analytical model to characterize the network performance. Based on the model, we suggest an adaptive contention window tuning mechanism which is able to effectively remedy the unfair issues.

Evaluation of the designed approaches, including both testbed implementation and large scale simulation, shows that our framework significantly improves the system performance in quality of service (streaming quality), goodput, and fairness.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Table 1: System Parameters for 802.11a MAC

Notation	Definition	Time	Slots[a]
σ	A time slot	$9\mu s$	1σ
T_{SIFS}	Time duration of SIFS	$16\mu s$	2σ
T_{DIFS}	Time duration of DIFS	$34\mu s$	4σ
T_{RTS}	Time to transmit an RTS frame plus a SIFS period	$52\mu s + 16\mu s$	8σ
T_{CTS}	Time to transmit a CTS frame	$44\mu s$	5σ
T_{ACK}	Time to transmit an ACK frame	$44\mu s$	5σ

[a] The number of slots is rounded to the nearest integer value.

Table 2: Key Notations

Notation	Definition
T	AP capacity in terms of time slots (802.11 slots)
n	Number of users (or clients)
N_{AP}	Number of APs in the network
N_r	Number of distinct link rate levels
N_v	Total number of tiles (or video sessions)[b]
M	The number of available resolution levels
$\mathbf{V}(i)$	The set of tiles interested by user i
r_{ij}	The estimated link rate between user i and AP j
r_i	The estimated link rate between user i and the associated AP
R_{iv}	Resolution level of tile v requested by user i
L_{iv}	Lowest resolution level of tile v guaranteed to user i
v^m	Tile v at resolution level m
s_v^m	The size (in bytes) of tile v at resolution level m
u_{iv}^m	Utility of tile v at resolution level m assigned to user i
F_i	Time to transmit data frame (including overhead) of node i
l_i	Average length of data frame from node i
W_i	Contention window size of node i
$\mathcal{L}_A$, $\bar{\mathcal{L}}_A$	The length, average length of a round of A (in time slots)
$\mathcal{L}_C$, $\bar{\mathcal{L}}_C$	The length, average length of a gap of C (in time slots)
E_{RTS}	Expected number of RTS requests from A in a cycle
S_i	Throughput of node i
$\Re_{ij}$, $\hat{\Re}_{ij}$	Expected data transmission ratio of node i to j

[b] A tile refers to a small view region of an entire video frame in zoomable video (Chapter 3); it refers to a single video session in multi-sessions multicast (Chapter 4).

CHAPTER 1

INTRODUCTION

Mobile data traffic has exploded over the past decade. The Cisco Visual Index report in 2015 [37] shows that the amount of mobile data in 2014 was nearly 30 times the amount of the entire Internet data in 2000. The growth of mobile traffic is mainly driven by user demands on mobile video streaming, which becomes one of the most popular applications in recent years. According to the forecast from this report, mobile video traffic will increase 13 times from 2014 to 2019 and will account for 72% of the total mobile data traffic by the end of 2019, up from 55% in 2014. The statistics indicates that the amount of mobile video traffic is increasing rapidly as the number of hand-held devices (e.g., smartphones, tablets) grows. Moreover, as technology evolves, high definition (HD) videos have become universally available and will be the dominant form of video content, contributing to the increasing amount of video traffic.

At the same time, wireless LANs based on 802.11 have been widely deployed, and its adoption is still growing. The physical layer link rates have increased from 1Mbps in the original 802.11b to 1Gbps in the 802.11ac standard, but the user level throughput has not seen a commensurate increase. The current wireless networks, however, are

severely inadequate in providing satisfactory quality when streaming videos to a large number of users, especially for HD videos. To bridge the gap between the rapidly increased video traffic demand and the limited wireless bandwidth provided by 802.11 networks, a considerable research effort has been devoted to improving the performance of wireless streaming systems.

When streaming a video to multiple clients, wireless multicast can be exploited to improve the system utilization while minimizing the wireless resource usage. Wireless multicast is a natural operation for delivering traffic to multiple clients simultaneously, which can arise in many scenarios, such as broadcasting live sports (e.g., ESPN, Sky-Sports), TV programs (e.g., Netflix, HBO), and traditional broadcast channels (e.g., BBC, NBC). Apart from leveraging wireless multicast, deploying multiple access points (APs) is another effective way to increase the network performance. To effectively utilize the deployed multiple APs, the mobile clients have to make intelligent decision about which AP to associate with. The network and host heterogeneities, however, could impair the effectiveness of these approaches, which will be discussed in the following section.

1.1 Network and Host Heterogeneity

The wireless networks are inherently heterogeneous, with different wireless channel qualities and mobile devices of different specifications. In this dissertation, we focus on the wireless network heterogeneities in link quality, streaming video quality, and network topology, which could lead to severe performance issues.

Heterogeneous Links

Due to the differences in distances, mobility, and locations, the mobile clients are typically experiencing different wireless channel qualities. The achievable wireless network speed can vary by multiple orders of magnitude.

To transmit more efficiently over the unstable wireless channels, various auto link rate adaptation mechanisms have been proposed [36, 17, 79, 82, 72, 20]. The main idea

of these approaches is to dynamically and accurately adapt the transmission bit-rate (link rate) based on the estimated link quality. Apart from the unstable link qualities, the diverse link rates used by multiple clients associated with a single AP could result in a severe unfairness problem known as *rate anomaly* [34]. The reason for this anomaly is the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol provides equal access probability regardless of the underlying link rate. As a result, the overall system throughput is significantly hampered by the lower link rate.

Heterogeneous Streaming Qualities

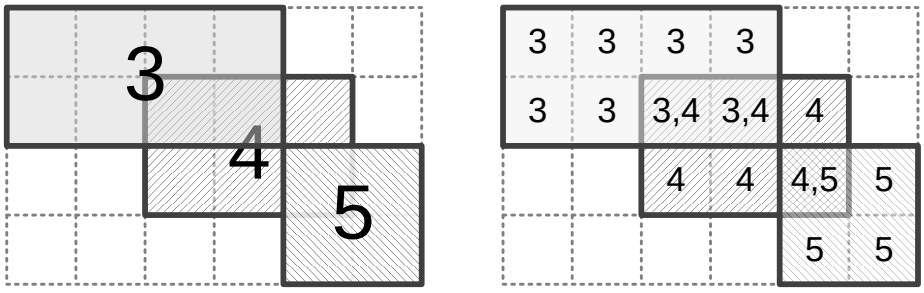
Recently, two emerging streaming approaches, *dynamic adaptive streaming* (e.g., DASH) [65, 10, 52] and *interactive streaming* (e.g., zoomable video) [49, 31, 62], are suggested to efficiently stream HD videos. Applying wireless multicast to these adaptive streaming approaches, however, is ineffective due to the disparities in the streaming video qualities. In particular, the streaming quality primarily depends on three factors: wireless link quality, device screen resolution, and region/video of interest.

First, as discussed in the previous subsection, the heterogeneous link qualities are prevalent in wireless networks. The key idea of adaptive streaming is to dynamically adapt video resolution according to the estimated link conditions. Specifically, a higher resolution level is delivered to the client with good link quality. Since the mobile clients may experience different link qualities, the resolution levels assigned to them are different.

Next, various types of mobile devices are connected to the wireless networks, which include laptops, smartphones, and tablets. The screen resolutions of these devices can be very different. To improve the quality of experience of HD video streaming systems for heterogeneous devices, the dynamic adaptive streaming approaches also take screen resolution into consideration.

Finally, to address the mismatch of video resolutions between the capture device and playback, zoomable video streaming has recently been proposed [49, 31]. In zoomable video, a user is able to zoom into a selected region of interest (RoI) in the

video, to view the RoI with higher resolution. The request content from users can be different in selected regions and the quality level that depends on the region size and screen resolution.



(a) The RoI regions with request resolution levels. The number represents the resolution level required by the user.

(b) The required resolution levels for each grid. The number(s) in a grid indicates the streamed resolution levels.

Figure 1.1: Three users request different RoI regions.

Figure 1.1 shows an example of heterogeneous video requests, where three users are interested in three different regions (or three sets of video sessions). As can be seen, three overlapping grids are requested by two users. Since different users are requesting different resolution levels, two multicast transmissions are required for each overlapped grid to meet all users requests (Figure 1.1(b)).

Heterogeneous Topologies

In the scenarios where neighboring access points (APs) operate on the same channel, the fully-connected network (single AP) will be divided into multiple collision domains, where the hidden terminals are present. The prior studies [14, 28, 76] demonstrate that hidden terminals may result in substantial performance degradation, especially under heterogeneous (or asymmetric) network conditions [28, 75, 76]. In this dissertation, two terms, *heterogeneous topology* and *asymmetric topology*, are used interchangeably.

An instance of well-known asymmetric topology with two APs is shown in Figure 1.2. The figure consists of two competing flows: a dominant flow (from AP_2 to C_2) and a weak flow (from AP_1 to C_1). This topology indicates that the receiver of

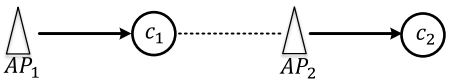


Figure 1.2: An asymmetric topology with two competing flows. There are two senders AP_1 and AP_2, and two clients C_1 and C_2. The arrows represent the directions of the data flows, and the dotted lines indicate overhearing links.

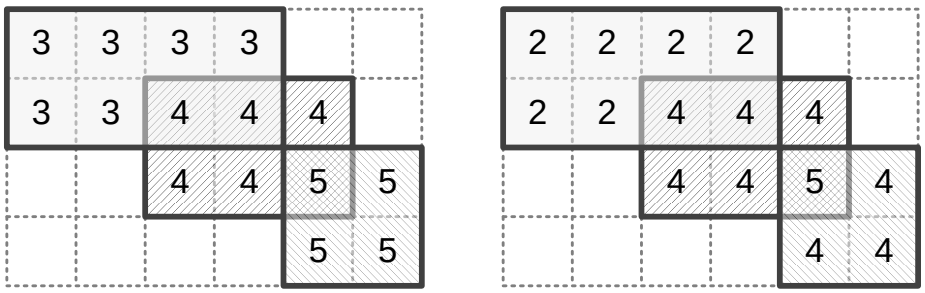
the weak flow (C_1) can hear the sender of the dominant flow (AP_2), while the receiver of dominant flow (C_2) cannot hear the sender of weak flow (AP_1). As a result, under high traffic load (video streaming) the sender AP_2 can always successfully compete for channel access and then receiver C_1 is suffering from collisions. Therefore, we will obtain an extremely low throughput ratio of C_1 to C_2.

1.2 Adaptive Schemes

This thesis employs the following adaptive techniques to overcome the inefficiencies due to the above discussed heterogeneities.

1.2.1 Adaptive Quality Allocation

Figure 1.1 illustrates the inefficiency in conventional adaptive video multicast streaming. We now discuss how to utilize the adaptive quality allocation scheme to alleviate the limitations of conventional multicast.



(a) The highest requested quality level from all the users

(b) A possible instance of allocating low resolution to unpopular tiles

Figure 1.3: Grids in a single RoI region may come from different resolution streams.

Mixed resolutions tiling scheme. The mismatch between increasingly large video

resolution and constrained screen size of mobile devices has led to the proposal of zoomable video systems based on tiled video. In the existing architecture, a tiled video frame is constructed from multiple tiles in a single resolution stream. In this work, we explore *mixed resolutions tiling scheme*, where tiles within a video frame could come from streams with different resolutions. Before playback, the tiles with higher/lower resolution levels will be scaled down/up to the desired resolution level at the client side.

By allowing resolutions to be mixed, we only need a single multicast transmission for each tile at the highest requested resolution level (Figure 1.3(a)) rather than two transmissions for each overlapped tile (Figure 1.1(b)). By doing so, we can save the bandwidth consumption by reducing the number of multicast transmissions. In scenarios with restricted time constraints, we can further reduce the bandwidth consumption by allocating less popular tiles with lower resolution levels (Figure 1.3(b)).

Adaptive multi-sessions multicast. Figure 1.3 has shown how to exploit wireless multicast for mixed resolutions tiling scheme in zoomable video. We also observe that if each tile is considered as an individual video session, the above resolution allocation problem is equivalent to the general adaptive multi-sessions multicast problem, where each user can subscribe to one or multiple video sessions.

1.2.2 Adaptive Wireless Multicast Rate

The previous section demonstrates how to save bandwidth consumption for adaptive video streaming by leveraging wireless multicast. The conventional wireless multicast transmission, however, has two critical issues: *low multicast rate* and *rate anomaly*. To address these deficiencies, our work dynamically adapts wireless multicast transmission rates.

The wireless multicast is typically transmitted at the basic link rate (1Mbps or 2Mbps) to ensure that all users can successfully receive the frames. While in most cases, only a subset of users is interested in a particular transmission. Therefore, instead of transmitting at the basic rate, we can multicast at the lowest unicast rate among all destined users [21].

Additionally, due to the link heterogeneity, our wireless network may suffer from the rate anomaly problem. Such rate anomaly problem could occur in two situations: (i) A user with considerably low link quality is requesting the same content (tile/session) as other users; (ii) A multicast group has a considerably low transmission link rate compared with other multicast groups. In such scenarios, we could intelligently allocate this user or multicast group with lower video qualities.

1.2.3 Adaptive Multicast Association Control

To further increase the wireless network capacity, multiple access points are typically deployed. To utilize the deployed multiple APs more effectively, the mobile clients have to make intelligent decision about which AP to associate with. The multicast association control for adaptive video streaming, however, is challenging because of the conflicts between high transmission rate, load balancing, and exploiting multicast opportunities. More specifically, if a client simply chooses the AP with the highest Received Signal Strength Indicator (RSSI) value (attains high transmission rate), this association mechanism could result in severely unbalanced workload between APs and reduce the multicast opportunities.

This thesis presents an association scheme that balances the trade-off between these factors by determining user to access points association, the video resolution version to be delivered for each session, and the transmission link rate for each video version.

1.2.4 Adaptive Contention Windows

In the large-scale wireless network with multiple APs, the network performance is mainly determined by CSMA/CA, which is a medium access control (MAC) protocol used in the popular IEEE 802.11 wireless networking standard.

In particular, the contention window (CW) size plays an important role in the widely used CSMA/CA protocol. By default, the same configuration of CW size is used by all senders (APs) to ensure fairness. Such uniform configurations, however, could lead to severe unfairness in the presence of asymmetric topologies (Figure 1.2).

To address the ineffectiveness of medium access control protocol in CSMA/CA networks, various solutions that dynamically tune the CW size for each AP have been suggested [35, 28, 70, 19, 83, 54, 76]. These prior works, however, are either limited to a fully connected network [35] or fail to provide closed-form expressions to quantify the adaptive windows [28, 70, 19, 83, 54, 76].

In contrast, our work provides closed-form expressions that are simple yet accurate, yielding a contention window tuning mechanism that ensures fairness under various asymmetric conditions and topologies.

1.3 Contributions

In this thesis, we work towards designing a scalable adaptive wireless streaming framework which is able to provide high quality of service for delivering HD videos to a large number of users. This section presents a brief overview of our contributions in this thesis.

1.3.1 Optimal Multicast Allocation for Adaptive Streaming

We first conduct a psychophysical assessment to explore the perceptual effect of mixing resolutions in tiled video (zoomable video), the results demonstrate that in most cases, the perceptual quality of mixing resolutions in tiled video is insignificant, as long as the variance of mixed resolution levels is low. Applying the adaptive multicast transmissions to mixed-resolutions tiled video, we obtain the following allocation problem: given the set of tiles that each user requested, the link rate of each user, and the available time slots, at which resolution should each tile be sent, to maximize the overall video quality received by all users. We designed an efficient algorithm to solve the problem above. This contribution is detailed in Chapter 3.

Our proposed optimal multicast significantly improves the average video quality by up to 12dB in terms of PSNR (Peak Signal-to-Noise Ratio) compared with the baseline schemes. Considering each tile as an individual video session, our algorithm can be applied to the optimal allocation of multi-sessions adaptive video streaming as well,

and has a lower, more practical, running time (grows linearly with the number of time slots) than the existing optimal allocation algorithms.

1.3.2 Multicast Allocation over Multiple APs

We generalized tiled/zoomable video to multi-sessions adaptive video streaming, where each tile is considered as an individual video session and each user can subscribe multiple video sessions. Moreover, multiple APs are deployed to improve the multicast system scalability, where neighboring APs operate on non-overlapping channels. We present JurCast, a joint user and rate allocation scheme to solve the maximization problem by determining user-to-access point association, the video resolution version (quality) to be delivered for each session, and the transmission link rate for each video session. The design of JurCast is presented in Chapter 4.

The evaluation of JurCast, including both system implementation and large scale simulation, shows that compared to the baseline schemes, our approach significantly improves the video quality (PSNR) and goodput by up to 3dB and 55%, respectively.

1.3.3 Mitigating Unfairness under Heterogeneous Topologies

Since the number of non-overlapping channels is insufficient in densely deployed AP environment, heterogeneous topologies may present when neighboring APs are operating on the same channel, as shown in Figure 1.2. In the scenarios with saturated traffic, such as, video streaming applications, these heterogeneous topologies could result in severe unfairness or even starvation. In this thesis, we develop a simple yet accurate analytical model to characterize the performance of such heterogeneous topologies. Our model allows us to estimate the throughput of two contending flows, given the contention window values. Moreover, the model can be applied to remedy unfairness issues by optimally adapting the contention window sizes. This contribution is described in Chapter 5.

In particular, we present simple closed-form equations that can be used to compute the appropriate contention windows that meet a given performance objective, such as fairness and throughput for heterogeneous topologies. Our model is flexible and can

support different notions of fairness, including max-min fairness, time fairness, and proportional fairness. The measurements in both testbed and simulation indicate that our adaptive CW tuning mechanism is effective in terms of both fairness and throughput.

1.4 Organization

The rest of this thesis is organized as follows: Chapter 2 presents a brief background of these adaptive techniques and discusses the related work. We present the problem formulation of allocating tile resolutions and design an optimal algorithm to solve the maximization problem in Chapter 3. Chapter 4 extends the network to multiple APs and takes AP association control into consideration. In Chapter 5, we analytically model the unfair topologies with multiple APs given contention window sizes. The conclusion and future work discussions are made in Chapter 6.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

In this chapter, we present the background and overview the existing research work that is closely relevant to this thesis. In particular, the first section reviews two adaptive streaming techniques: the adaptive video quality streaming and the tiled/zoomable video; the second section discusses the prior studies in adaptive wireless transmissions.

2.1 Background of Adaptive Streaming

In this section, we review two emerging adaptive streaming techniques: *adaptive video quality* and *tile-based zoomable video*.

2.1.1 Adaptive Video Quality

As the growth of mobile video traffic outpaces that of wireless network speed, adaptive video streaming technology that enables dynamic adaptation of video bit-rates is leveraged to adapt to the changing wireless network conditions. In the past, most video streaming technologies utilize the UDP-based protocols, such as RTP and RTSP. Re-

cently, the adaptive streaming technologies are almost exclusively based on HTTP, such as *dynamic adaptive streaming over HTTP* (DASH) [65, 10, 52]. This section briefly describes the background of these adaptive techniques.

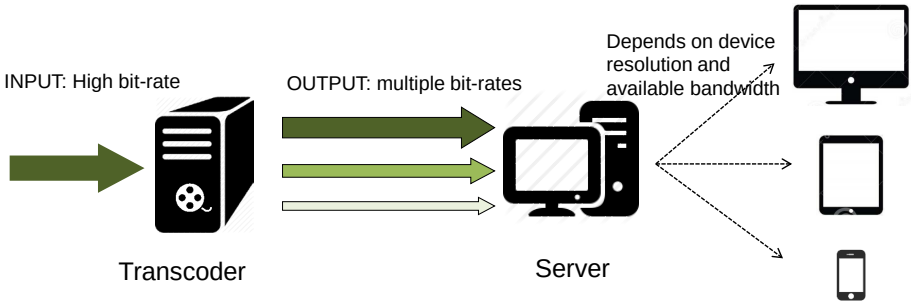


Figure 2.1: Overview of adaptive streaming architecture

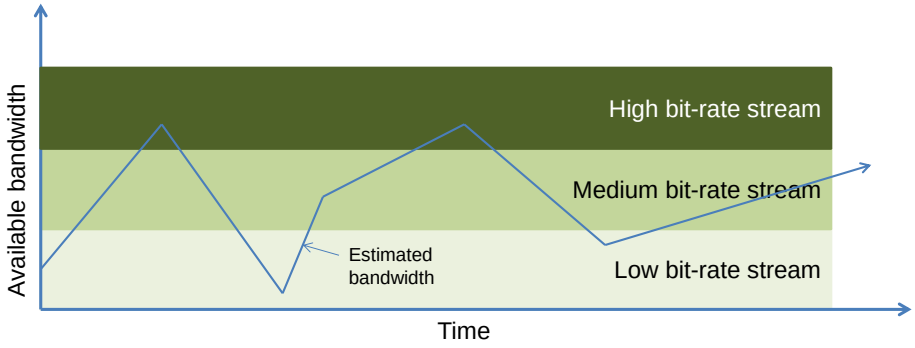


Figure 2.2: Stream bit rate as a function of network condition

Figure 2.1 presents an architecture overview of adaptive video streaming system. As can be seen from the figure, each video is encoded into multiple different resolution versions (bit-rates), which are hosted by the web server. The resolution version of the video that is transmitted to a client primarily depends on two factors: the screen resolution of the mobile device, which determines the request bit-rate version; and the available bandwidth between the client and the server. Generally, the video with higher bit-rate is streamed to the client with higher display resolution under good wireless link conditions. When the measured bandwidth cannot afford the relatively high requested bit-rate, the server will downgrade the bit-rate of the video to match the available bandwidth (shown in Figure 2.2).

The foremost benefit of using adaptive bit-rate streaming is the enhancement of perceived quality for the end user. Although quality of the video may vary according to the wireless channel condition, applying the adaptive mechanism, the delay due to channel quality variations is nearly unnoticeable. Additionally, this adaptive video streaming scheme can be easily deployed on a video server: despite the extra encoding equipment that is required initially to create the necessary streams in several resolution versions, hosting of dynamic adaptive streaming does not require any additional specialized hardware.

Due to previous discussed advantages, various commercial examples of dynamic adaptive streaming systems are implemented, including Apple's QuickTime Streaming Server [6], Move Networks [8], and Microsoft's SmoothStreaming [9]. In this thesis, we also employ the technique of dynamically adapting video quality (resolution) to exploit the available wireless bandwidth and improve the HD video streaming performance.

2.1.2 Tile-based Zoomable Video

While consumer video resolution has increased from HD to 8K resolution (has been supported on YouTube since June, 2015[1]), the physical screen size of mobile devices is normally constrained to ensure portability and ease of use. Due to screen size constraints, especially on mobile devices, and bandwidth constraints, however, video streaming playback is still limited in resolution. As a result, high resolution videos are typically scaled down before transmission (e.g., in dynamic adaptive streaming), leading to a loss in information.

To address the mismatch of video resolution between the capture device and playback, zoomable video streaming has recently been proposed [59, 50, 51, 71]. A zoomable video supports zoom and pan as two new operations for a user to interact with the video. In particular, a user is able to zoom into a selected region of interest (RoI) in the video, to view the RoI with higher resolution. The user essentially views the video through a viewport that defines a rectangular region in the high resolution video, from which the

[1] A Video called *Ghost Towns*

displayed video is cropped. While zooming in, users can pan around by moving the viewport to view different regions in the video.

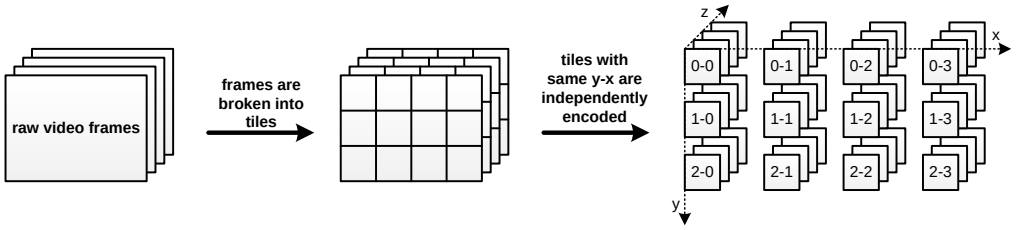


Figure 2.3: Tiled video

Zoomable video streaming is typically achieved using a technique called *tiled streaming*, where video frames are broken into a grid of tiles (Figure 2.3). We can view the video as a three dimensional matrix of tiles. Tiles at the same y-x position in the matrix are temporally grouped and coded along z axis. The video is encoded into different resolutions to support zooming. The zoom-out view corresponds to the lowest resolution. As the user zooms in, a minimum set of tiles from the higher resolution video covering the RoI region is streamed. The location of RoI can be changed by panning, while the resolution can be changed by zooming.

The tiles in the same y-x are decoded together by the zoomable player at the client side. The tile groups with different y-x positions can be decoded in parallel, each frame is formed by the uncompressed tiles with same z position. The frame will be displayed in the original order by the zoomable player when all the corresponding tiles are uncompressed. In the following part of this section, we will discuss two specific implementations of such tiling scheme: *fixed tile size* and *fixed number of tiles*.

Fixed Tile Size. One conventional implementation of tiled video approach is fixing tile size in the pixel domain [26, 59, 60]. For convenience, we assume that tiles are aligned with macroblock boundary. In other words, each tile consists of a set of macroblocks ($n \times n$ macroblocks). In existing works [59, 60, 27], at the server side, an original video is normally encoded into different versions (streams): frames of a low-resolution stream are constructed from a smaller number of tiles; and frames of higher-resolution streams are constructed from a larger number of tiles. At the client

side, the number of tiles required to cover the physical screen resolution is fixed, therefore, the bandwidth consumption for each user will be mostly constant. Initially, a low resolution version of the video will be sent to users. When a user zooms into a RoI within the video, the server will first determine a suitable high-resolution stream based on the requested RoI size (zoom level). It then selects tiles covering the requested RoI from this stream. This mechanism allows users to see their regions of interest in detail without consuming more bandwidth.

This fixed tile size RoI cropping technique performs well in small scale networks by unicasting video stream. In one of the use cases we consider, the video stream is consumed by a large number of users at one location (e.g., in a concert hall or stadium). To overcome the scalability issues with such a large number of users and RoI requests, wireless multicast scheme is employed. When the RoI regions from multiple users partially overlap, tiles from the overlapped regions could be potentially multicasted to all interested users to save bandwidth consumption. In zoomable video, different users, however, may have different zoom levels (i.e., different RoI sizes) and will need tiles from different versions encoded at different resolutions, which prevents the potential benefits of wireless multicast.

Fixed Number of Tiles. Instead of fixing tile size, using a fixed number of tiles to encode and decode videos could be more effective [74]. At the server side, an original video will be encoded into different resolution versions, but all versions consist of the same number of tiles. The same amount of tiles is required at the client side to decode each video frame. Within a frame, however, different tiles could come from different resolution streams. If a tile comes from a stream with resolution lower/higher than requested level, it will be scaled up/down accordingly. In zoomable video, when a user zooms into a region of interest (RoI) within the video, the server will first determine the tiles covering this RoI, and then associate each tile with an appropriate stream version, depending on their popularity and the resource constraints.

The proposed *mixed resolutions tiling* scheme has the following two essential ad-

vantages in tiled video streaming. First, benefiting from the scaling up/down operations for each tile, the multicast transmissions are considerably reduced. Next, by intelligently allocating resolution version to each tile, the mixing resolutions approach may considerably reduce bandwidth consumption without impairing much perceived video quality. As can be seen from Figure 1.3, the popular regions/tiles requested by many users could come from high-resolution streams; while tiles requested by one or few users could come from a low-resolution stream under limited bandwidth condition.

2.2 Adaptive Wireless Transmissions

This section discusses three essential adaptive techniques in wireless networks: *multicast link rate adaptation*, *multicast association control*, and *contention window adaptation*.

2.2.1 Multicast Data Rate Adaptation

Unicast Link Rate Adaptation

To improve the wireless system throughput in time-varying channel conditions, various studies have extensively investigated the *link rate adaptation* protocols that dynamically adapt its modulation and coding bit rate of unicast scenarios [36, 17, 79, 82, 72, 20]. Prior work on link rate adaptation primarily uses one of two information signals: *frame reception rate* or *signal-to-noise ratio* (SNR). Frame-level rate adaptation [3, 17, 79] estimates frame loss rates over tens or hundreds of frames or more. As a consequence, frame-level schemes are not responsive to channel variations that occur on shorter timescales. On the other hand, SNR-based protocols [36, 82, 20] can operate on shorter timescales by estimating the SNR on each reception and mapping it to the expected *bit error rate* (BER) using known SNR-BER relationships. The BER at a given SNR, however, might vary by many orders of magnitude between environments, these protocols must be carefully trained for each operating environment. Additionally, SNR measurements also require hardware-specific calibration [82], which is not available on many mobile devices.

Multicast Link Rate Adaptation

Based on the existing unicast link rate adaptation mechanisms, many multicast link rate adaptation approaches are recently suggested to exploit the shared nature of wireless spectrum [78, 57, 21, 61]. Instead of using the basic multicast rate (1Mbps or 2Mbps), these approaches transmit at a relatively high broadcast rate that is adapted according to the station experiencing the worst channel condition. For instance, DirCast [21] multicasts packet at the link rate of the worst client for each access point (AP). In video coding, such as scalable video and MPEG coding, the frames are unequally important. Inspired by this observation, Medusa [61] prioritizes the frames according to their importance and transmits the less import frames at higher link rates. By utilizing this heuristic frame level rate assignment, Medusa achieves higher video quality with limited resource constraints.

As the frame level retransmissions are disabled in wireless multicast, a major challenge in broadcast rate adaptation is to handle high broadcast loss rate. To protect the data from packet loss, the error protection/recovery protocols are utilized [78, 21, 61]. Similar as DirCast [21], the rate adaptation framework by Wong et al. [78] includes both rate selection and forward error correction (FEC) mechanisms to enable high rate transmission while protecting data from high loss ratio. This work designs an efficient algorithm to maximize the throughput of the worst receiver by jointly controlling the link rate and error protection level. To protect frame loss, XOR-based method [61] that retransmits a simple XOR-based coding of packets is another widely used approach.

Multicast Link Rate Adaptation with Adaptive Streaming

To further improve video streaming performance, considerable research efforts have been advocated to jointly adapt video data rate and multicast link rate.

The adaptive multicast for real time video in WiMAX network is firstly studied by Deb et al. [25], where *layer encoded video* is disseminated. In particular, the following problem within any scheduling frame is studied: for any time slot within the scheduling frame, which layer of which multicast group should be multicasted at which

link rate level. The authors model this problem as an optimal resource allocation problem. They prove that the formulated optimization problem is NP-Hard, and present an approximation algorithm to solve it.

To attain an optimal solution for the problem investigated in [25], Li et al. [44, 45] formulate the same problem as a knapsack problem. By employing dynamic programming, a pseudo-polynomial algorithm is proposed to optimally solve the resource allocation problem for single multicast session. The proposed algorithm, however, fails to efficiently solve the maximization problem with multi-sessions multicat. The computational complexity grows quadratically with the number of available time slots. To reduce the computational complexity especially for the case of multi-sessions, a fully polynomial time approximation algorithm is presented [45]. The approximation factor, however, linearly decreases with the number of multicast sessions.

Most recently, MuVi [81] has been designed to investigate the optimal multicast scheduling problem for videos encoded with I, P, and B frames. As B frames are less important that I frames and P frames, the B frames are ignored first. MuVi models the I frames and P frames optimal scheduling problem as a knapsack problem that is similar to prior works [44, 45]. A dynamic programming algorithm is designed to solve this problem optimally. The authors then propose a sub-optimal algorithm to solve the allocation problem considering B frames.

In this thesis, we also employ the mechanism of jointly adapts video quality and multicast link rate. In contrast to previous work, we focus on a scenario where each user is interested in a subset of video tiles/sessions and user interests may partially overlap. Our algorithm designed for multi-sessions multicast has a lower, more practical, running time (grows linearly with the number of time slots) than the existing optimal allocation algorithm [44, 45, 81].

2.2.2 Multicast Association Control

Unicast Association Control

A tremendous amount of early work explores association control to improve the wireless system capacity in presence of multiple deployed APs [11, 12, 24, 53]. These studies mainly focus on unicast traffic.

Instead of using the RSSI (Received signal strength indication) value as the association criteria, many new proposed heuristics define different metrics and associate each client with the AP that maximizes these metrics. These metrics typically take into account factors such as the packet error rate, the number of users currently associated with an AP, the transmission rate (correlated to RSSI), and the achievable bandwidth of a new user if it is associated with an AP. For example, Balachandran et al. [11] propose to associate new users with the AP that can provide a minimal bandwidth required by the user. If there are multiple such APs, the one with the strongest RSSI is selected. Similarly, DenseAP [53] employs a metric, where clients associate with the AP that is the least loaded and offers the best data rate.

In addition to defining the metrics for association control, load balancing and max-min fairness have been studied in [12]. The formulas indicate the strong correlation between fairness and load balancing, which inspires authors to utilize load balancing technique to achieve max-min fairness. In particular, they design a constant factor approximation to obtain max-min fair bandwidth allocation.

Multicast Streaming Association Control

We now discuss the most relevant pieces of prior work that investigates the association control for multicast streaming traffic.

Chen et al. [23] study three different objectives: *maximize the number of users*, *balance the load among the APs*, and *minimize load of APs*. Three different approximations have been proposed to achieve these objectives separately. In their work, however, each user is only allowed to subscribe a single multicast video session. To address this

shortcoming, DirCast [21], which is designed to support multiple subscription, is proposed. None of these works, however, take adaptive video streaming into consideration.

2.2.3 Contention Window Adaptation

Apart from the link rate adaptation, the impact of contention window (CW) on CSMA/CA network has received considerable research attentions during the last decade. Based on the geometric topology, the CMSA/CA networks that consist of multiple access points (operate on the same channel) can be classified into two categories: *fully connected network* and *multiple-collision domain network*.

Fully Connected Network

The Bianchi's model [15] is well known as the most prominent study on performance analysis of 802.11 network, where Markov Chain is exploited to considerably simplify the analytical model. In the analysis, the authors assumed that the number of stations is fixed, and each station operates in the saturated condition. The primary contribution of this work is the simple and scalable model that is able to precisely predict the saturation throughput under ideal channel conditions (i.e., no transmission errors or physical layer capture effects [40]). Various subsequent work [80, 48, 68, 42] extended or simplified the model by Bianchi.

Modeling the network throughput, the model by Cali et al. [18] identified the theoretical upper bounds on the MAC protocol capacity. They revealed that the CSMA network is remarkably underutilized with the standard configuration parameters, which is severer in the large-scale scenarios. Motivated by the identified inefficiency, a heuristic CW tuning mechanism is suggested to achieve the theoretical throughput limit.

Apart from the throughput issue, fairness is another important performance metric. Since a WiFi network may have different client stations with different geometric position and environment, these clients may experience different link rates. The CSMA/CA protocol, however, is designed to provide fair transmission opportunity to all clients regardless of the corresponding link rate. Hence, the use of lower link rates can significantly reduce the overall system throughput [34]. Heusse et al. [35] proposed to

dynamically control CW to counter unfairness induced by the rate asymmetry problem. These solutions [18, 35], however, are limited to nodes in a single collision domain.

Multiple-Collision Domain Network

Topologies with hidden terminals, identified by Bharghavan et al. [14] have been analytically modeled extensively. For instance, the prior work [28, 19] comprehensively analyze the two-flow asymmetric topologies. No solution, however, is proposed to remedy the unfairness problems in these topologies. A large body of prior work generalizes the model to large-scale networks [58, 77]. Due to accuracy and complexity issues, none of them could be applied to analyze or address the unfairness with asymmetric conditions. Recently, Nardelli et al. simplified the analyses and obtained a closed-form model [54]. The model, however, cannot be applied to the scenario with RTS/CTS enabled.

The work most closely related to our work are the various studies of tuning CW with hidden terminals [63, 38, 83, 76]. The primary limitation of these studies is that they fail to provide closed-form expressions to quantify the CW tuning. A comprehensive set of experiments is conducted by Nardelli et al. [55] to evaluate existing optimal CSMA approaches [38] to counter unfairness. The results also show that these solutions could only prevent starvation, but cannot ensure fairness. The work presented in [83] is a model based approach that explores a portion of the possible CW combinations and does not ensure that the solution is fair. As the solution is search based, it is also difficult to acquire useful intuition on how CW affects the system. Most recently, Wang et al. experimentally demonstrate that the asymmetric topology depicted in Fig. 1.2 is surprisingly common in practical mesh networks [76]. To alleviate the unfairness, Fairmesh is proposed to accurately detect the asymmetric conditions and achieve approximate max-min fairness by employing a CW tuning mechanism. In this thesis, we propose a complementary theoretical model to precisely determine CW sizes to achieve different notions of fairness.

Compared with these existing works, our model (in Chapter 5) provides closed-

form expressions that are simple yet accurate, yielding an adaptive CW tuning mechanism that ensures fairness under various asymmetric conditions and topologies.

2.3 Summary

In this chapter, we have discussed several prominent adaptive transmission mechanisms in wireless networks. Our work is unique in that we consider problems whereby all the above mentioned adaptive techniques (application as well as link layers) are integrated and applied in multiple APs settings. In particular, the adaptive video streaming and tiled video format are leveraged to effectively multicast HD videos. Furthermore, we jointly adapt wireless multicast link rate and association control to improve the network resource utilization with a large number of users. When multiple access points are operating on the same wireless channel, contention window adaptation could be used to improve the network fairness.

CHAPTER 3

WIRELESS MULTICAST FOR ZOOMABLE VIDEO STREAMING

In this chapter, we are concerned with wireless multicasting of zoomable video streams, which can arise in scenarios such as interactive TV or live events such as broadcasting lectures in campus [49, 31, 62], stage performances in concert, and sports in stadium (including eSports for spectating RTS games). Multicast is a natural operation for transmitting these contents, as existing studies have reported that users tend to zoom into a small clusters of regions in the video [59] with substantial overlaps in their views.

In live zoomable video streaming system [62], multiple resolution levels are available for each video stream. For a given screen pixel size, the desired resolution level of a user depends on the size of the selected region of interest (RoI). To stream efficiently, the video is broken into a grid of small, independently decodable regions, each is termed as a *tile* in this chapter. Instead of transmitting the whole frame, a minimum set of tiles covering the selected RoI with the desired resolution level is delivered.

The specific problem that we consider in this chapter is the following: given the

available time slots for video transmission and the selected RoI regions, how to determine, for each tile, at which resolution level should it be multicasted to maximize the overall utility of all users? There are two challenges in the aforementioned problem. First, the scheme has to deal with changes in both RoI and the wireless channel that affects the supported link rates. Second, the solution has to be computationally efficient and scalable (with respect to number of users/sessions, video qualities, link rate, and time horizon).

In this work, we propose a novel and efficient algorithm to optimally solve this zoomable multicast problem. Our algorithm is inspired by several recent works [44, 81] that look into the design of optimal algorithms for video multicast allocation with a focus on heterogeneous link rates. To evaluate our algorithm, we implemented the algorithm on a testbed that consists of the following key components: (i) mobile clients that support zoomable video functions, (ii) video server that supports streaming of zoomable video, and (iii) a proxy that collects client RoI requests and wireless link conditions, runs the resource allocation algorithm, and multicasts the videos obtained from the server to the clients.

The major contributions of this chapter are as follows:

- We model the zoomable video multicast problem as an optimization problem and develop an optimal algorithm that decides which resolution of which tile should be transmitted at which link rate. The proposed optimal multicast improves the average video quality by up to 12dB, 6dB, and 3dB in terms of PSNR compared with three baseline schemes, *adaptive unicast*, *adaptive multicast*, and *approximate multicast*, respectively.

- If we consider each tile as an individual video session (Chapter 4), our proposed algorithm can be applied to the optimal allocation of multi-sessions adaptive video streaming as well, and has a lower, more practical, running time (grows linearly with the number of time slots) than the existing optimal allocation algorithms [44, 81].

- We evaluate our solution on a wireless streaming testbed with up to 10 Android phones.

The rest of the chapter is structured as follows. In Section 3.1, we conduct a psychophysical study to assess the perceptual quality impairment of mixed resolutions tiling scheme. Section 3.2 states our maximization problem. We present our optimal algorithm in Section 3.3. The system implementations are detailed in Section 3.4 and performance evaluation results of our algorithm on Android platform are presented in Section 3.5. The summary is made in Section 3.6.

3.1 Perceptual Quality Assessment of Mixed-Resolutions Tiling

Although *mixed resolutions tiling* scheme saves bandwidth, the impairment to the perceived quality is still unclear. Thus, to understand if, and at what thresholds, users could notice and/or accept the difference between original video and tiled video with mixed resolutions, we conduct a psychophysical study with 50 participants, which is presented in this section.

Using the *method of limits* from psychophysics [29], we measure two perceptual thresholds – Just Noticeable Difference (JND) and Just Unacceptable Difference (JUD) – to understand the user perception about the quality of mixed-resolution tiled video. The two identified difference thresholds partition the quality degradation level (introduced by mixing tile resolutions) into the following three intervals: without noticeable quality degradation, with noticeable (but acceptable) quality degradation, and with unacceptable quality degradation.

Table 3.1: The number of pixels in each frame and each tile at different resolution levels.

level	frame	16×9 tiles	80×45 tiles
5	1920×1080	120×120	24×24
4	1600×900	100×100	20×20
3	1280×720	80×80	16×16
2	960×540	60×60	12×12
1	640×360	40×40	8×8

3.1.1 Setup

Our experiments assess the quality of mixed-resolution tiled video using three standard HD (1920×1080p) test video files, *Crowd-Run* (dense motion, 50fps), *Old-Town-Cross* (medium motion, 50fps), and *Rush-Hour* (low motion, 25fps)[1]. The configurations for constructing the mixed-resolution tiled videos are detailed in the following two subsections.

Mixing Resolution Levels

We have five resolution levels for each video file, these levels are labeled from 5 to 1 (Table 3.1). The pixels of the original video frame at five resolution levels are: 1920×1080, 1600×900, 1280×720, 960×540, and 640×360.

In the experiments, we construct mixed-resolution tiled video by mixing two resolution levels, where the higher resolution level is denoted as R_H and the lower resolution level is denoted as R_L. Specifically, given a pair of R_H and R_L, we randomly allocate resolution level R_H or R_L to each tile with equal probability. For any particular pair of R_H and R_L, we restrict the range of R_H as $3 \leq R_H \leq 5$ and the range of R_L as $1 \leq R_L \leq R_H$. Figures 3.1, 3.2, and 3.3 show the screenshots of mixed-resolution tiled video.

Tile Size

Since the aspect ratio of the test HD video frame sequences is 16:9, we break the video frames into 16×9 tiles by default. As a result, each tile size (view region size) is $\frac{1}{16\times9}$ of the entire view region. To evaluate the impact of tile size, in addition to the default configuration, we generate another set of videos where each video frame is broken into 80×45 tiles. The number of pixels for a tile at each resolution level is shown in Table 3.8.

[1] Available at http://media.xiph.org/video/derf/

(a) Tiles from HD stream

(b) Mixing tiles from stream levels 5 and 3

(c) Mixing tiles from stream levels 5 and 1

Figure 3.1: Mixing tile resolutions of Crowd-Run

(a) Tiles from HD stream

(b) Mixing tiles from stream levels 5 and 3

(c) Mixing tiles from stream levels 5 and 1

Figure 3.2: Mixing tile resolutions of Old-Town-Cross

Data Rate

The average data rate (Mbps) of mixing resolution levels 5 and R_L in tiled video with 16×9 tiles and 80×45 tiles are represented in Table 3.2 and Table 3.3, respectively.

(a) Tiles from HD stream

(b) Mixing tiles from stream levels 5 and 3

(c) Mixing tiles from stream levels 5 and 1

Figure 3.3: Mixing tile resolutions of Rush-Hour

Table 3.2: Video data rates (*Mbps*) for configurations (5, R_L) with 16×9 tiles.

	5-5	5-4	5-3	5-2	5-1
Crowd-Run	26.44	22.32	19.91	17.78	15.53
Old-Town-Cross	7.15	5.81	5.06	4.68	4.2
Rush-Hour	5.12	4.40	3.83	3.37	3.01

Since the video data rate closely depends on its motion density, the full HD version of Crowd-Run experiences the highest data rate and the test sequence Rush-Hour has the lowest data rate. Besides the motion density, the tile size is another important factor determining the video encoding efficiency. Due to the number of pixels in a finer-grained (80×45) tile is considerably smaller than in a coarse-grained (16×9) tile, encoding coarse-grained tiled video is more efficient than finer-grained tiled video (data rate), which is verified in the tables.

Regarding the bandwidth efficiency of mixing tile resolutions, the tables demonstrate that mixing tiles from resolution levels 5 and 4, the video consumes 14%-20% less bandwidth, compared to the video consisting of tiles from only level 5. Moreover,

Table 3.3: Video data rates (*Mbps*) for configurations (5, R_L) with 80×45 tiles.

	5-5	5-4	5-3	5-2	5-1
Crowd-Run	29.02	24.34	21.98	19.47	17.19
Old-Town-Cross	9.67	7.83	6.96	6.39	5.74
Rush-Hour	5.62	4.76	4.28	3.70	3.33

the bandwidth consumption can be reduced further with a smaller R_L value. For instance, around 35% bandwidth will be saved by mixing tiles from resolution levels 5 and 2 in all test sequences.

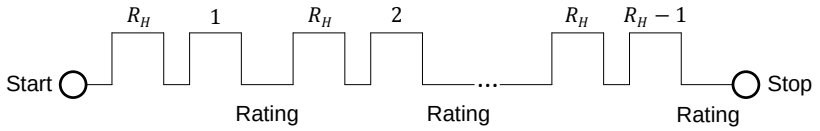
3.1.2 Procedures

Fifty adult participants were invited to participate in our assessment, primarily graduate students and research staffs from National University of Singapore. The sample consisted of 16 women and 34 men; all had normal vision. They were asked to watch the mixed-resolution tiled videos online[2] using a monitor with full HD display resolution.

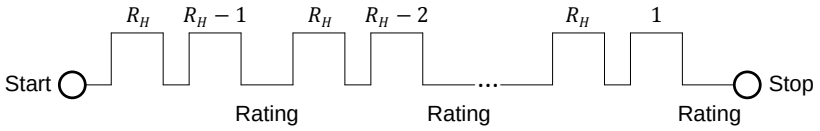
For configurations with 16×9 tiles, we vary the high resolution level R_H from 5 to 3, 9 stimuli series are generated over three test videos. For configurations with 80×45 tiles, we generate stimuli series with $R_H = 5$. As a result, we have 12 stimuli series in total, which are shuffled in a random order and played.

For each series, the stimuli is randomly manipulated in either an ascending or a descending order, the procedures are depicted in Figure 3.4. In a stimuli series, we fix the high resolution level R_H and vary the low resolution level R_L. As shown in the figure, each pair presents a standard video where $R_L = R_H$ and a mixed-resolution tiled video. After watching the videos in a pair ($10s$ per video), the participant is asked to rate the level of the difference between two videos. In particular, two questions are asked: (i) *is the quality difference noticeable* and (ii) *is the quality difference unacceptable*. In the case of ascending series, we increase R_L from 1. On each successive trial, we increase R_L by 1 until the participant eventually reports the difference is unnoticeable or

[2]Online website is available at:
http://liubei.ddns.comp.nus.edu.sg/resMix



(a) Ascending stimuli series



(b) Descending stimuli series

Figure 3.4: Experiment procedure. The video is composed by tiles with resolution level R_H and R_L. The numbers above represent the value of R_L, the first video in each pair is a standard tiled video where $R_L = R_H$, and the second video is a mixed-resolution tiled video.

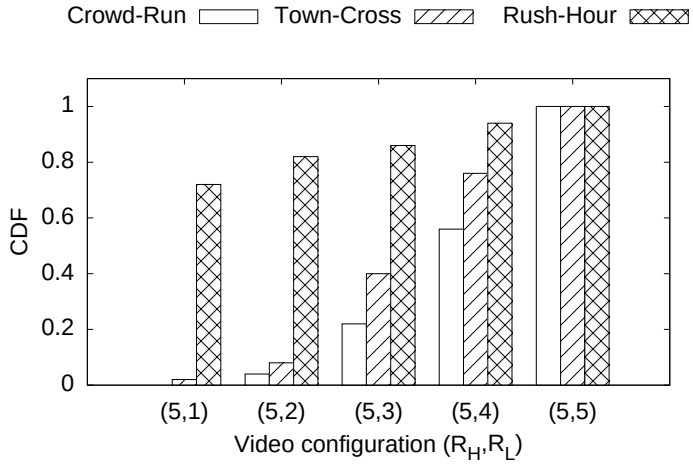


Figure 3.5: CDF distribution of participants that cannot notice any difference between mixed-resolution tiled video (5, R_L) and standard HD tiled video (5, 5).

$R_L = R_H - 1$. If the series is descending, the stimuli operates in an opposite direction. We start from $R_L = R_H - 1$ and gradually decrease R_L until the participant reports the difference is unacceptable or $R_L = 1$.

Using the above procedure, the obtained results fall into the following three categories: (i) The noticeable difference threshold and unacceptable difference threshold are both detected; (ii) Only the noticeable difference threshold is detected; and (iii) Neither noticeable difference threshold nor unacceptable difference threshold can

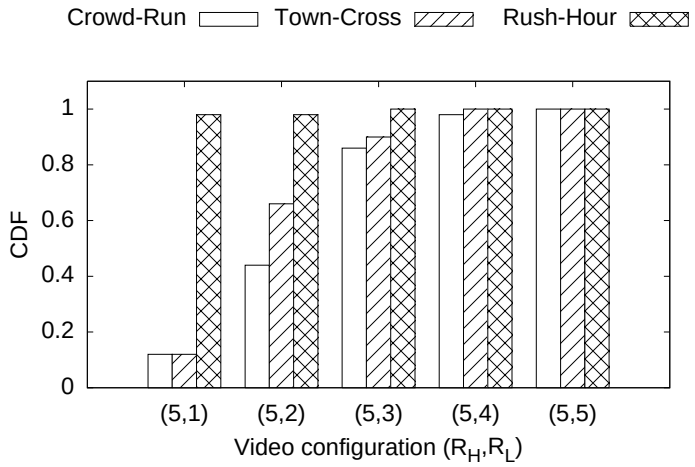


Figure 3.6: CDF distribution of participants that accept the quality difference between mixed-resolution tiled video (5, R_L) and standard HD tiled video (5, 5).

be detected. Assuming that we have detected the noticeable difference threshold and unacceptable threshold, denoted by T_{ND} and T_{UD}, respectively, then according to *the method of limits* [29], we estimate the Just Noticeable Difference threshold as $(T_{ND} + (T_{ND} + 1))/2 = T_{ND} + 0.5$. Similarly, we express Just Unacceptable Difference threshold as $(T_{UD} + (T_{UD} + 1))/2 = T_{UD} + 0.5$. For the cases where we failed to detect the difference threshold, we set the corresponding Just Noticeable/Unacceptable Difference threshold to 0.

3.1.3 Results

We first examine the configuration with 16×9 tiles. Figure 3.5 depicts the CDF distribution of participants that cannot notice any difference between mixed-resolution tiled video (5, R_L) and standard tiled HD video (5, 5). The CDF distribution of participants that accept the quality difference is present in Figure 3.6. The average measured thresholds of Just Noticeable Difference and Just Unacceptable Difference for R_H in the range from 5 to 3 are shown in Table 3.4 and Table 3.5, respectively.

From the results, we have the following observations.

Feasibility of Mixing Tile Resolutions. The measured thresholds confirm the feasibility of mixed-resolution tiled video. The CDF distribution from Figure 3.5 implies

Table 3.4: The average Just Noticeable Difference threshold (number within parenthesis is the 95% Confidence Interval value).

R_H	Crowd-Run	Old-Town-Cross	Rush-Hour
5	3.68 ($\pm$0.52)	3.25 ($\pm$0.47)	0.81 ($\pm$0.23)
4	2.74 ($\pm$0.39)	2.31 ($\pm$0.34)	0.24 ($\pm$0.10)
3	2.09 ($\pm$0.30)	1.73 ($\pm$0.26)	0.11 ($\pm$0.06)

Table 3.5: The average Just Unacceptable Difference threshold (number within parenthesis is the 95% Confidence Interval value).

R_H	Crowd-Run	Old-Town-Cross	Rush-Hour
5	2.03 ($\pm$0.31)	1.76 ($\pm$0.27)	0(0)
4	1.64 ($\pm$0.26)	1.28 ($\pm$0.21)	0(0)
3	1.28 ($\pm$0.21)	0.69 ($\pm$0.14)	0(0)

that we can mix tiles with resolution levels 5 and 4 without being noticed in most cases. Further, the depicted result from Figure 3.6 indicates that more than 85% participants accept the quality difference with configurations where $3 \leq R_L \leq R_H = 5$; under these configurations, up to 30% bandwidth can be saved by mixing tile resolutions. When we construct video from tiles at resolution level 5 and 2, almost all participants noticed the difference for video *Crowd-Run* and *Old-Town-Cross*. 40% to 65% of the participants, however, still accept the quality difference.

Impact of R_H. As expected, both the average JND threshold and the JUD threshold are positively correlated with the high resolution level R_H (Tables 3.4 and 3.5). The similar relationship is observed for the measured variations as well. The average thresholds, however, are not proportional to R_H. For instance, the average JND threshold gap of video *Crowd-Run* between $R_H = 5$ and $R_H = 4$ is 0.94, while the gap between $R_H = 4$ and $R_H = 3$ is only 0.65.

Impact of Content. With the same configuration, the results from Tables 3.4 and 3.5 show a great disparity in the measured average JND and JUD across three test videos. Overall, video *Crowd-Run*, which has the highest amount of motion among the three test videos, is most sensitive to the resolution mixing, as the highest average

threshold and the greatest variation are detected. Interestingly, video *Rush-Hour*, which has the lowest amount of motion among the three test videos, performs remarkably different from others. It is difficult to notice the quality difference between the mixed-resolution tiled video and the standard version, thus the average measured thresholds and the variations are much smaller compared with other test videos.

Gap between JND and JUD Thresholds. For many cases, although participants could notice the difference, it is still acceptable. Generally, a greater gap value indicates a higher video quality tolerance degree when the quality difference is noticeable. From the Tables 3.4 and 3.5, we observe a significant gap between the average measured JND and JUD thresholds, especially for $R_H = 5$. In particular, the average gap quantities for video *Crowd-Run* and *Old-Town-Cross* with $R_H = 5$ are 1.65 and 1.49, respectively. As the tolerance space is reduced with smaller R_H value, the size of the threshold gap between JND and JUD will be reduced as well, as can be seen in both tables.

Table 3.6: The average Just Noticeable Difference threshold where $R_H = 5$ (number within parenthesis is the 95% Confidence Interval value).

	Crowd-Run	Old-Town-Cross	Rush-Hour
16×9	3.68 (±0.52)	3.25 (±0.47)	0.81 (±0.23)
80×45	3.30 (±0.48)	3.04 (±0.44)	0.76 (±0.20)

Table 3.7: The average Just Unacceptable Difference threshold where $R_H = 5$ (number within parenthesis is the 95% Confidence Interval value).

	Crowd-Run	Old-Town-Cross	Rush-Hour
16×9	2.03 (±0.31)	1.76 (±0.27)	0(0)
80×45	1.76 (±0.29)	1.63 (±0.25)	0(0)

Impact of Tile Size. The comparison between the configurations with 16×9 tiles and 80×45 tiles is present in Tables 3.6 and 3.7. The threshold values with 80×45 tiles is slightly smaller than the corresponding threshold values with 16×9 tiles, which indicates that the quality degradation introduced by mixing resolutions is slightly less obvious for the finer-grained tile size (80×45) compared with the coarse-grained tile

size (16×9). The finer-grained tiles, however, are generally less efficient in terms of encoding and transmission bandwidth. Therefore, we need to balance the trade-off between the video quality and the efficiency to obtain an appropriate configuration.

3.1.4 Summary

The subjective assessment demonstrated that in most cases, the perceptual quality loss of mixing resolutions in tiled video is insignificant, as long as the variance of mixed resolution levels is low. From the evaluation results, we have the following two important observations:

- In most cases, tiles from 1920×1080p stream and 1600×900p stream could be mixed together without being noticed;
- Even when participants could notice quality degradation in videos combined with tiles from 1920×1080p stream and tiles from 960×540p stream, greater than 80% of participants still accept the quality difference for low and medium motion videos; and more than 40% of participants accept the quality difference for the dense motion video.

This section confirms the feasibility of *mixed resolution tiling* scheme, which will be applied to wireless multicast of tiled video streams in the rest of this chapter. Instead of randomly mixing resolutions of tiles, we now look into how to optimally allocate resolution versions to each tile to better utilize the wireless bandwidth and improve overall utilities of users.

3.2 Problem Definition

We now describe an optimization problem to determine which tile should be sent at which resolution and at which link rate, given the wireless network constraint. Let T be the number of slots available on average for the delivery of a single frame, where a slot refers to a minimum transmission time unit in 802.11 network (e.g., $9\mu s$ in $802.11a$). The wireless network supports N_r different link rates. Let n be the number of users in

our system; the physical link rates of these n clients are: $r_1, r_2, \ldots, r_n$. Without loss of generality, we assume that link rate r_i is a non-decreasing function of index i.

We generate M resolution versions (or levels) for each frame, and every frame is broken into N_v view regions. Each view region is termed as a tile (or grid). Instead of using the y-x notation in Figure 2.3, we simply number the tiles $1, 2, \ldots, N_v$ when we discuss the algorithm. A tile is considered as a logical entity – when transmitted, a tile has to have a specific resolution level. A tile v with resolution level m ($1 \leq m \leq M$) is denoted by v^m, the size of which is s_v^m. The sequence of $s_v^1, s_v^2, \ldots, s_v^M$ is strictly increasing.

The set of tiles in the RoI of user i is denoted as $\mathbf{V}(i)$. Let R_{iv} be the request resolution level of tile $v \in \mathbf{V}(i)$ from user i. In zoomable applications, the request tile resolutions from an RoI are typically identical (i.e., $R_{iv} = R_{iv'}$, for all $v, v' \in \mathbf{V}(i)$). With restricted bandwidth condition, we may not be able to satisfy all the user requests. As a result, some tiles may be streamed with resolution levels lower than the desired resolution level. To avoid significant perceptual quality loss introduced by downgrading tile resolution levels, for user i, we have a lower bound L_{iv} of resolution levels, which is guaranteed to be satisfied for tile v. More specifically, for tile v in $\mathbf{V}(i)$, the resolution level to be decoded (the highest received level) by user i should be at least L_{iv}.

Receiving v^m at user i yields utility u_{iv}^m, which follows the rules below:

- If $v \notin \mathbf{V}(i)$, then $u_{iv}^m = 0$ (for all $1 \leq m \leq M$);
- If $v \in \mathbf{V}(i)$ and $m < L_{iv}$, we have $u_{iv}^m = -\infty$;
- If $v \in \mathbf{V}(i)$ and $L_{iv} \leq m < m' \leq R_{iv}$, we have $u_{iv}^m < u_{iv}^{m'}$;
- If $v \in \mathbf{V}(i)$ and $m > R_{iv}$, we have $u_{iv}^m = u_{iv}^{R_{iv}}$.

For simplicity, we use a tile size-based utility assignment mechanism. In particular, $u_{iv}^{R_{iv}}$ is the maximum achievable utility at user i by receiving tile v, the utility assignments of receiving other levels are proportional to the corresponding tile sizes.

The utility function, however, can be any general function (e.g., the PSNR of tiles) subject to the above rules.

Given the RoI selection and the corresponding utility assignment of tiles with each resolution level at each user, the objective is to maximize the total utility received by all users subject to the total transmission slot constraint.

Lastly, we discuss the parameter settings for the average available time slots T and the tile size with a specific resolution level. All pixels belonging to the same tile across different frames will be encoded as a group of picture (GOP). Due to the dependency in a GOP, if we pick a resolution level m for a tile, we have to transmit this tile at the same resolution m for all frames within the same GOP. In our model, we therefore model s_v^m as the average tile size in a GOP and model the average number of time slots needed per frame as T. In the implementation, however, the time slots allocated to frames in a GOP is proportionally distributed according to the actual frame sizes, as there is a considerable diversity in the sizes of I, B, and P frames.

3.3 Optimal Broadcast Algorithm

This section presents a dynamic programming algorithm to solve the utility maximization problem defined in the previous section. The solution consists of three major components: (i) an algorithm that determines an appropriate quality lower bound for each user; (ii) an optimal algorithm for determining the link rate and resolution level of a single tile; and (iii) an efficient algorithm for determining the link rate and resolution level over multiple tiles.

3.3.1 Adaptive Utility Assignment

The mixture of resolution levels could result in two potential issues when the available bandwidth is insufficient to meet the requirements from all users. First, as discussed in Section 3.1, the significant disparity of resolution levels between tiles for a user may severely impair the visual perception. Next, the utility-oriented optimization algorithm could result in severe unfairness. To address these issues, we suggest an algorithm to

adaptively tune the lower bound $L_{iv}(1 \leq i \leq n)$ of resolution level that is guaranteed to be satisfied.

For a particular tile v, recall that R_{iv} is the requested resolution level from user i, and L_{iv} is the resolution level guaranteed to be satisfied for user i among its interested tiles. Given R_{iv} and L_{iv}, the rules for utility assignment are specified in Section 3.2. It is clear that when all requests of users are satisfied, we have $L_{iv} \geq R_{iv}$ for any $1 \leq i \leq n$, and the overall utility is optimal. Hence, we set $L_{iv} = R_{iv}$ at the beginning, then we validate the feasibility of current configuration for L_{iv} and adapt accordingly.

We define an indicator variable x_{iv}^m, which takes the value of 1 if resolution level m of tile v is transmitted at link rate r_i, and 0 otherwise. Let $\mathcal{M}(i,v)$ be the maximum resolution level of tile v to be received by user i. Since user i can only receive the transmissions with link rates not higher than r_i, the expression of $\mathcal{M}(i,v)$ can be written as $\mathcal{M}(i,v) = \max\left\{m | x_{i'v}^m = 1 \text{ and } 1 \leq i' \leq i\right\}$. Now we can formulate the feasibility validation problem as

$$\sum_{v=1}^{N_v} \sum_{i=1}^{n} u_{iv}^{\mathcal{M}(i,v)} \geq 0, \tag{3.1}$$

$$\text{subject to } \sum_{v=1}^{N_v} \sum_{m=1}^{M} \sum_{i=1}^{n} \left(x_{iv}^m \cdot \left\lceil \frac{s_v^m}{r_i} \right\rceil \right) \leq T, \tag{3.2}$$

where $u_{iv}^{\mathcal{M}(i,v)} = -\infty$ if $g \in \mathbf{V}(i)$ and $\mathcal{M}(i,v) < L_{iv}$; the unit of expression s_v^m / r_i is a 802.11 time slot. To obtain an appropriate setting of L_{iv}, we keep decreasing L_{iv} by 1 for all i until Inequality (3.1) is feasible subject to time limit constraint (3.2).

To solve the feasibility problem defined above, we first independently calculate the minimum required time slots for every tile v $(1 \leq v \leq N_v)$ and then simply integrate the required time slots across all N_v tiles. The total required slots should be less or equal to T, if the current lower bound requirement (L_{iv}) is achievable. The following paragraph presents an algorithm to calculate the minimum required time slot for any single tile v.

For user i, the lower bound requirement of resolution level L_{iv} can be satisfied by

either transmitting at link rate r_i or at lower link rate $r_{i'}$, where $1 \leq i' \leq i$. Define $T_v(i,l)$ as the minimum required time slots satisfying non-negative utility requirement with users up to i and with resolution level l has not been satisfied from users with indexes larger than i. The recursive equation for $T_v(i,l)$ can be written as

$$T_v(i,l) = \begin{cases} \min\left\{T_v(i-1,H), T_v(i-1,0) + \left\lceil \frac{s_v^m}{r_i} \right\rceil \right\}, & \text{if } v \in \mathbf{V}(i); \\ T_v(i-1,l), & \text{if } v \notin \mathbf{V}(i), \end{cases} \tag{3.3}$$

where $H = \max\{l, L_{iv}\}$. The minimum time slots required for delivering tile v while satisfying the quality lower bound is $T_v(n,0)$, which could be easily calculated by leveraging recursion (3.3). Now we are able to simplify the feasibility validation problem to $\sum_{v=1}^{N_v} T_v(n,0) \leq T$.

3.3.2 Optimal Allocation for a Single Tile

For ease of analysis, we begin with designing an optimal resource allocation algorithm for a single tile. We denote this particular tile as v. The optimal allocation approach determines *the resolution levels of tile v to be transmitted* and *the link rate for each transmission*.

Optimal Allocation Algorithm

Let t ($0 \leq t \leq T$) be the total slots available for the transmissions of tile v. The utility optimization problem can be formulated as

$$\begin{aligned} &\text{maximize} \sum_{i=1}^{n} u_{iv}^{\mathcal{M}(i,v)}, \\ &\text{subject to} \sum_{m=1}^{M} \sum_{i=1}^{n} \left(x_{iv}^m \cdot \left\lceil \frac{s_v^m}{r_i} \right\rceil \right) \leq t. \end{aligned} \tag{3.4}$$

As we assume that the users with higher link rate can receive all transmissions at lower rates, we have the following important observation: *for any tile, a higher resolution version is always transmitted with higher link rate*. By utilizing this observation,

we have the following definition of the maximum utility function. For tile v, define $\mathcal{U}_v(i,m,t)$ as the optimal utility with users $u_1, u_2, \ldots, u_i$, with resolution levels up to m, and within transmission time limit t.

Every state $\mathcal{U}_v(i,m,t)$ falls into category of either *user i is not interested in tile v* or *user i is interested in tile v*. If user i is not interested in tile v ($v \notin \mathbf{V}(i)$), the state transition equation could be simply written as

$$\mathcal{U}_v(i,m,t) = \mathcal{U}_v(i-1,m,t). \tag{3.5}$$

It is slightly more complicated to analyze the transitions of state $\mathcal{U}_v(i,m,t)$ when user i is interested in tile v. There are two transition possibilities for this state:

(i) if the resolution level m of v is not transmitted, the recursive function is

$$\mathcal{U}_v(i,m,t) = \mathcal{U}_v(i,m-1,t). \tag{3.6}$$

(ii) If the resolution level m is transmitted at link rate level $r_{i'}$ ($i' \leq i$), the recursive function is

$$\mathcal{U}_v(i,m,t) = \max_{1 \leq i' \leq i} \left\{ \mathcal{U}_v(i'-1, m-1, t - \left\lceil \frac{s_v^m}{r_i} \right\rceil) + \sum_{c=i'}^{i} u_{cv}^m \right\}. \tag{3.7}$$

The terminating conditions for the recursion and the corresponding value assignments are

$$\begin{aligned} \mathcal{U}_v(i,m,t) &= -\infty, && \text{if } t < 0 \text{ or } m < 0; \\ \mathcal{U}_v(0,m,t) &= 0, && \text{if } t \geq 0 \text{ and } m \geq 0. \end{aligned}$$

We start the recursion from state $\mathcal{U}_v(n,M,t)$ with the given available time slots t, the highest resolution level M, and user n with the highest link rate. The recursion can be solved by applying Eqs. (3.5), (3.6), and (3.7). The transition complexity for Eqs. (3.5) and (3.6) are both $\mathcal{O}(1)$. Eq. (3.7) enumerates the user link rate for every

transmission to attain the optimal transition. As a result, the transition complexity for Eq. (3.7) is $\mathcal{O}(n)$. The overall computational complexity of our optimization algorithm is $\mathcal{O}(n^2tM)$, which grows quadratically with n.

Virtual Clustering

This section applies a clustering method to make our optimal algorithm scalable with n (number of users). Since Eq. (3.7) is the most time consuming operation, we will concentrate on analyzing this equation.

Assuming that a specific link rate $r_{i'}$ is used for transmitting resolution level m in Eq. (3.7), all clients with no smaller than link rate $r_{i'}$ are able to receive this resolution level. Instead of enumerating user i', only the distinct link rates are required to be considered. As a consequence, we could cluster the users with identical link rate to a virtual user in the algorithm. The clustering process is achieved by simply integrating the corresponding utility values. Specifically, the utility of tile v at resolution level m for a virtual user at link rate r could be defined as $\sum_{i=1}^{n} u_{iv}^{m}$, where $r_i = r$.

By clustering, the number of users n is reduced to at most N_r, which is the maximum number of distinct link rates. As the number of link rate levels is noticeably small (8 in 802.11a [7]), with user clustering, our algorithm scales with any number of users without considering the frame losses and retransmissions.

3.3.3 Optimal Allocation for Multiple Tiles

This section presents an algorithm that is able to achieve the maximum utility by optimally allocating resources over all N_v tiles. First, we extend the algorithm in Section 3.3.2 to incorporate multiple tiles. Next, we analyze the computational complexity of the algorithm and demonstrate its inefficiency. Finally, we reduce the computational overhead of the algorithm to make it more efficient and practical for deployment.

Given time limit $t(v)$ for tile v, the optimal utility is $\mathcal{U}_v(n, M, t(v))$, which is calculated in Section 3.3.2. The overall system utility is the integrated utility over all

N_v tiles, the optimization problem can be represented as

$$\begin{aligned} &\text{maximize} \sum_{v=1}^{N_v} \mathcal{U}_v(n, M, t(v)), \\ &\text{subject to} \sum_{v=1}^{N_v} t(v) \leq T. \end{aligned} \tag{3.8}$$

From the formulas, we observe that optimization problem (3.8) is to optimally distribute the total time slots T to all tiles.

Define function $\mathcal{U}(v,t)$ as the maximum utility achieved with tiles from 1 to v within time limit t. Enumerating the allocated time slots t' for transmissions of tile v yields

$$\mathcal{U}(v,t) = \max_{0 \leq t' \leq t} \{\mathcal{U}(v-1, t-t') + \mathcal{U}_v(n, M, t')\}. \tag{3.9}$$

The maximum system utility is $\mathcal{U}(N_v, T)$. This equation is employed by Li et al. [44, 45] as well to incorporate the allocation of multiple multicast sessions into their optimal algorithm.

We now discuss the complexity of this multiple tiles allocation algorithm. We precomputed all $\mathcal{U}_v(n, M, t)$, where $1 \leq v \leq N_v$ and $0 \leq t \leq T$, the complexity is $\mathcal{O}(n^2 T M N_v)$. As shown in Eq. (3.9), the transition complexity for each state is $\mathcal{O}(T)$, the complexity of the recursion procedure to calculate $\mathcal{U}(N_v, T)$ is $\mathcal{O}(T^2 N_v)$. Combining the precomputing and the recursion complexity gives $\mathcal{O}(n^2 T M N_v + T^2 N_v)$ in total.

The parameters of n (reduced to N_r), M, and N_v are constants for a given video, so the computational cost depends on T. Assuming that the video frame rate is $25fps$, the slots available on average for a single frame is $40ms \approx 4444$ slots ($9\mu s$ per slot in $802.11a$). When this value of T is substituted into $\mathcal{O}(n^2 T M N_v + T^2 N_v)$, the overhead is clearly too large to be practical. Therefore, it is essential to further reduce the computational complexity.

The key idea of reducing computational overhead is to trade space for algorithm

running time. Define the optimal utility function $\mathcal{U}^*(v,i,m,t)$ as

$$\mathcal{U}^*(v,i,m,t) = \max_{0\le t'\le t} \{\mathcal{U}(v-1,t-t') + \mathcal{U}_v(i,m,t')\}. \tag{3.10}$$

Same as the analysis for the allocation algorithm of a single tile, the category that each state $\mathcal{U}^*(v,i,m,t)$ falls into depends on whether user i is interested in tile v.

If user i is not interested in tile v, substituting Eq. (3.5) into Eq. (3.10) yields

$$\begin{aligned}\mathcal{U}^*(v,i,m,t) &= \max_{0\le t'\le t} \{\mathcal{U}(v-1,t-t') + \mathcal{U}_v(i-1,m,t')\} \\ &= \mathcal{U}^*(v,i-1,m,t).\end{aligned} \tag{3.11}$$

On the other hand, if user i is interested in tile v, by substituting Eqs. (3.6) and (3.7) into Eq. (3.10), we attain Eq. (3.12).

$$\begin{aligned}&\mathcal{U}^*(v,i,m,t) \\ &= \max_{0\le t'\le t} \left\{\mathcal{U}(v-1,t-t') + \max\left[\mathcal{U}_v(i,m-1,t'), \max_{1\le i'\le i}\left(\mathcal{U}_v(i'-1,m-1,t'-\left\lceil\frac{s_v^m}{r_{i'}}\right\rceil) + \sum_{c=i'}^{i} u_{cv}^m\right)\right]\right\} \\ &= \max\left\{\mathcal{U}^*(v,i,m-1,t), \max_{1\le i'\le i}\left[\mathcal{U}^*(v,i'-1,m-1,t-\left\lceil\frac{s_v^m}{r_{i'}}\right\rceil) + \sum_{c=i'}^{i} u_{cv}^m\right]\right\}.\end{aligned} \tag{3.12}$$

The initial conditions and recursive transitions at boundaries for $\mathcal{U}^*(v,i,m,t)$ are

$$\begin{aligned}\mathcal{U}^*(v,i,m,t) &= -\infty, && \text{if } t<0 \text{ or } m<0;\\ \mathcal{U}^*(v,0,m,t) &= \mathcal{U}^*(v-1,n,M,t), && \text{if } v\ge 1, t\ge 0, m\ge 0;\\ \mathcal{U}^*(0,i,m,t) &= 0, && \text{if } t\ge 0, m\ge 0.\end{aligned}$$

The recursive Eqs. (3.11) and (3.12) clearly illustrate the procedure to solve the optimal multiple tiles allocation problem. The maximum utility is $\mathcal{U}^*(N_v,n,M,T)$.

The transition Eq. (3.11) consumes $\mathcal{O}(1)$ complexity. Eq. (3.12) enumerates user id i' instead of time slots, thus the transition complexity is $\mathcal{O}(n)$. Taking all transitions into consideration, we have a total computational complexity of $\mathcal{O}(n^2TMN_v)$. Here, n can be replaced by N_r by clustering users according to the available link rate levels.

Compared with previous multiple tiles allocation algorithm, the computational complexity of current algorithm is significantly reduced by a factor of T. In the evaluation section, we will demonstrate the effectiveness of our optimal algorithm.

3.4 Experimental Setup

To evaluate our algorithm, we setup the following experimental system.

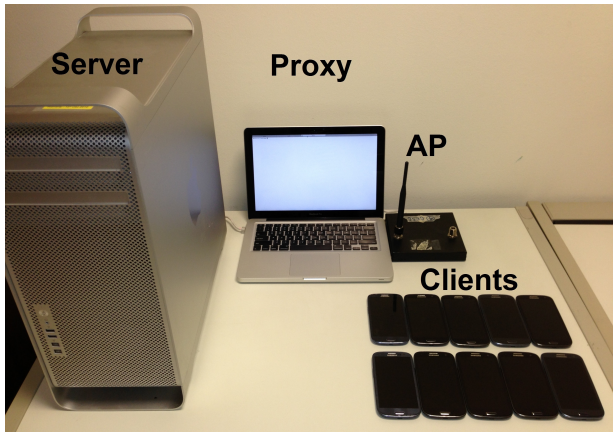


Figure 3.7: System Setup.

3.4.1 System Setup

Our system uses a zoomable video streaming server that runs on a Mac Pro with a 3.2GHz Quad-Core processor and 8GB memory. The proxy runs on a MacBook with a 2.9GHz dual-core processor and 8GB memory. The video server, proxy, and WiFi AP used for multicast are all connected through wired Ethernet. The mobile devices, all Samsung Galaxy SIII, communicate with the AP using IEEE 802.11a operating at 5GHz.

The AP used supports two Complex IEEE 802.11abg adapters featuring the Atheros AR5414 chipset and runs OpenWRT Kamikaze 7.09 with kernel version 2.6.25.16. The driver of the wireless adapter used is MadWifi (version 0.9.4). To enable packet level rate assignment, we use the Click modular router [41] (version 1.6.0). For each video packet transmission, we extract the rate value that is specified by the proxy in the header of every video packet, then passes the assigned rate value to the MadWifi driver. The

setup is shown in Figure 3.7.

3.4.2 Rate Adaptation

As the WiFi SNR values on the mobile devices are not available, we use frame loss as a basis for rate adaptation [17, 79, 56]. In particular, we implement History-Aware Robust Rate Adaptation Algorithm (HA-RRAA) [56] that extends the work of RRAA [79].

RRAA uses two parameters, *Maximum Tolerable loss* (MTL) and *Opportunistic Rate Increase* (ORI), for rate adaptation. The corresponding threshold for these parameters are denoted by P_{MTL} and P_{ORI}, where $P_{ORI} < P_{MTL}$. RRAA measures the frame loss rate P over a period of *Estimation Window* and adapts the link rate as follows. The rate decreases to next lower one if P is greater than P_{MTL}. If P is smaller than P_{ORI} the rate is increased to next higher one. When P is between P_{MRL} and P_{ORI}, the current rate is retained.

To limit transmissions at the adjacent high loss rates, HA-RRAA is suggested [56]. HA-RRAA exponentially increases the window size of next lower rate upon transmission failure of current rate ($P > P_{MTL}$) and reset the window size when transmissions of current rate are successful ($P < P_{MTL}$). To be responsive to fast channel deterioration as RRAA, the algorithm additionally computes the loss over a *small window*. When the loss rate over this small window is greater than P_{MTL}, the current rate is directly moved to the next lower rate.

From our experiments, we observe that the HA-RRAA tuning mechanism may still result in the oscillation between two adjacent rates. We slightly modify the algorithm so that the window size is halved instead of being reset when transmissions of the current rate is successful. Furthermore, since we may broadcast packets at different rates under heterogeneous links, a client may receive packets sent at a rate higher than its current rate – these packets serve as "free" probes that prevent a client from increasing its rate unnecessarily. As a result, our rate adaptation is stable and responsive.

For tractability, packet losses and frame retransmissions are not incorporated into

our algorithm. Therefore, conservative threshold parameters are used in our work. In particular, we set $P_{MTL} = 10\%$ and $P_{ORI} = 3\%$. The minimum *Estimation Window* size equals the interval between two consecutive allocation algorithm runs, this interval is also used as the *small window* to maintain responsiveness.

3.4.3 Video Coding and Streaming

In the evaluation, we do not need to play the video on the mobile devices and hence do not send actual video data. Instead, the following is done.

As depicted in Figure 2.3, each raw video frame from the test video is broken into N_v tiles, and the tiles with same y-x are encoded using FFmpeg tool (version 1.2.1) with H264 codec at the server. During our experiments, instead of transmitting the corresponding tiles from the test video, the server simply transmits the same number of arbitrary bits as the actual video tile. The metadata containing the tile size, y-x position, resolution level, and the frame ID for identification, is embedded. A client running on the mobile device extracts these fields from each received tile and periodically provide the reception bitmap to the server. When the transmission is over, we gather the reception bitmaps from all the clients, and reconstruct the mixed-resolutions video frames with decoded tiles at the server side. Here, the lost tiles (indicated by bitmaps) in a group of pictures (GOP) are concealed by the default method in FFmpeg.

3.5 Evaluation

In this section, we present the evaluation results of our proposed optimal multicast algorithm through extensive experiments using up to 10 mobile devices.

Compared Algorithms: We compare performance of optimal multicast (oMulticast) against the following baseline schemes. These schemes use HA-RRAA link adaptation as well.

Adaptive Unicast (aUnicast): This scheme transmits packets using wireless unicast only. To ensure the lowest quality (resolution level 1) is received by every user, the algorithm calculates the number of time slots required to transmit every tile at resolution

level 1. The algorithm then loops through each user, and if there is sufficient available time slot remaining, the resolution of the tiles transmitted to the user is replaced by the desired resolution level. The loop terminates when the requests from all users are satisfied or the remaining time slots are insufficient for any user.

Adaptive Multicast (aMulticast): Similar to aUnicast, the lowest resolution level 1 is guaranteed for each user and the remaining available slots are utilized to upgrade the resolution level tile by tile. As in DirCast [21], the assigned link rate for a particular tile is the lowest supported link rate among all interested users. As multicast is used, at most one multicast transmission is required for any tile.

Approximation: We apply the approximation method in [45] to our maximization problem, where the utility slots instead of the time slots is used as a state dimension in the dynamic programming. The approximation factor bound of this approach is $1-\varepsilon N_v$. A better approximation factor is obtained with a finer-grained utility unit (a smaller ε). As the computational complexity of the approximation algorithm grows quadratically with the number of utility units, the finer-grained utility unit significantly increases the computational complexity. In our experiment, the same $\varepsilon = 0.2$ is used, and the running time is close to our optimal multicast.

In our work, all the above algorithms collect the RoI requests and run the allocation algorithm every 2 seconds. The average running time of our optimal algorithm is 49.18ms, which only incurs 2.5% overhead.

We measure the peak signal-to-noise ratio (PSNR), a standard metric for measuring the video quality, and *goodput* of the system to compare the performance of the algorithms.

Video Setup: We evaluate the algorithms using two standard HD (1920x1080p) test video files, *controlled-burn* (dense motion) and *rush-hour* (low motion)[3]. Table 3.8 presents the video configurations and data rates.

Wireless Channels: We place the mobile devices at different locations and distances from the AP, to vary the channel conditions between the mobile devices and the

[3] Available at http://media.xiph.org/video/derf/

Table 3.8: The data rate ($Mbps$) of different resolution levels.

level	resolution size	# tiles	low rate[a]	medium rate[b]	high rate[c]
5	1920×1080	16×9	6.2	10.9	20.2
4	1600×900	16×9	4.5	6.6	11.1
3	1280×720	16×9	3.2	4.6	8.4
2	960×540	16×9	2.2	2.9	5.0
1	640×360	16×9	1.2	1.5	2.5

[a] *Rush-hour*, compressed using FFmpeg with parameter $qp = 25$.
[b] *Controlled-burn*, compressed with $qp = 25$.
[c] *Controlled-burn*, compressed with $qp = 22$.

AP. Table 3.9 shows the minimum, maximum, and average achieved link rates when there are up to 10 mobile devices.

Table 3.9: The achieved link rates of mobile users ($Mbps$).

# users	min rate	max rate	average rate
1	6	6	6
3	6	36	20.0
5	6	36	21.6
8	6	36	22.5
10	6	36	21.0

RoI Variation: User requests and RoI used in the evaluations are based on the real interaction logs from 10 users who have used zoomable video system [59].

3.5.1 Baseline Comparison

The average PSNR with error bars (standard deviation) across different users streaming at medium video rate are depicted in Figure 3.8. The corresponding achieved average goodput is present in Table 3.10. As the unicast scheme cannot fit the lowest resolution level requirement for more than 5 clients, no data point is presented in this range in the results. From the results, we can draw the following observations:

(i) *PSNR gains*. The multicast algorithms are able to satisfy up to 5 users without notable PSNR degradation. On the other hand, the video quality with unicast dramat-

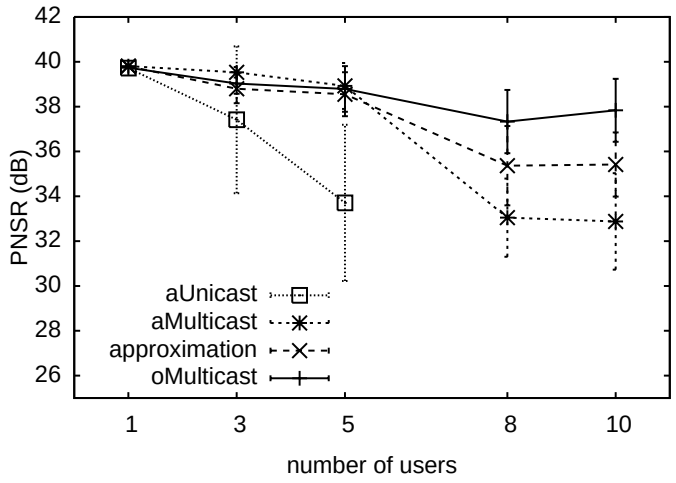


Figure 3.8: Average PSNR with medium video rate.

Table 3.10: Average goodput ($Mbps$) achieved with heterogeneous link qualities at medium video rate

# users	aUnicast	aMulticast	approximation	oMulticast
1	3.83	3.79	3.81	3.82
3	2.95	3.45	3.46	3.41
5	1.8	3.07	3.05	3.07
8	\	2.1	2.27	2.56
10	\	1.99	2.25	2.67

ically decreases beyond 3 users, and only up to 5 users can be supported by adaptive unicast. With more than 5 users, all three multicast schemes experience some PSNR loss. The optimal multicast, however, considerably outperforms approximation and adaptive multicast under heavy load, with the improvements of about 3dB and 5dB in PSNR, respectively.

(ii) *Goodput gains*. Due to zooming, the demands between different clients are not identical. Hence, the trend in average goodput does not strictly follow that of video quality (Table 3.10). As predicted from Figure 3.8 and Table 3.10, the multicast algorithms outperform unicast when there are more than 3 users in terms of both PSNR and goodput. When there are more than 5 users, the improvements of optimal multicast over approximation and adaptive multicast with 10 users are 19% and 34%, respectively.

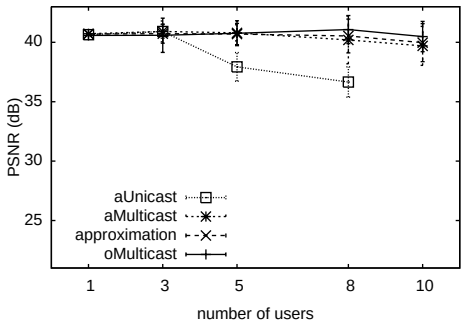


Figure 3.9: Average PSNR with low video rate.

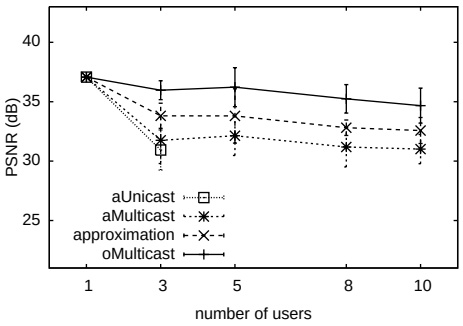


Figure 3.10: Average PSNR with high video rate.

(iii) *Fairness gains.* The error bars in Figure 3.8 indicate that our optimal multicast achieves the best fairness among all algorithms, due to adaptive utility assignment (Section 3.3.1) in our algorithm. Although a similar allocation method is used by adaptive multicast and adaptive unicast, they performs remarkably different in terms of fairness. While multicast transmission can benefit multiple users, unicast transmission does not, which may lead to less fairness among the users.

3.5.2 Impact of Video Rate

To evaluate the impact of video data rate (and thus the traffic load), we repeat the experiments using a different video with a lower rate and the same video encoded with a higher rate. We generate *low rate* and *high rate* videos in addition to the previously used *medium rate*. The configurations are detailed in Table 3.8. Wireless link quality settings are the same to the previous section. Figure 3.9 and Figure 3.10 depict the average achieved PSNR for low rate and high rate videos, respectively.

Figure 3.9 demonstrates that all four algorithms perform better with lighter workload as expected. Specifically, the multicast algorithms scale up to 10 users without significant quality degradation, and the unicast scheme is able to support more clients.

For higher traffic load, all algorithms perform worst. Compared with other schemes, our optimal algorithm, however, still provide relatively fair quality under the higher load. In general, if a client does not induce lower link rate or request higher resolution

level, no additional multicast traffic will be introduced. Thus, the video qualities are only slightly reduced even as more clients are added to the multicast sessions.

3.5.3 Impact of RoI Similarity

Intuitively, larger amount of RoI overlapping increases the relative performance gap of multicast over unicast. The impact of RoI overlapping is evaluated in this section. In order to control the amount of overlap, we do not use collected traces to simulate RoI variation. Instead, we manually vary the RoI sizes and positions so that they can change in a uniform and controlled manner. Here, the RoI sizes and the request resolution levels of all clients are identical. We vary the positions of RoI to generate different similarity.

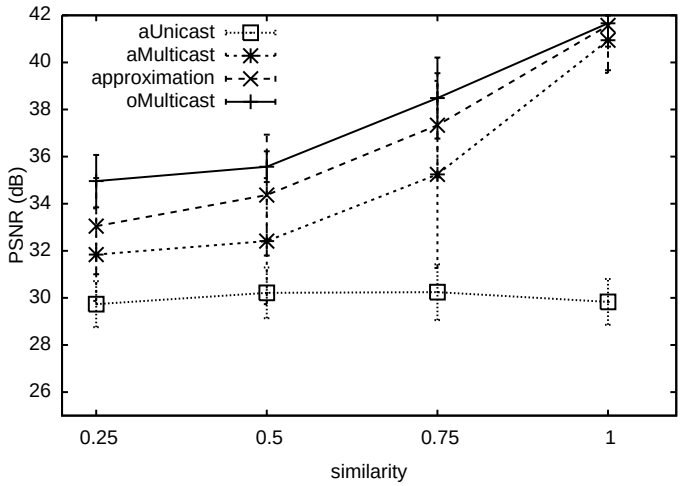


Figure 3.11: Average PSNR with different similarity.

To measure the degree of overlapping, we first define the *popularity* of a tile v, p_v as the fraction of the number of users interested in it. The degree of overlapping for user i is then the total popularity of all tiles in $\mathbf{V}(i)$, excluding the tiles only interested by user i, divided by the number of tiles in $\mathbf{V}(i)$. We then define *similarity* as the average overlapping degree across all users. We present how PSNR changes with different similarity, for 8 users, in Figure 3.11.

The relatively stable performance in terms of video quality shows that the unicast scheme is not affected by the amount of RoI overlap. As expected, the improvement of multicast over unicast increases with the increasing RoI similarity. When the RoIs are

identical (all users want the same regions), the improvement is about 12dB in PSNR. Interestingly, with increased similarity value, the PSNR quantities of three multicast algorithms converge to an identical point. Such convergence is caused by both the decrease in traffic demand and the fact that the same data is requested.

3.5.4 Client Mobility

The previous sections demonstrate the effectiveness of our optimal multicast algorithm with stationary clients. In this section, we evaluate the performance of our optimal algorithm with client mobility. In particular, we keep two clients static, the obtained link rates for them are 6Mbps and 36Mbps. One additional mobile client starts from a location close to the AP, moves away from it, and then moves back. Figure 4.6 plots the average PSNR of the mobile client for every two seconds. The movement period is from 40s to 120s.

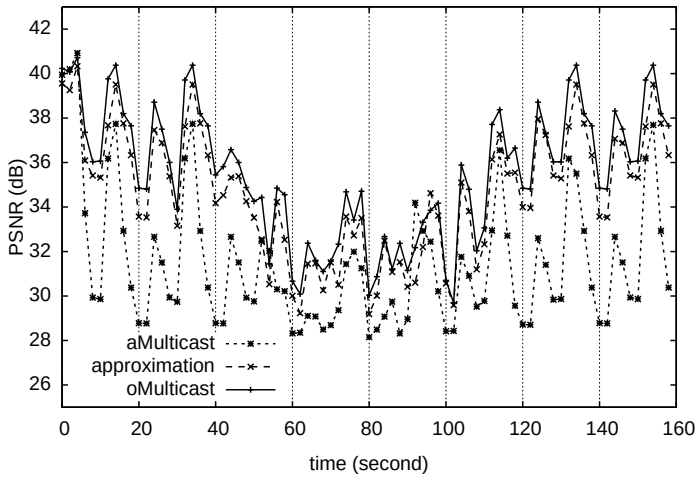


Figure 3.12: Average PSNR of the mobile client.

In the experiment, the high rate video is used and a segment of 20s is played repeatedly. Although, the RoI of each user and the allocations are fixed under static condition, the PSNR of different frames are different. This disparity is due to the fact that sensitivity of different frames with mixed resolution tiles are different. The same trend of PSNR variations under static conditions can be observed between different playbacks.

From the figure, we observe that our optimal algorithm consistently outperforms two baseline algorithms. The average enhancements of our optimal multicast over approximation and adaptive multicast are about 1dB and 4.5dB, respectively. Moreover, our algorithm can quickly adapt to the link rate and the video quality returns quickly to the level similar to the static period after the movement (at 120 second).

3.6 Summary

In this chapter, we have developed and implemented an efficient algorithm for multicasting mixed resolution tiles to heterogeneous users, for interactive video applications that support zoom and pan. Our algorithm optimizes the total utility of all clients and achieves significant improvements in video quality: up to a 3dB improvement over approximation multicast approach, 6dB improvement over an adaptive multicast scheme, and 12dB improvement over adaptive unicast scheme in our experiment settings. Additionally, our approach can be directly applied to design an optimal allocation algorithm for a general multi-sessions video multicast. Next, we shall extend this work to the scenarios with multiple access points (APs), where the AP association mechanism could be exploited to further enhance the multicast performance.

CHAPTER 4

ADAPTIVE VIDEO MULTICAST OVER MULTIPLE ACCESS POINTS

The previous chapter has designed and implemented an algorithm to optimally allocate multicast transmissions for mixed resolution tiled video. In this chapter, we generalized tiled video multicast problem to multi-sessions multicast, where each user subscribes to a set of video sessions. Moreover, to increase network capacity, multiple access points are deployed.

Video broadcast over 802.11 wireless networks with multiple deployed APs, however, is challenging because of the conflicts between high transmission rate (association control), load balancing, and exploiting multicast opportunities. More specifically, if a client simply chooses the AP with the highest receiving Received Signal Strength Indicator (RSSI), this could result in severe unbalanced workload between APs and reduce the multicast opportunities.

In this chapter, we present JurCast, a joint user and rate allocation scheme for video multicast over multiple APs. In particular, the following problem is considered: given a set of heterogeneous clients in the system, the set of videos interesting to each client (multiple subscriptions are allowed), and the estimated link condition between each client and each AP, how to determine (i) the client to AP association, (ii) the resolution level of each interested video to be delivered to each client, and (iii) the multicast transmission rate for each video version. Our goal is to maximize the overall perceived video quality of all clients. To this end, we first build a novel model to characterize this maximization problem, and then propose a heuristic algorithm to effectively solve the formulated problem.

To summarize, our key contributions are:

- A model that jointly characterizes the association schedule and the multicast resource allocation to maximize the overall system utility;
- A methodology to simplify the model and based on the simplified model, we suggest an effective heuristic to solve the maximization problem;
- Evaluation of the proposed heuristic, including both system implementation and large scale simulation, to show that, compared to the baseline schemes, our approach significantly improves the video quality (PSNR) and goodput by up to 3dB and 55%, respectively.

The rest of this chapter is organized as follows. In Section 4.1, we present the overall system architecture and our proposed algorithm. In Section 4.2, we describe the implementation details of our testbed. The evaluation is presented in Section 4.3. We summarize in Section 4.4.

4.1 JurCast Design

The architecture of our JurCast comprises one or multiple video servers, a gateway (or central controller), multiple WiFi access points, and a set of wireless clients, as depicted in Figure 4.1. The basic operations of the streaming system are as follows:

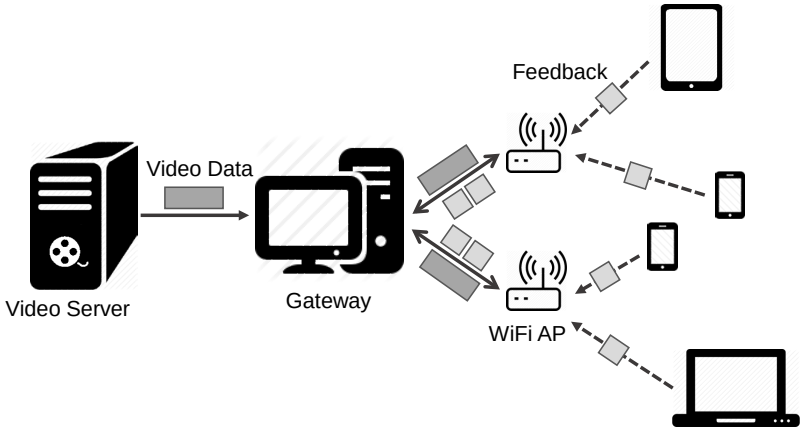


Figure 4.1: JurCast architecture, which consists of one or multiple video servers, a gateway, multiple WiFi APs, and a set of wireless clients.

1. To join the system, a client sends a join request to the gateway through the associated AP. The request message contains the interested video session ids, the corresponding resolution levels requested, and the detected *Received Signal Strength* (RSS) values of all APs.

2. Upon receiving the joining request from a new client, admission control is performed at the gateway: if the lowest guaranteed video quality cannot be satisfied for some client, the request from this new client will be declined. Otherwise, the gateway runs our algorithm to determine: (i) the "best" AP to associate with for each client; (ii) the resolution versions of each video to be multicasted from which AP; and (iii) the physical link rate used for each multicast transmission, which determines the destined clients of the transmission.

3. Once the above allocation is determined, the gateway will fetch the corresponding video data from the video server and then distribute the video data to APs for multicast.

4. The clients will send bitmap feedback periodically, which indicates the received packets for the past unacknowledged transmissions. The feedback information will be leveraged to accurately estimate the link rate level between the client and

the AP.

In addition to the above operations, the gateway continuously collects the update information (e.g., updates of the interested videos and wireless link quality) from all clients. The re-evaluation is carried out when clients join, update, or leave the system.

As described above, the gateway is the core component of our system, which consists of: (i) collecting the clients information and feedbacks; (ii) running the algorithm to effectively allocate resource for the multicast transmissions; and (iii) fetching data from the video server and distributing it to the destined APs.

Regarding the access points, the following two small modifications are required: (i) forward the control packets between the associated clients and the gateway; and (ii) transmit each video packet using the link rate specified in the packet header by the gateway.

In the following subsections, we present the problem formulation and the algorithm design details in the gateway component.

4.1.1 Preliminaries and Assumptions

The problem studied in this chapter is as follows. There are N_{AP} access points in the system, and we assume that the neighboring APs operate on non-overlapping channels, the same assumption that has been made in [21, 23, 53]. The available capacity of each AP is T in terms of the available time slots for multicast transmissions. The number of distinct link rates is N_r; and the number of video sessions is N_v. We have a set of n clients, along with the interested video sessions of each client, and the estimated link rates between each client and each AP. The objective of our algorithm is to allocate resource for multicast video frames to maximize the total system utility of all clients.

Video Encoding. A video sequence is partitioned into Group of Pictures (GOP) with a certain number of frames. Each GOP consists of I, P, and B frames. For simplicity, we assume that the number of frames in a GOP is fixed to be J for any video sequence, which can be easily generalized to videos with different frame rates. In adaptive video streaming system, each video is encoded into M resolution versions (or levels).

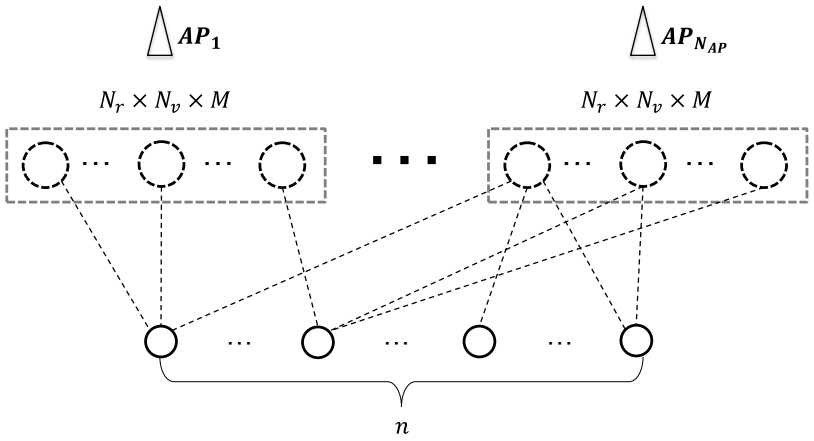


Figure 4.2: The network model. We have $N_r \times N_v \times M$ transmission states for each AP station and n client nodes. Each state is represented by the dash circle and the client node is represented by the circle with solid line.

The average frame size of video v at resolution m is denoted by s_v^m. In our model, the average analysis technique is applied to simplify the resource allocation problem. More specifically, transmitting a GOP of frames could be regarded as transmitting a single frame with the average frame size. Thus, the network capacity T of each AP is actually set as the number of 802.11 slots in $1/J$ second.

Utility Assignment. Let $\mathbf{V}(i)$ be the set of video sessions interested by client i. The resolution of video v ($v \in \mathbf{V}(i)$) requested by this client is denoted by R_{iv}. Due to the dissimilarity of videos in bandwidth consumption, popularity, and priority, the requested video resolution levels of a client could be different. Note that if video v is not in video set $\mathbf{V}(i)$, we have $R_{iv} = 0$. Since the available bandwidth is rather restricted, we may not be able to meet the requests from all clients. As a result, some videos may be streamed with resolution levels lower than the desired resolution levels. To avoid significant quality degradation, we have the lowest resolution level guaranteed to be received, which is L_{iv}. Receiving video v^m at client i yields utility u_{iv}^m. It is clear that $u_{iv}^m = 0$ for all $v \notin \mathbf{V}(i)$. For video session $v \in \mathbf{V}(i)$, the utility function should

follow the following rules:

$$
\begin{aligned}
&u_{iv}^{m} = -\infty, \text{if } m < L_{iv}, \\
&u_{iv}^{m} < u_{iv}^{m'}, \text{if } L_{iv} \leq m < m' \leq R_{iv}, \\
&u_{iv}^{m} = u_{iv}^{R_{iv}}, \text{if } m > R_{iv}.
\end{aligned} \tag{4.1}
$$

The utility function can be any general function subject to the above constraints. Here, we use the estimated PSNR as the utility function, where the highest achievable utility $u_{iv}^{R_{iv}}$ is the coding PSNR of video v at resolution level R_{iv}. For a resolution level m lower than R_{iv}, the frame with requested resolution level R_{iv} is used as the reference frame to calculate the received PSNR.

4.1.2 Problem Formulation

For each AP station, a possible *transmission state* is identified by *multicast link rate*, *video session id*, and *video resolution level*. Therefore, there are $N = N_r \times N_v \times M$ distinct transmission states in total for each AP. We build a network model to characterize our problem, which is shown in Figure 4.2. In the figure, each transmission state is represented by a dash circle. For state k ($1 \leq k \leq N$) of AP_j, the corresponding multicast link rate level, video id, and resolution level are k_r, k_v, and k_m, respectively. If the link rate between client i and AP_j is greater than k_r (i.e., $r_{ij} \geq k_r$) and this client is interested in video k_v (i.e., $k_v \in \mathbf{V}(i)$), a dash line is added between this client and virtual node k of AP_j. The attainable utility is $u_{ik_v}^{k_m}$.

The network model shown in Figure 4.2 clearly demonstrates that solving our maximization problem is to optimally determine: (i) which client should be associated with which AP; (ii) which AP should schedule which transmission states subject to the bandwidth capacity constraints; and (iii) which client should attain utility from which scheduled and associated (dash lines from the associated AP that is determined by (i)) transmission states.

Let binary variable x_{ij} indicates that client i is associated with AP_j. The constraint

that *every client is associated to exactly one AP* can be formulated by the equation: $\sum_{j=1}^{N_{AP}} x_{ij} = 1$.

We define an indicator variable y_{jk}, which takes the value of 1 if the transmission state k is scheduled at AP_j for delivery. The time cost of transmitting this particular state equals to the video size over the link rate, which can be written as $s_{k_v}^{k_m}/k_r$, where $s_{k_v}^{k_m}$ is the frame size of video k_v at resolution level k_m. As the overall multicast traffic at each AP should be subject to the capacity limit T, we have the following expression:

$$\sum_{k=1}^{N} y_{jk} \cdot \left\lceil \frac{s_{k_v}^{k_m}}{k_r} \right\rceil \leq T, \quad \text{for all } 1 \leq j \leq N_{AP}, \tag{4.2}$$

where $N = N_r \times N_v \times M$.

We have another indicator variable z_{ijk}, which takes the value of 1 if client i attains utility from the transmission state k of AP_j. Regarding the value of the binary variable z_{ijk}, we have the following two restrictions:

(i) The value of $z_{ijk} = 1$ implies that client i should be associated with AP_j ($x_{ij} = 1$) and state k is scheduled to be delivered at AP_j ($y_{jk} = 1$). This dependence relationship could be formally written as:

$$z_{ijk} \leq x_{ij} \cdot y_{jk}, \quad \text{for all } 1 \leq i \leq n, 1 \leq j \leq N_{AP}, 1 \leq k \leq N; \tag{4.3}$$

(ii) Client i attains utility from exactly one state for any particular interested video session $v \in \mathbf{V}(i)$: $\sum_{k=1}^{N} z_{ijk} = 1$, where $k_v = v$.

In summary, we can formulate our maximization problem as

$$\begin{aligned}
\text{maximize } & \sum_{i=1}^{n}\sum_{j=1}^{N_{AP}}\sum_{k=1}^{N} z_{ijk}\cdot u_{ik_v}^{k_m}, \\
\text{subject to } & \sum_{j=1}^{N_{AP}} x_{ij} = 1, \\
& \sum_{k=1}^{N} y_{jk}\cdot\left\lceil \frac{s_{k_v}^{k_m}}{k_r}\right\rceil \leq T, \\
& z_{ijk} \leq x_{ij}\cdot y_{jk}, \\
& \sum_{k=1}^{N} z_{ijk} = 1, \forall v \in \mathbf{V}(i), k_v = v.
\end{aligned} \tag{4.4}$$

which is a 0-1 integer programming problem with three variables: x_{ij}, y_{jk}, and z_{ijk}. If the constraint that every client is associated to exactly one AP is elimintated, the formulated problem can be reduced to the NP problems [22, 23]. As multiple binary variables are present in this linear integer programming formulation, it is generally unclear how to solve it efficiently. In the subsequent section, we first present a methodology to simplify the model and then suggest an efficient heuristic algorithm.

4.1.3 Greedy Algorithm

The previous section formulated the problem as a 0-1 integer programming problem with a considerable number of constraints and three binary variables, which is difficult to solve. In this section, we will show how to simplify these constraints.

Eliminate Dependency of States

At any point of time, a client only associates to exactly one AP, which leads to the dependency of scheduling transmission states. More specifically, one client only attains utility from transmission states that come from an identical AP (the associated AP), which is not captured by the model in Figure 4.2. To eliminate the scheduling dependency, we design a new network model to characterize this issue, where each state takes a set of videos (interested by a client) instead of a single video as an element.

We define $\mathcal{V} = \{\mathbf{V}(1), \mathbf{V}(2), \ldots, \mathbf{V}(n)\}$, where element $\mathbf{V}(i)$ is the set of videos interested by client i. For a particular element $\mathbf{V}(i)$, the total number of resolution level combinations is $M^{|\mathbf{V}(i)|}$, which is clearly too large to be practical as multiple subscription is allowed. Instead of enumerating all possible combinations, we use a uniform resolution level for all videos in $\mathbf{V}(i)$, and the number of combinations for element $\mathbf{V}(i)$ is reduced to M. As the request resolutions of videos in $\mathbf{V}(i)$ could be different, this design may reduce the cost efficiency. We will present how to alleviate the inefficiency in the following sections.

Now we have $N' = N_r \times |\mathcal{V}| \times M$ transmission states for each AP. Similarly, for state k ($1 \leq k \leq N'$), the corresponding video set is represented by $k_{\mathbf{V}}$ and the transmission cost is calculated by $(\sum_{v \in k_{\mathbf{V}}} s_v^{k_m})/k_r$. For a scheduled state k from AP_j, the set of clients that can benefit from this state is: $\{i | \mathbf{V}(i) \subseteq k_{\mathbf{V}}, r_{ij} \geq k_r\}$. With the pre-computed client list, the attainable utility for each state can be easily calculated as well.

Update Residual Utility

Apart from the above association restriction, another critical restriction is that client can only receive one resolution level of each interested video and attain the corresponding utility. In our new model, this restriction implies that a client can only attain utility from exactly one state that covers all interested video sessions. To avoid redundant utility counting, we calculate the residual utility for each state. Once a transmission state is scheduled for delivery, we have to update the residual utility for all the correlated unscheduled states.

Quantify Cost of AP

The last restriction is that the overall multicast traffic from each AP should not exceed the capacity. Although the new network model eliminates the states dependency, the benefit of wireless multicast is not taken into consideration. Therefore, integrating the cost of all scheduled transmission states at each AP will considerably overestimate the workload, which is mainly due to the following issues: (i) scheduling multiple states

that partially overlap in video sessions could result in redundant transmissions of the overlapped videos; (ii) to reduce the number of states, a uniform resolution level is transmitted for the videos from each state. As a result, the video transmissions of this state may not be fully utilized when the request resolution level of some video is lower than the transmitted resolution level; and (iii) if a client attains utility from a new scheduled state at a different AP, this client will re-associate to this new AP. The occupied bandwidth consumption of previous AP, however, may not be utilized.

To address the afore discussed issues, we apply a dynamic programming algorithm to accurately predict the workload for each AP instead of integrating the cost of the scheduled states. Given the existing selected transmission states, the set of clients that are associated to a particular AP_j is denoted as $\mathbf{C}_j$. To simplify notations, we relabel the client ids in $\mathbf{C}_j$ as $1, 2, \ldots, |\mathbf{C}_j|$. Without loss of generality, we assume that the physical link rate sequence $r_{1j}, r_{2j}, \ldots, r_{|\mathbf{C}_j|j}$ is non-decreasing. The union of videos covered by these selected transmission states is represented by $\mathbf{V}_j = \cup_{i \in \mathbf{C}_j} \mathbf{V}(i)$.

For client i ($1 \leq i \leq |\mathbf{C}_j|$) and interested video session v ($v \in \mathbf{V}(i), v \in \mathbf{V}_j$), given the existing scheduled states, the highest effective resolution level expected to receive is denoted by h_{iv} ($h_{iv} \leq R_{iv}$). The remaining part of this section illustrates how to calculate the minimum required time slots for AP_j while satisfying the resolution level requirement h_{iv}, for all $i \in \mathbf{C}_j, v \in \mathbf{V}(i)$.

We separately analyze the transmission time slots for each video session v ($v \in \mathbf{V}_j$). The resolution level of video v expected to be received by client i can be transmitted at link rate r_{ij} or at lower link rate $r_{i'j}$, where $1 \leq i' \leq i$. Define $T_v(i, l)$ as the minimum required time slots satisfying the requirements from clients 1 to i and at least one transmitted resolution level should be greater than l (the level required from clients with indexes larger than i). The recursive equation for $T_v(i, l)$ can be written as

$$T_v(i, l) = \min \left\{ T_v(i-1, H), T_v(i-1, 0) + \left\lceil \frac{s_v^H}{r_{ij}} \right\rceil \right\}, \tag{4.5}$$

where $H = \max\{l, h_{iv}\}$. The minimum time slots required for delivering video v while

satisfying the quality lower bound is $T_v(|\mathbf{C}_j|, 0)$, which could be easily calculated by leveraging recursion (4.5). The cost of AP_j is expressed as

$$T_j = \sum_{v \in \mathbf{V}_j} T_v(|\mathbf{C}_j|, 0). \tag{4.6}$$

Greedy Algorithm

By leveraging the simplified model in previous subsections, we develop a heuristic algorithm that greedily chooses the transmission state with the maximum cost efficiency [21, 23] in every iteration among the unselected states.

The greedy algorithm procedure is illustrated in Algorithm 1. Let $S_{j,k}$ be state k at AP_j. We define the initial set of the unselected states as $\mathcal{S} = \{S_{j,k} | 1 \leq j \leq N_{AP}, 1 \leq k \leq N'\}$.

Algorithm 1 Greedy Algorithm

1: Compute the cost $T(S_{j,k})$ and utility $u(S_{j,k})$ of state $S_{j,k}$.
2: **repeat**
3: Compute the cost efficiency $\frac{u(S_{j,k})}{T(S_{j,k})}$.
4: Let S^* be the unscheduled state with maximum cost efficiency, while $T_{j^*} \leq T$, update $\mathcal{S} = \mathcal{S} \backslash S^*$.
5: Update cost of T_{j^*} using Equations (4.5) and (4.6).
6: Update residual utility of all relevant states in $\mathcal{S}$.
7: **until** S^* cannot be found
8: Compute the optimal utility for each AP with the above determined association.

After this greedy iteration procedure, we obtain the "best" AP association arrangement for every client. With the obtained association, we leverage the optimal broadcast algorithm from Chapter 3 to calculate the optimal resource allocation scheme for each AP. The total number of states in our model is $N_{AP} \times N_r \times |\mathcal{V}| \times M$, where $|\mathcal{V}|$ ($|\mathcal{V}| \leq n$) is the number of unique elements in $\mathcal{V}$.

4.2 Implementation

In this section, we describe our system setup and the implementation details.

System Setup: Our testbed consists of the following components: (i) a video

server that runs on a Mac Pro with a 3.2 GHz Quad-Core processor and 8GB memory; (ii) a proxy runs on a typical Linux machine with 3.4 GHz Quad-Core processor and 8GB memory; (iii) two APs with IEEE 802.11abg adapters featuring the Atheros AR5414 chipset and runs OpenWRT Kamikaze 7.09 with kernel version 2.6.25.16. The driver of the wireless adapter used in MadWifi (version 0.9.4); and (iv) the mobile devices, all LG Nexus 5.

The video server, gateway, and WiFi APs are all connected through wired Ethernet. The mobile devices communicate with the associated AP using IEEE 802.11a operating at 5GHz.

Multicast Rate Adaptation: In addition to transferring data between clients and the gateway, the primary modification at the AP is to support multicast link rate adaptation. To enable the packet level rate adaptation, we significantly extend and modify the Click modular router [41] (version 1.6.0), which is installed on each AP. The multicast transmission rate of each video packet is determined by the gateway and specified in the packet header. We extract the rate value from the header and pass the assigned value to the MadWifi driver.

The above assigned multicast link rate level closely depends on the wireless link conditions between each client and the associated AP. As the signal-to-noise ratio (SNR) of each received packet is not exposed at the smartphones, we used the measure received signal strength indicator (RSSI) to roughly estimate the initial link rate and use frame loss rate that is reported periodically from each client as a basis for rate adaptation. The History-Aware Robust Rate Adaptation Algorithm (HA-RRAA) [56] is implemented and employed in our testbed. The threshold and configuration is identical to that in Section 3.4.2.

RaptorQ FEC: Since there is no MAC level retransmission for wireless multicast, the clients may not receive all packets. To overcome packet losses, the redundant FEC packets are transmitted in advance with the source video data.

To enhance the reliability of multicast services, Third Generation Partnership Project

(3GPP) introduced the use of an Application Layer FEC (AL-FEC) scheme for both of the defined Multimedia Broadcast/Multicast Services (MBMS) delivery methods. The chosen AL-FEC scheme is based on the fountain Raptor code [64] because of the higher performance of the Raptor FEC compared with existing AL-FEC codes.

The major flaw of Raptor code is the high transmission overhead, which is defined as the amount of redundant information divided by the amount of source data. To overcome the limitation of the standardized Raptor code, an enhanced Raptor code has been proposed at the Internet Engineering Task Force (IETF) [46], which is known as RaptorQ code. The key advantage of RaptorQ over Raptor codes is the low transmission overhead. The decoding failure probability of RaptorQ code can be modeled as:

$$p_f = \begin{cases} 1, & \text{if } n < k \\ 0.01 \times 0.01^{n-k}, & \text{if } n \geq k \end{cases} \tag{4.7}$$

where p_f denotes the decoding failure probability of a RaptorQ protected block with k source symbols if n symbols have been received.

In our experiment, we leverage the library provided by OpenRQ [4] that implements the RaptorQ FEC scheme described in RFC 6330. The block size is set to 256. The number of FEC packets is adaptively determined by the thresholds in link rate adaptation and the historical frame loss rate. The average encoding time cost measured over a block of 256 packets (1470 bytes per packet) is 61.27ms. As there is no coding overhead for the source symbols, we can transmit the source packets while generating the redundant FEC packets.

4.3 Evaluation

In this section, we perform a comprehensive set of experimental and simulation measurements to evaluate the efficacy of our approach. In particular, we perform two sets of evaluations. In the first set of experiments, we evaluate our approach using a wireless testbed with up to 8 Android smartphones. In the second set, we evaluate large scale scenarios through simulation of up to 50 users.

Reference Schemes: We compare the performance of our JurCast with the following two adaptive approaches.

Best-RSSI based association (Best-RSSI): This scheme employs the traditional WiFi association control mechanism where each client chooses the AP with the highest RSSI value received. With the determined association, we run the optimal broadcast algorithm suggested in Chapter 3 to obtain the optimal allocation scheme for each AP separately.

Customized DirCast (DirCast$^+$): We choose DirCast approach [21] as another baseline algorithm because it is the prominent study that addresses the similar adaptive multicast problem over multiple APs. The adaptive video quality, however, is not incorporated in DirCast. Thus, we slightly modified the algorithm of DirCast to take adaptive video resolution into account. In particular, we enumerate the resolution level for each state of DirCast, and then run their heuristic algorithm to obtain the association and multicast transmission allocations.

Clusters: In real world networks, the clients are often unevenly distributed across all the available APs. Although every client is typically covered by many APs, the clients tend to be close to some particular APs. This clustering phenomenon is common in many scenarios, such class/conference room, concert, stadium. Here, a cluster refers to *a group of clients that associate with an identical AP at the initial status*, where the Best-RSSI association mechanism is employed. In the experiments, we vary the number of clusters by generating different initial statuses.

Metrics: The perceived video quality is measured by PSNR (peak signal-to-noise ratio), which is widely used by prior works [61, 81, 73]. In addition to measure video quality, we also characterized the network performance by goodput. Since multicast is enabled, multiple video versions may be received. For each client, every interested video with exactly one resolution level (minimum of the highest received and the requested levels) will contribute to the goodput.

Video Coding: Our video server encodes videos using the standardized FFm-

peg tool (version 2.4.3) with H.264 codec. In the experiment the full HD video sequences are encoded at 10Mbps with 25fps (shown in Table 3.8, where video sequence *controlled-burn* is compressed with qp=25). We generate five resolution levels for each video sequence and the numbers of pixels at different resolution levels are: 1920×1080, 1600×900, 1280×720, 960×540, and 640×360.

4.3.1 Testbed-Based Evaluation

We evaluate the performance of JurCast and compare it to Best-RSSI and DirCast$^+$ schemes using up to 8 mobile devices. In particular, the benefits of *balancing workload* and *exploiting wireless multicast* are evaluated in the following two subsections, respectively. The wireless devices communicate with the AP operating in 802.11a mode.

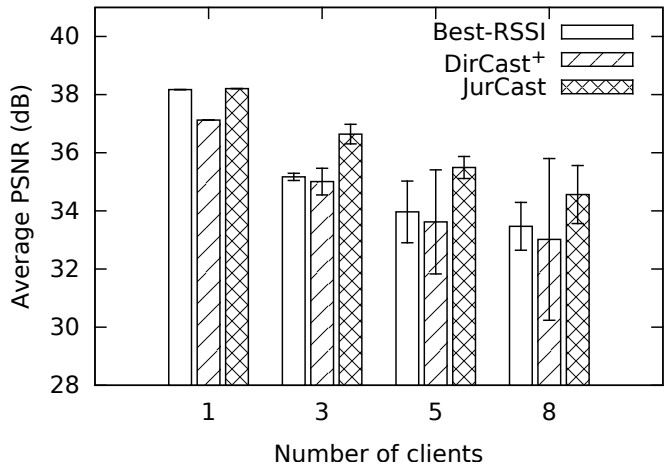


Figure 4.3: Average PSNR per video session.

Baseline Comparison

As only two APs are deployed in the testbed, we initially cluster all mobile devices to one AP by deploying them relatively close to one particular AP, while each client can also communicate with another AP at a lower transmission rate. On the other hand, to create distinct channel conditions between the clients and APs, these smartphones are placed in different locations.

In these experiments, we have six video sessions in total and each client randomly

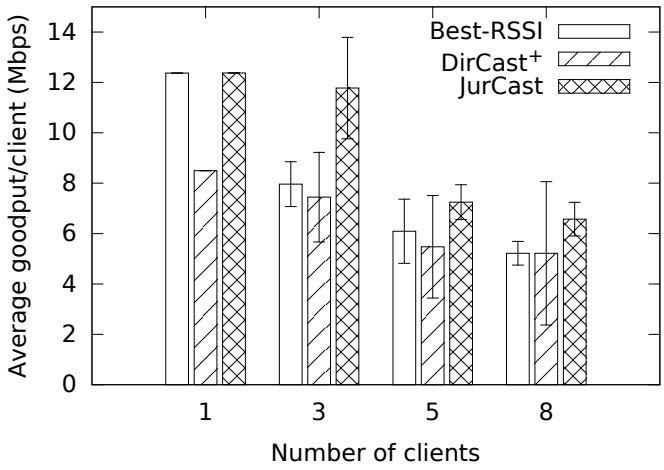


Figure 4.4: Average Goodput per client.

subscribes two video sessions. The request resolution levels of the interested video sessions are between 3 to 5.

The average PSNR values are depicted in Figure 4.3. For each multicast scheme, we aggregate the PSNR values of all videos for each client and present the average of them from 10 runs. The result from the figure shows that JurCast significantly outperforms other two schemes when multiple clients are present in the system. On the average, JurCast improves the video quality by about 2dB over Best-RSSI and DirCast$^+$. As expected, the video quality reduces for all three schemes as the number of mobile devices increase. From the figure, we reveal that the highest video quality improvement between our approach and other methods is achieved when there are 3 or 5 users. As more workload is introduced by more devices, the enhancement is slightly reduced.

The figure also shows that the DirCast$^+$ performs slightly worse than Best-RSSI scheme, especially when only a single client is present in the system. Best-RSSI employs the optimal resource allocation algorithm that presented in previous chapter to intelligently determine the resolution version transmitted for each video session. By contrast, DirCast$^+$ takes the uniform resolution version to transmit for a set of videos interested by one client, where the transmissions may not be fully utilized. Moreover, DirCast$^+$ cannot benefit from the association control with one or few clients.

During the same experiments, we also measure the goodput for each client and present the results in Figure 4.4 with respect to the different number of clients. A similar goodput pattern is present as that of PSNR. Since goodput is also closely related to the video packet receptions, the more packets a client receives, the higher goodput and PSNR values are observed. In particular, the goodput improvements over other two schemes with 3 and 5 clients are about 40% and 30%, respectively.

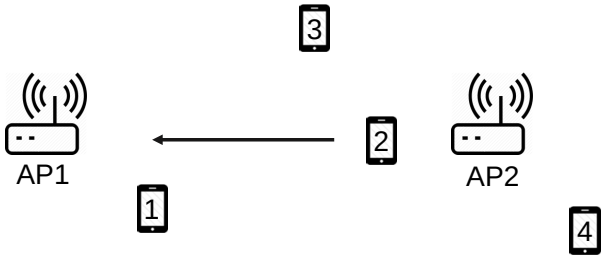


Figure 4.5: Mobility experiment: the testbed consists of two APs and four clients. Client 2 is the mobile client, the moving direction is represented by the arrow. Clients 1 and 3 subscribe the same set of videos; and clients 2 and 4 subscribe another set of videos.

Client Mobility

We have evaluated the performance of our proposed scheme under static conditions where all clients remain stationary. In this subsection, we measure the performance with client mobility. The experimental testbed consists of two APs and four clients, with three stationary and the other moving at walking speed. The illustration diagram is present in Figure 4.5. The frame PSNR value is plotted in Figure 4.6, where the moving period is between frame 100 to 400. During the moving period, the network conditions (RSSI and historical packet loss rate) are continuously updated to the gateway to intelligently determine the association scheme.

As depicted in the diagram, clients 3 and 4 are closer to AP_2 and client 1 is closer to AP_1, and these three clients are static. Client 2 is moving from AP_2 towards AP_1. Since clients 1, 2, and 3 are placed between two APs, they all observe fair wireless condition from the farther AP. According to this network condition, Best-RSSI scheme will group clients 2, 3, and 4 to AP_2 and associate client 1 to AP_1, in the initial stage.

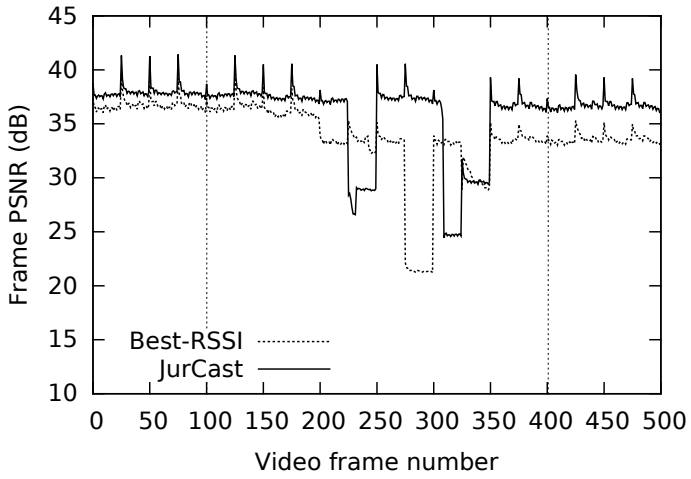
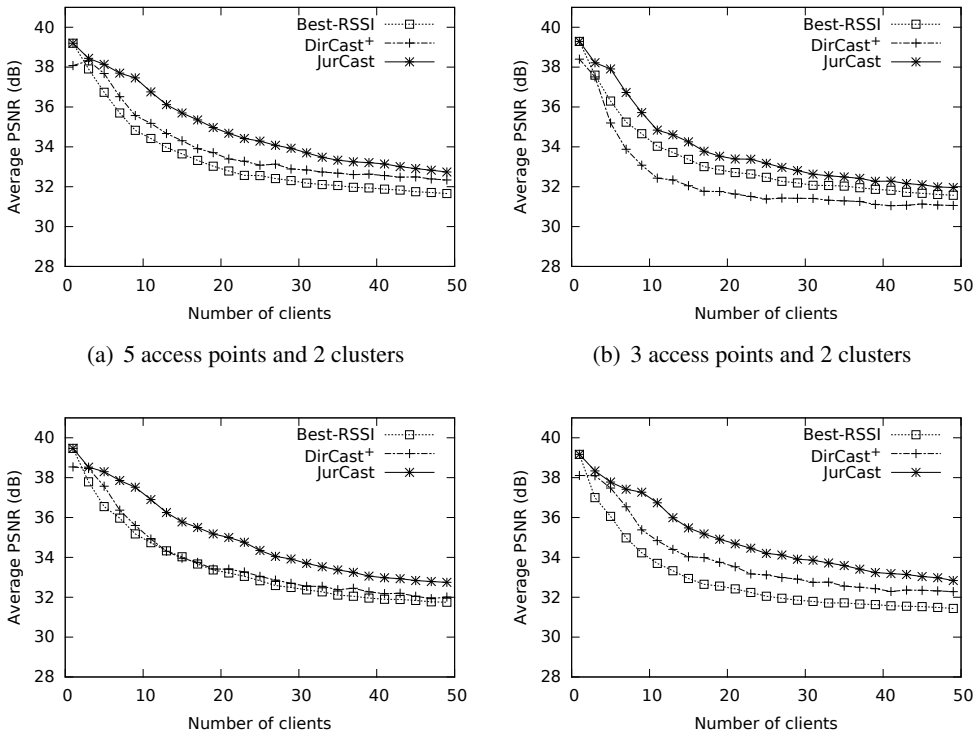


Figure 4.6: Frame psnr value of the mobile client (client 2). The moving period is from frame number around 100 to 400.

Since the network condition between client 3 and AP_1 is fair, client 3 is associated to AP_1 to exploit wireless multicast in JurCast, which reduces the load of AP_2. As a result, the frame PSNR in JurCast is slightly higher than that of Best-RSSI before the movement.

During the mobility period (frame number 100 to 400), client 2 is moving towards AP_1, meanwhile, this client has been re-associated to AP_1 in Best-RSSI. In JurCast, client 2 is still associated with AP_2 as long as the supported link rate is fair, although it becomes lower as moving farther from AP_2. Figure 4.6 shows that after moving period (frame number 400), both JurCast and Best-RSSI suffer from video quality degradation. From the results, we can reveal the following two findings:

First, the frame PSNR of JurCast is slightly reduced as the supported multicast link rate is lower than the initial status. Second, client 2 in Best-RSSI experiences remarkably quality degradation because of the re-association. Although client 2 re-associates to the AP (AP_1) with lighter load and attains a higher link rate, which reduces the multicast opportunity. At the initial stage, clients 2, 3, and 4 are associated to AP_2, where clients 2 and 4 shares the identical interests. Benefiting from multicast and intelligent resource allocation, clients 2 and 4 will receive higher video resolution quality to maxi-



(a) 5 access points and 2 clusters

(b) 3 access points and 2 clusters

(c) 5 access points and 3 clusters

(d) 5 access points and 1 clusters

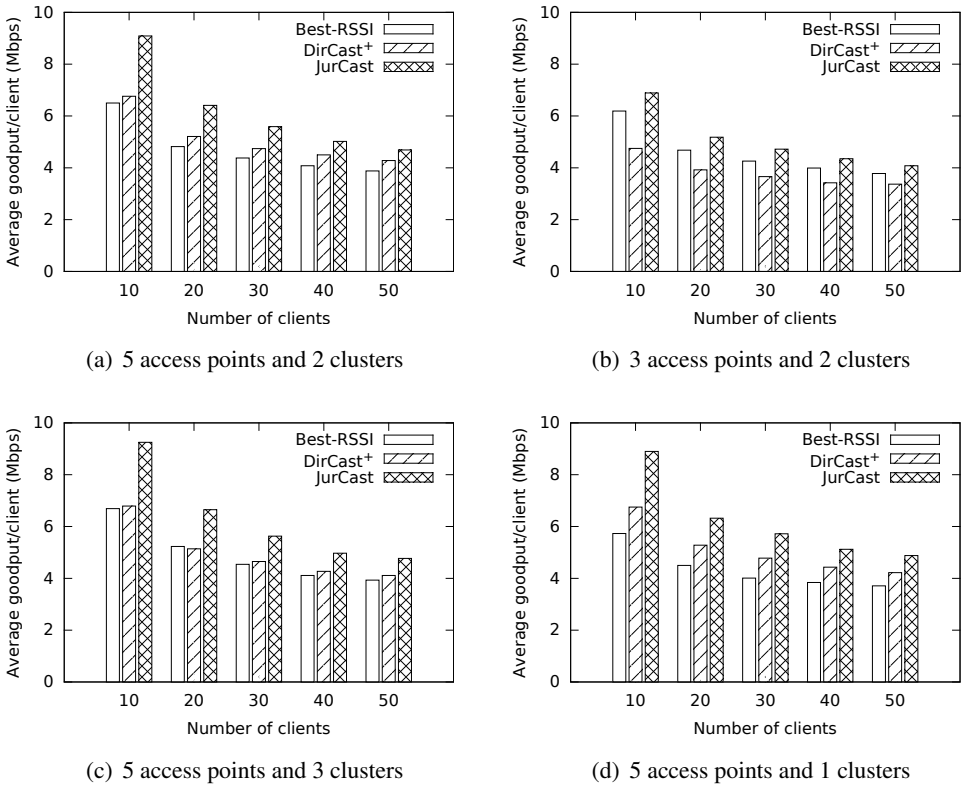
Figure 4.7: Average PSNR per video with different configurations.

mize the overall utility. After the re-association, without multicast, client 2 suffers from considerable quality degradation as the load of AP_1 is substantially increased.

4.3.2 Simulation-Based Evaluation

In this section, we extend our evaluation using simulations to determine the scalability of JurCast to larger deployments. Specifically, (i) we investigate the impact of workload by varying the number of clients and number of APs deployed in the system; (ii) we evaluate how the three methods perform with the different degrees of clustering (initial status); (iii) we present and discuss the algorithm computational overhead.

We implement these algorithms on ns-3 simulator (version 3.22) in C++. The neighboring APs operate on non-overlapping 802.11a channels. To create the distinct link conditions, we randomly generate the received signal strength expected to be received at each client from different APs. Furthermore, we have 12 distinct video ses-



(a) 5 access points and 2 clusters

(b) 3 access points and 2 clusters

(c) 5 access points and 3 clusters

(d) 5 access points and 1 clusters

Figure 4.8: Average goodput per client with different configurations.

sions, and each client randomly subscribes 3 videos.

The average PSNR and goodput values are presented in Figures 4.7 and 4.8, respectively. The results indicate that JurCast considerably outperforms the other two schemes under all configurations we evaluated. In particular, JurCast achieves up to 2.77dB and 3.04dB PSNR improvement, compared to DirCast$^+$ and Best-RSSI, respectively. Moreover, the corresponding goodput improvements are up to 45% and 55%.

Impact of Workload

In our measurement, we create different levels of workload by varying the number of clients (Figure 4.7(a)) and the number of APs (Figure 4.7(b)). It is clear that introducing more clients or deploying less number of APs leads to the heavier load. In particular, We examine the algorithm performance with up to 50 clients under the configuration

with 5 or 3 APs.

Figure 4.7 clearly shows that all three schemes suffer from video quality degradation as more clients are deployed. JurCast achieves the highest improvement over other schemes when there are about 10 clients. When more than one clients are present, AP association control will be exploited by DirCast$^+$, especially for more APs. As a result, DirCast$^+$ achieves higher performance over Best-RSSI (shown in Figure 4.7(a)). Comparing Figures 4.7(a) and 4.7(b) confirms that deploying less number of APs also declines the PSNR value with the same number of clients.

Degree of Clustering

We evaluate the impact of the clustering degree at the initial stage, which relates to the decision of association control and therefore determines the transmission link rate. The degrees of clustering are distinct in all sub-figures of Figures 4.7. We make the following observations regarding these results.

First, the performance impacts of varying clustering degree on JurCast and DirCast$^+$ are almost unnoticeable when different number of clusters are configured (Figures 4.7(a), 4.7(c), and 4.7(d)). The observed consistent trends are mainly attributed to the employed association control mechanisms. With association control employed, to balance workload, most of clients are generally not associated with the AP that has the highest RSSI value.

Second, unlike other two schemes, the AP with the highest RSSI is associated to each client in Best-RSSI scheme. As a result, it is most sensitive to the changes of clustering degree. In particular, with the same number of APs deployed, the configurations with more clusters achieve a better video quality and goodput.

Algorithm Running Time

Apart from the measured performance, algorithm computational overhead is another paramount factor that determines the efficacy of the proposed solutions. In the previous experiments, the algorithm running time under the scenarios with 5 APs is recorded as well. Figure 4.9 plots the average running time values with respect to different number of clients.

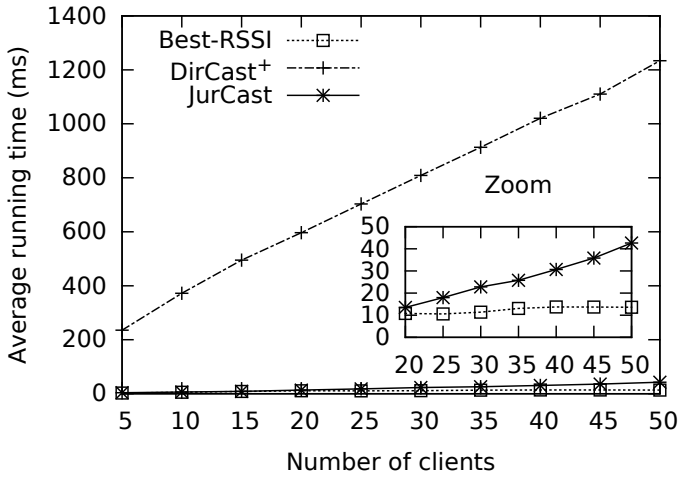


Figure 4.9: Average algorithm running time. The number of available video sessions is 12 ($N_v = 12$).

The results show that the algorithm running time of JurCast and DirCast$^+$ is linearly increasing with the number of clients (denoted by n). The number of distinct transmission states in both JurCast and DirCast grow linearly with n. The coefficients of the linear functions for JurCast and DirCast with respect to the number of video sessions are $\mathcal{O}(N_v)$ and $\mathcal{O}(2^{N_v})$, respectively. As the quantity of the transmission states in DirCast is exponentially increasing with the number of video sessions, the running time of DirCast is substantially greater than JurCast, which is impractical to be employed in real-time adaptive system, especially with large N_v.

Best-RSSI scheme employs the optimal allocation algorithm proposed in previous chapter, thus the computation complexity of the algorithm is independent of the number of clients. As a consequence, with more clients introduced in the system, the running time does not vary much.

4.4 Summary

This chapter presents JurCast, a joint user and rate allocation scheme for video multicast over multiple APs. JurCast effectively allocates resource for multicast video streaming by balancing the trade-off between high transmission rate, load balancing, and multicast opportunities. Our extensive measurements in both testbed and large scale simulation

demonstrate that JurCast significantly outperforms other compared adaptive schemes with up to 3dB and 55% enhancements in terms of PSNR and goodput, respectively.

CHAPTER 5

CONTENTION WINDOW ADAPTATION UNDER ASYMMETRIC CONDITIONS

In Chapter 4, we extended the network to multiple access points, where we assumed that neighboring APs are operating on non-overlapping channels. In many cases, the number of non-overlapping channels, however, is insufficient, especially for densely deployed APs. As a result, two or more APs may transmit on the same channel; and the fully-connected network will be separated to multiple collision domains.

The performance of the widely used CSMA/CA protocol, however, is severely inadequate in scenarios with multiple collision domains. In such scenarios, the existence of hidden terminal topologies with *information asymmetry* (or heterogeneity) [28] and *channel asymmetry* [19] is known [14] to severely hamper network performance. To compound the problem, physical capture effect [40] and rate asymmetry (or anomaly) [34] could also lead to extremely low system utilization and serious fairness issue.

These performance issues have prompted researchers to devote significant effort to develop models to analyze the behavior of CSMA/CA MAC protocol, since the pioneering work of Kleinrock and Tobagi [39]. In spite of these efforts, the state of art in estimating the impact of multi-collision domain heterogeneous topologies is still rather limited. Much of the existing work [14, 35, 28, 70, 19, 83, 54] fails to provide simple and effective solutions that can address all the above mentioned performance issues in a holistic manner.

In this chapter, we develop a simple and accurate analytical model that is able to handle channel asymmetry, information asymmetry, rate asymmetry, and capture effect for well known heterogeneous topologies. To the best of our knowledge, it is the first theoretical model that can be utilized to resolve the heterogeneous topologies and rate anomaly problems at the same time. Our model is based on the average value approximation (AVA) method [67] and utilizes the contention windows (CW) used by the competing flows as the key parameters. The key novelty is the development of a model to approximate the ratio of successful transmissions between two competing flows.

To summarize, this chapter makes the following contributions:

- We present simple closed-form equations that can be used to compute the appropriate contention windows that meet a given performance objective, such as fairness and throughput ratio for heterogeneous topologies, including those without known solution in the literature [28, 55, 76].

- Our model is flexible and can support different notions of fairness, including max-min fairness, time fairness, and proportional fairness.

- Our solution can be applied to general two-flow asymmetric topologies, as well as many multiple-flow asymmetric topologies.

The remainder of this chapter is structured as follows. Section 5.1 presents the background of IEEE 802.11 distributed coordination function. In Section 5.2, we review the well-known problematic asymmetries and state our assumptions in our work. We

present the analytical model for basic two-flow asymmetric topologies in Section 5.3. We illustrate how our model can be applied to tune the throughput of contending flows in the identified two-flow asymmetric topologies in Section 5.4. We further extend our analysis and show how our model can be applied to a broad class of three flows asymmetric topologies and multi-flow chain topology in Section 5.5. Section 5.6 validates our model and evaluates its usefulness by means of simulations and experiments. Section 5.7 concludes the work.

5.1 Background of IEEE 802.11 Distributed Coordination Function

The distributed coordination function (DCF) is a protocol based on CSMA/CA. The binary exponential backoff (BEB) scheme is adopted as the collision avoidance scheme of DCF [30, 33]. In particular, *basic access mechanism* and *RTS/CTS access mechanism* are specified as the two DCF access standards.

Figure 5.1 depicts the basic access mechanism, which operates as follows. Before transmission, a station monitors the channel activity to check whether the medium is available by using virtual career-sensing and physical carrier-sensing. The medium is considered busy if either carrier-sensing indicates so. Virtual carrier-sensing considers the medium to be idle if the Network Allocation Vector (NAV) is zero, otherwise it considers the medium to be busy. Only when NAV is zero, physical carrier-sensing is performed. NAV is set by the received DATA or ACK packet, which carries the information of the length of the packet to be transmitted. If the channel is idle for a period of time equal to a distributed inter-frame spacing time (DIFS), the station transmits. Otherwise, the station persists to monitor the channel until it is measured idle for a DIFS. At this point, the station then waits for a random backoff interval uniformly chosen between $(0, W)$, where W is called *contention window*. If at any time during the period above the medium is sensed busy, the station freezes its counter and the countdown resumes only after the medium becomes idle again for a DIFS. When the counter

decrements to zero, the node transmits the packet immediately.

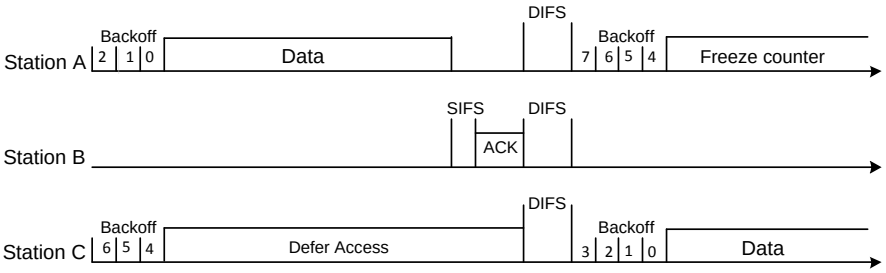


Figure 5.1: Basic Access Mechanism: stations A and C are two contenders, both of them are transmitting to the station B.

In order to reduce the collisions due to the hidden terminal problem (see, for instance [69]), a four-way handshake protocol is deployed in the RTS/CTS access mechanism. The sensing for a DIFS idle duration and backoff mechanisms are identical to the basic access mechanism. The slight difference can be stated as that a request to send packet (RTS) rather than a data packet is transmitted when the backoff time expires. Upon successful transmission of the RTS frame, the destination sends a clear to send (CTS) packet back to the sender after waiting for a SIFS interval. The source will wait for another SIFS period after the reception of the CTS frame, then transmits the DATA immediately. The remaining transmission is performed as in basic access mechanism. Here, a station that hears either RTS, CTS, DATA, or ACK frame updating its NAV value based on the duration information in the corresponding frame. Since the RTS and CTS frame sizes are pretty small, such a four-way handshake scheme significantly reduces collision probability.

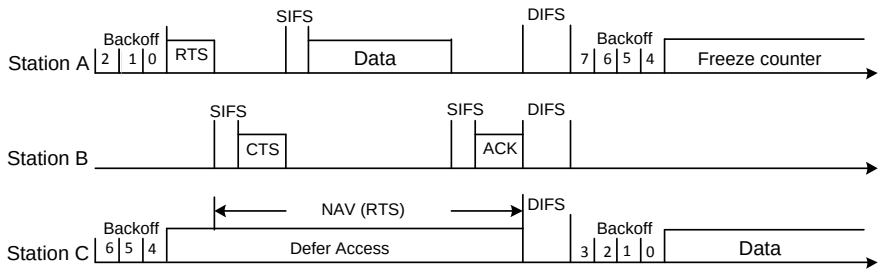


Figure 5.2: RTS/CTS Access Mechanism: stations A and C are two contenders, both of them are transmitting to the station B.

The BEB procedure is implemented by means of the backoff counter and backoff stages. Initially, upon receiving a new frame to be transmitted, the station starts at the backoff stage 0, and the corresponding contention window size W is set as W_{min}. At each packet transmission attempt, the backoff time is uniformly chosen in the range $(0, W)$. Following each unsuccessful transmission, the backoff stage is incremented by 1. As a result, the current contention window size W is doubled, up to a maximum value $W_{max} = 2^m \times (W_{min} + 1) - 1$, after which the backoff stage unchanged on subsequent collisions. The backoff stage is set back to the initial value 0 after each successful transmission or after reaching the maximum retry limit for the data packet.

5.2 Motivation and Assumptions

In this section, we motivate our work and state our assumptions using examples of well known heterogeneous topologies with multiple collision domain. We assume that RTS/CTS is used to mitigate the hidden terminal problem [14].

5.2.1 Heterogeneous Networks

We now briefly describe the three fundamental asymmetric conditions that cause severe unfairness in CSMA/CA.

Information Asymmetry (IA). The two topologies depicted in Figure 5.3 consist of two competing flows: a *dominant* flow (from C to D) and a *weak* flow (from A to B). The key property of this information asymmetry topology is that, while the receiver of the weak flow (B) can hear the sender of the dominant flow (C), the receiver of the dominant flow (D) cannot hear the sender of the weak flow (A). As a result, under high traffic load, the sender C can always successfully compete for channel access, while B is unable to respond with a CTS due to the suppression by C. Therefore, the throughput of $flow_{AB}$ is seriously hampered by such information asymmetric scenarios; results [28, 55] have shown an extremely low throughput ratio of $flow_{AB}$ to $flow_{CD}$. There is currently no effective solution to ensure fairness for the topologies shown in Figure 5.3.

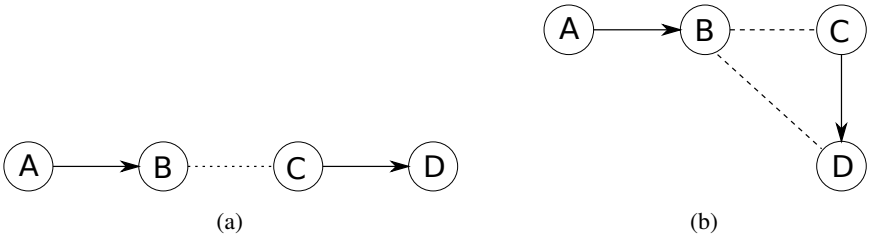


Figure 5.3: Information asymmetric topologies. The arrows represent the directions of the data flows, and the dotted lines indicate overhearing links.

Channel Asymmetry (CA). The other well known asymmetry is *channel asymmetry* [19], shown in Figure 5.4, where two competing senders compete for the same receiver. The problem arises when there is a significant difference in channel qualities between the two flows. This problem is exacerbated if there are physical layer capture effects [40], causing transmission on the better channel to always dominate over the weaker channel. The use of RTS/CTS access mechanism will not mitigate this problem. In such a scenario, under heavy traffic, node A would starve.

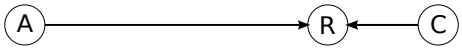


Figure 5.4: The shorter and longer arrows represent the dominant and weak flows, respectively.

Rate Asymmetry (RA). A WiFi network may have different client stations with different channel access rate (depending on the channel conditions) or different interface standards (e.g. 802.11a, 802.11b, 802.11g or 802.11n). When such diverse sets of channel rates are used by the clients associated with a single AP, the problem of *rate asymmetry* (rate anomaly) can occur [34]. This problem is because the CSMA/CA protocol is designed to provide fair transmission opportunity to all clients regardless of the corresponding link rate. Hence, the use of lower link rates can significantly reduce the overall AP throughput. The work in [35] provides an elegant solution for rate asymmetric problem in fully-connected networks. We suggest a complementary solution that jointly considers all three asymmetric issues mentioned.

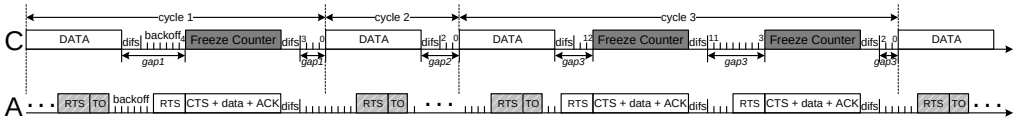


Figure 5.5: The behavior of CSMA under the topology in Figure 5.3(b). The DATA frame consists of a data packet and all corresponding overhead, which includes RTS, CTS, DIFS, etc. The block TO represents the TimeOut period.

5.2.2 Assumptions

We assume the following in our analysis. First, we assume the use of RTS/CTS. RTS/CTS is introduced to resolve the hidden terminal problem [14]. Since we are considering the HD video streaming applications, the network flows could be easily saturated. Under saturated conditions, without RTS/CTS, there would be significant amount of collision, resulting in extremely low throughput [28, 55].

Second, we assume that CW can take any integer values, not necessary in the form of $2^k - 1$ for $4 \leq k \leq 10$ as dictated by the standards. Such fine tuning of CW in turns allows fine tuning of throughput among the competing flows, yielding better fairness and higher throughput. Our model, however, can still work in settings where CW is limited to the form of $2^k - 1$ (by rounding the CW computed by our model to the nearest $2^k - 1$ for some k).

Third, we assume that while CW is tunable, no Binary Exponential Backoff (BEB) is performed. When an AP (sender) detects that channel is busy, it picks a backoff interval that is uniformly distributed between 0 and W. In our analysis, the parameter W remains constant without performing BEB. We do not assume the use of BEB for the following reasons: (i) Our analysis allows us to select the "correct" contention window size given a specific flow topology. BEB is designed to handle general topologies and does not account for the number of flows and how these flows interact. When our model is applicable, the use of BEB becomes less relevant. (ii) Measurements by us and other researchers [32, 16] have shown that BEB is not implemented uniformly on popularly WiFi chipsets, including chips from Intel, Atheros, and Broadcom. For example, no BEB is performed on some Atheros chips. Use of BEB only on some nodes will in fact

exacerbate the unfairness problem. (iii) Existing work [38, 54, 55, 83] implies that use of RTS/CTS, CW tuning, and disabling of BEB are prerequisites for mechanisms that can effectively mitigate unfairness problems in heterogeneous topologies.

Hence, we argue that not modeling BEB is not a significant drawback. Our analysis can be extended to include BEB, though its inclusion would make the model more complicated and will be left for future work.

5.3 Analytical Model

We now present our model that characterizes the interaction between two competing, saturated, flows in an heterogeneous topology as a function of CW.

5.3.1 Interaction Model

A key to the simplicity of our model is the abstraction we used to model the interaction between the weak flow and dominant flow in the asymmetric topology.

We illustrate the interaction model using a basic interaction between two contending flows in the topology depicted in Figure 5.3(b). We define a *cycle* as a successful data transmission from C, followed by a backoff. The backoff period starts after the data transmission of C is completed and the backfoff interval is randomly selected between 0 and W_C. During this backoff period, there can be zero or more transmissions by node A. Note that if sender A is able to successfully transmit during C's backoff period, node C freezes its backoff timer and resumes the timer once A's transmission completes. We call the interval between two transmissions from C, excluding any data transmission from A, a *gap*, and denote $\mathcal{L}_C$ as the backoff time plus the time duration of a DIFS (T_{DIFS}). For A to transmit successfully, A must insert the entire RTS plus a SIFS into a gap of C. An illustration of the detailed MAC behavior of CSMA with asymmetric topology described is shown in Figure 5.5.

We highlight an important observation here. Since the receiver D is free from collision, C can always transmit RTS successfully when it countdowns to zero. As a result, C performs only one backoff between two successive transmissions. Note that

A could insert several data frames into a single backoff period of C (see gap 3 in Figure 5.5), which is omitted in Garetto's model [28]. Such multiple RTS insertions into a single backoff interval is non-negligible, especially when the contention window of C is significantly larger than A.

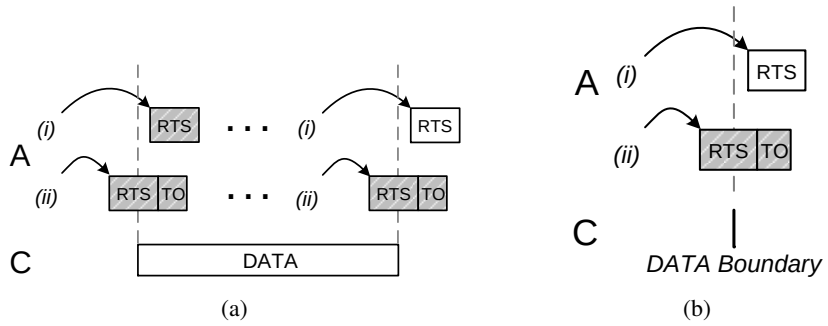


Figure 5.6: The scenarios of the round overlaps the boundaries of DATA.

To model the interaction between the two competing flows, we first simplify this interaction by abstracting away the unnecessary details to focus on the insertion of A's RTS (plus SIFS) into C's gap. Let $\Re_{AC}$ be the expected number of successful RTS/CTS transmission of A within one gap of C. This value is *independent of the data length of* A because when A is able to complete RTS/CTS and transmits data, C should have received the CTS and freezes its backoff timer accordingly. Therefore, we can abstract away the data transmission of A in our analysis, and simply regard a *round* of A's activity as consisting of a random backoff period, followed by an RTS transmission. The length of a round is T_{RTS} plus a period uniformly distributed in the range of $(0, W_A)$, where W_A is the CW size of A.

The rounds of A that are completely contained within C's data can be ignored, since B cannot respond to A's RTS request and there can be no transmission by A. Therefore, our model focuses on A's rounds that overlap with C's data. Two cases are shown in Figure 5.6(a), either A's RTS overlaps with C's data boundary, or it does not. Considering that A's RTS does not overlap with the ending boundary of C's data, an extra possibility that the timeout period (equals the length of a CTS) following an RTS overlaps with C's data ending boundary may exist, which delays the next backoff

for T_{CTS}. The occurring probability of this case can be approximately calculated as $T_{CTS}/(T_{CTS}+T_{RTS}+W_A/2)$, and T_{CTS} times this probability is the expected delay of the next backoff, which is about one slot (when $W_A = 15$). As the introduced delay is small, we will ignore this scenario in the rest of the discussion to simplify the model.

Next, if we assume that data length is large, then the probabilities that an RTS overlaps with the starting boundary of C's data and ending boundary of C's data are independent and, by average value modeling [67, 68], the same. Hence, we can further abstract away the data length of C in modeling the interaction between A and C, as shown in Figure 5.6(b).

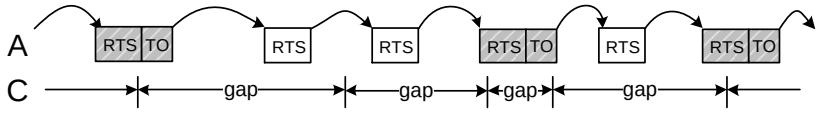


Figure 5.7: Abstracted interaction between two stations. The unsuccessful RTS requests are marked by grey background.

The final interaction model is illustrated in Figure 5.7, where only RTS remains and CTS, data, ACK, SIFS, and DIFS periods are abstractly away. This simplified abstraction is the key to derive simple closed-form equation for $\Re_{AC}$.

5.3.2 Analytical Model

With the interaction abstraction above, we can now derive an equation that estimates the average number of successful RTS transmission by A, using average value modeling [67].

We first compute the expected number of RTS requests from A within a backoff interval of C (between two transmission from C), E_{RTS}, as follows. The average gap length, $\bar{\mathcal{L}}_C$, is $W_C/2+ T_{\text{DIFS}}$. Let the probability that an RTS frame crosses a gap boundary be P_R, and $\bar{\mathcal{L}}_A$ be the expected length of a round of A. We have

$$\bar{\mathcal{L}}_A = W_A/2 + T_{\text{RTS}} + P_R T_{\text{CTS}}. \tag{5.1}$$

Since A sends one RTS frame in each round, the expected number of RTS requests from

A is given by

$$E_{RTS} = \frac{\bar{\mathcal{L}}_C}{\bar{\mathcal{L}}_A} = \frac{W_C/2 + T_{\text{DIFS}}}{W_A/2 + T_{\text{RTS}} + P_R \cdot T_{\text{CTS}}}. \tag{5.2}$$

As shown in Figure 5.7, every starting point of A's RTS is covered by a gap. As the gaps of C are completely independent of the RTS from A, considering the length of the gap that covers a particular starting point of RTS yields,

$$Pr(\mathcal{L}_C = T_{\text{DIFS}} + i) = \frac{T_{\text{DIFS}} + i}{\sum_{j=0}^{W_C} \{T_{\text{DIFS}} + j\}}, \tag{5.3}$$

for $0 \leq i \leq W_C$. As each RTS starting point randomly samples a point in the gap with length $\mathcal{L}_C$, the probability of this RTS frame crosses the boundary of the gap is $\min\{T_{\text{RTS}}/\mathcal{L}_C, 1\}$. Finally, the average probability P_R is expressed as

$$\begin{aligned} P_R &= \sum_{i=0}^{W_C} \left\{ Pr(\mathcal{L}_C = T_{\text{DIFS}} + i) \cdot \min\{\frac{T_{\text{RTS}}}{\mathcal{L}_C}, 1\} \right\} \\ &= \frac{2W_C T_{\text{RTS}} + T_{\text{RTS}} + T_{\text{DIFS}} - (T_{\text{RTS}} - T_{\text{DIFS}})^2}{(W_C + 1)(W_C + 2T_{\text{DIFS}})}. \end{aligned} \tag{5.4}$$

We can now estimate $\Re_{AC}$, the expected number of successful RTS requests by A within a gap of C. By using Equation (5.2), $\Re_{AC}$ can be written as

$$\Re_{AC} = E_{\text{RTS}}(1 - P_R) = \frac{(W_C/2 + T_{\text{DIFS}})(1 - P_R)}{W_A/2 + T_{\text{RTS}} + P_R T_{\text{CTS}}}. \tag{5.5}$$

Substituting Equation (5.4) into (5.5) yields a closed-form equation of $\Re_{AC}$. Equation (5.5) implies that the accuracy of $\Re_{AC}$ is closely related to the absolute error of P_R.

As it is difficult to observe any intuition from Equation (5.5), we make the following approximation

$$\frac{(T_{\text{RTS}} - T_{\text{DIFS}})^2 + T_{\text{RTS}} - T_{\text{DIFS}}}{(W_C + 1)(W_C + 2T_{\text{DIFS}})} \approx \frac{1}{W_C + 2T_{\text{DIFS}}}. \tag{5.6}$$

This approximation is made based on the following reasoning. When W_C is large, say $W_C = 1023$, the absolute values of the expressions for both the left side and right side are negligible, less than 0.001 after substituting the values of T_{RTS} and T_{DIFS} given in Table 1. When W_C is small, the error is larger. For the smallest value of $W_C = 15$, the largest absolute error of this approximation is 0.01 based on the default values. The approximation (5.6) is thus able to highly simplify the expressions of P_R and $\Re_{AC}$, the relative inaccuracy introduced in the worst case is only 3.4%. The simplified equations are as follows,

$$\hat{P}_R = \frac{2T_{\text{RTS}} - 1}{W_C + 2T_{\text{DIFS}}} \tag{5.7}$$

$$\hat{\Re}_{AC} = \frac{W_C/2 + T_{\text{DIFS}} - T_{\text{RTS}} + 1/2}{W_A/2 + T_{\text{RTS}} + \frac{T_{\text{RTS}}-1/2}{W_C/2+T_{\text{DIFS}}} \cdot T_{\text{CTS}}}. \tag{5.8}$$

Equation (5.8) shows that $\hat{\Re}_{AC}$ can be approximated as $\frac{W_C}{W_A}$ when $W_C, W_A \gg T_{RTS}, T_{DIFS}, T_{CTS}$. This condition can be satisfied either by enlarging the window sizes or by decreasing the values of T_{RTS}, T_{DIFS}, and T_{CTS}. Interestingly, Magistretti et al. [47] suggest that one way to mitigate the unfairness in presence of hidden terminals including asymmetric topologies is to shorten the RTS and CTS lengths. Our model provides an analytical justification for their approach.

5.3.3 Throughput Model

The previous model yields $\hat{\Re}_{AC}$, the expected number of successful RTS transmissions A can insert into C's gap. In this section, we develop a pair of equations that estimate the throughput of A and C using renewal theory. The renewal instance in the model is defined as the duration between the start of two RTS transmissions of C, and the average length of this renewal instance is denoted as $\bar{T}_{cycle}$.

As shown in Figure 5.5, the system state is at either *successful transmission of A or C* or *idle state of C*. Let l_i and r_i denote the payload length (bytes) and data rate of $flow_{ij}$, respectively. Without considering the headers and overhead of physical layer, the payload transmission time is l_i/r_i. Based on these notations, we deduce the

expected duration of each state within a renewal cycle in the following paragraphs.

Let the total time to transmit a data frame of node i, including overhead, as F_i. With RTS/CTS enabled, we have

$$F_i = T_{\text{RTS}} + T_{\text{CTS}} + T_{\text{SIFS}} + \frac{l_i}{r_i} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{DIFS}},$$

where T_{ACK} is the length of ACK frame. As shown in Figure 5.5, the idle state is completely attributed to the backoff of C. More precisely, an average backoff period of C excluding the expected duration of successful RTS transmissions by A represents the average idle period, which is equal to $W_C/2 + T_{\text{DIFS}} - \hat{\Re}_{AC} T_{\text{RTS}}$. As a consequence, we have

$$\bar{T}_{cycle} = \hat{\Re}_{AC} F_A + F_C + W_C/2 + T_{\text{DIFS}} - \hat{\Re}_{AC} T_{\text{RTS}}. \tag{5.9}$$

Without loss of generality, we assume that nodes have an equal payload size of l. Let S_i be the throughput of node i. By applying the renewal theory, we have

$$\begin{aligned} S_C &= \frac{l}{\hat{\Re}_{AC} F_A + F_C + W_C/2 + T_{\text{DIFS}} - \hat{\Re}_{AC} T_{\text{RTS}}} \\ S_A &= \hat{\Re}_{AC} S_C \\ &= \frac{\hat{\Re}_{AC} l}{\hat{\Re}_{AC} F_A + F_C + W_C/2 + T_{\text{DIFS}} - \hat{\Re}_{AC} T_{\text{RTS}}}. \end{aligned} \tag{5.10}$$

So far, we have attained a model to characterize the performance of topology shown in Figure 5.3(b). The same model captures the performance of channel asymmetric topology (Figure 5.4) as well. Additionally, the model can also capture the performance of the topology in Figure 5.3(a), with a slight change in average gap length, $\bar{\mathcal{L}}_C$.

5.4 Contention Window Tuning

We now present an application of our analytical model in preventing starvation and improving fairness among two competing flows in asymmetric topologies. Our model allows fine control of the throughput of both flows, by tuning the contention window size. Unlike existing mechanisms to mitigate unfairness and starvation, which is mostly heuristic in nature, our model allows any contention window tuning mechanism to qualify the effect of adaptation on the throughput of the both flows. Further, our method is flexible enough to support different notions of fairness. We illustrate this flexibility in the following three subsections.

5.4.1 Max-Min Fairness (M-MF)

To achieve max-min fair among two competing flows, we maximize the minimum throughput of nodes A and C. Such max-min fairness can be achieved by setting W_A to the minimum possible contention window size (15 in 802.11a) and $\hat{\Re}_{AC}$ to 1, as stated in the following lemma.

Lemma 1. *The max-min fair is achieved when W_A is minimum and $\hat{\Re}_{AC} = 1$.*

Proof. When $\hat{\Re}_{AC} = 1$, we have $S_A = S_C$. It remains to show that when W_A is minimum, S_A is maximum. Given W_A, we have a unique W_C^* that satisfies the equation $\hat{\Re}_{AC} = 1$. For practical settings of contention window size $W_i \in [15, 1023]$, W_C^* is a monotonically increasing function of W_A. Thus, setting a minimum W_A would lead to minimum W_C^*. Equation (5.10) implies that the throughput of A and C is a monotonically decreasing function of W_C^*. When both W_A and W_C^* are minimum, the throughput is maximized. □

5.4.2 Time Fairness (TF)

With *rate asymmetry*, the host with lower data rate may substantially decrease the channel occupation time of the faster host. In this case, applying max-min fairness could result in extremely low resource utilization. One option is to apply *time-fairness* where

each host is given equal channel access duration and the faster host can transmit significantly more data. To this end, we simply set $\hat{\Re}_{AC} = r_A/r_C$. Combining with the equation for $\hat{\Re}_{AC}$ (Equation (5.8)) gives

$$\frac{W_C/2 + T_{\text{DIFS}} - T_{\text{RTS}} + 1/2}{W_A/2 + T_{\text{RTS}} + \frac{T_{\text{RTS}}-1/2}{W_C/2+T_{\text{DIFS}}} \cdot T_{\text{CTS}}} = \frac{r_A}{r_C}, \tag{5.11}$$

where the value of W_C is uniquely determined given W_A.

5.4.3 Proportional Fairness (PF)

W_A and W_C can be tuned to achieve proportional fairness. Let s be the function defined by $s(W_A, W_C) = S_A \times S_C$. Then the proportional fairness is formulated as finding W_A and W_C such that $s(W_A, W_C)$ is maximized.

For simplicity, we assume that $r_A = r_C$ and $F_A = F_C$ (Our analysis could be applied to rate asymmetry as well). In this case, the CW tuning mechanism substantially increases W_C to achieve the fairness objectives. As a result, the RTS collision probability (Equation (5.7)) becomes remarkably low. On the other hand, since the timeout period (T_{CTS}) is small, we can approximate $\hat{\Re}_{AC}$ as

$$\hat{\Re}_{AC} = \frac{W_C + 2T_{\text{DIFS}} - 2T_{\text{RTS}} + 1}{W_A + 2T_{\text{RTS}}}. \tag{5.12}$$

With the homogeneous data rate assumption ($r_A = r_C$), the function $s(W_A, W_C)$ is specified as follows,

$$s = \frac{\hat{\Re}_{AC} l^2}{\left((1 + \hat{\Re}_{AC})F_C + W_C/2 + T_{\text{DIFS}} - \hat{\Re}_{AC} T_{\text{RTS}}\right)^2}. \tag{5.13}$$

By incorporating this equation into the fairness objective, we have the following lemma:

Lemma 2. *The proportional fairness is achieved by minimizing* $W_A = 15$ *and setting*

W_C as

$$W_C = \frac{(W_A + 2T_{\text{RTS}})(F_C + T_{\text{RTS}} - 1/2)}{F_C + W_A/2} + 2T_{\text{RTS}} - 2T_{\text{DIFS}} - 1. \tag{5.14}$$

Proof. To achieve the optimal point, we make $\frac{\partial s}{\partial W_C} = 0$. By solving the equation, we obtain Equation (5.14). Let's denote this specific W_C as ${W_C}^*$. Additionally, we have $\frac{\partial s}{\partial W_C} > 0$, when $W_C < {W_C}^*$ and $\frac{\partial s}{\partial W_C} < 0$ when $W_C > {W_C}^*$. Therefore, ${W_C}^*$ is the optimal value for a particular W_A. By substituting Equation (5.14) into Equation (5.13), we get

$$s = \frac{l^2}{(2F_C + W_A)(2F_C + 2T_{\text{RTS}} - 1)}, \tag{5.15}$$

which clearly implies that the optimal s^* is attained when W_A is minimum (i.e., 15 for 802.11a settings). □

5.5 Extensions

Thus far, we have addressed the unfairness problem for asymmetric two-flow topologies. In this section, we will discuss extension of our model and CW tuning mechanism to *asymmetric two-cluster topologies* and *asymmetric three-flow chain topology*.

We concentrate on analyzing these topologies for the following reasons. First, they are prevalent in general large-scale topologies. Due to heterogeneity, these problematic topologies may severely impact the performance of the system. Next, in addition to asymmetric conditions, other well-known topologies are embedded as well, which make the situation worse (e.g. flow-in-middle in Figure 5.8(c)). Last, there are numerous potential interesting applications, in which these topologies could be utilized. The clients associated to an AP in WiFi networks may be highly clustered (e.g. in a room or hall), where Figure 5.8(a) or (b) can be applied.

5.5.1 Asymmetric Two-Cluster Topology

An asymmetric two-cluster topology is a topology where the flows can be grouped into two clusters, such that (i) each cluster consists of one or more flows; (ii) if we abstract each flow cluster as single flow, then the topology reduced to one of the two-flow asymmetric topologies; (iii) every receiver in the first cluster (weak flows) can hear all senders in the second cluster (dominant flows). Similar to two-flow asymmetric topologies, the dominant flow(s) can cause unfairness or starvation for the weak flows in such topologies.

To apply our model, we treat each flow cluster with two or more flows as a single *super flow* and solve for the desired CW for this super flow. To illustrate this, we consider two cases: where the topology within the cluster is (i) symmetric and (ii) asymmetric.

Symmetric Cluster. We illustrate the scenario where a flow cluster is symmetric with a comprehensive set of three-flow topologies, shown in Figure 5.8. Take the topologies in Figure 5.8(a) for instance. We have a super (dominant) flow that consists of $flow_{23}$ and $flow_{45}$. The cases with symmetric cluster are easier to solve, since flows in the cluster are symmetric, the CW sizes of flows within a cluster should be identical to achieve fairness among the flows.

To illustrate how we can apply our model to tune the CW size of all three flows for fairness, we outline the steps for the topology in Figure 5.8(c), which performs worst among these asymmetric topologies. For ease of illustration, we assume that all three flows transmit data frame with the same duration F. Let the contention windows of $flow_{01}$, $flow_{23}$, and $flow_{45}$ be W_0, W, and W, respectively ($W_2 = W_4 = W$). The average gap lengths of Contenders 2 and 4 are identical to $\bar{\mathcal{L}} = T_{\text{ACK}} + T_{\text{DIFS}} + W/2$ (Node 1 cannot hear the ACK from Node 3 or 5). In the following, we calculate the expected overlapped gap length of Contenders 2 and 4.

We regard a *round* of Contender 2's or 4's activity as consisting of a constant average gap period ($\bar{\mathcal{L}}$), followed by a data transmission (F). Contender 2 is clearly

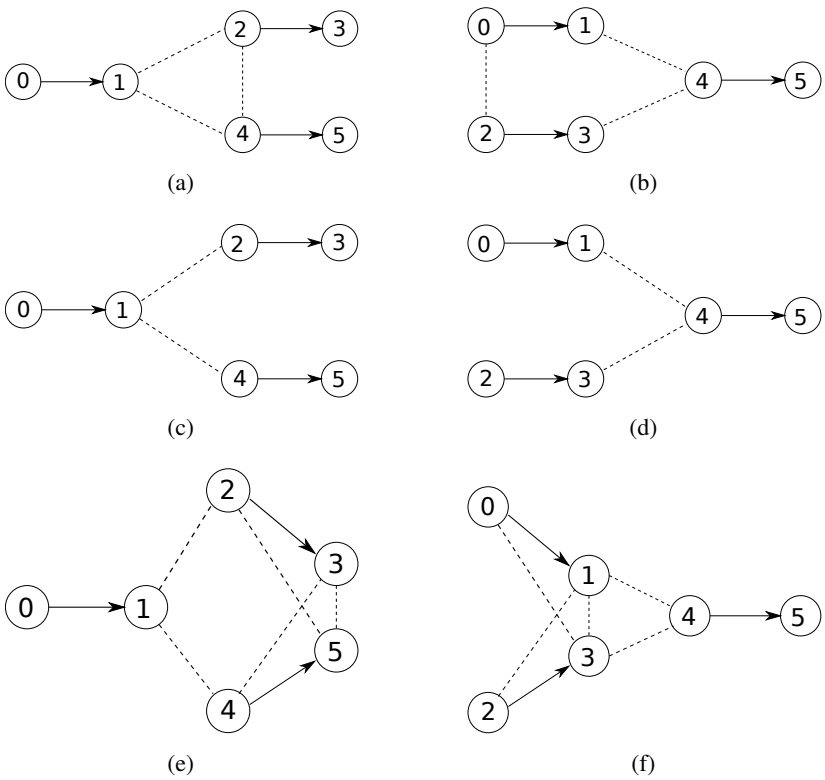


Figure 5.8: One flow competes with a symmetric cluster.

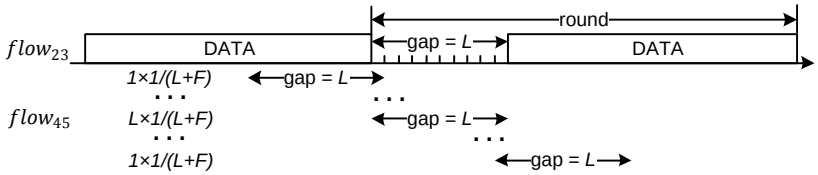
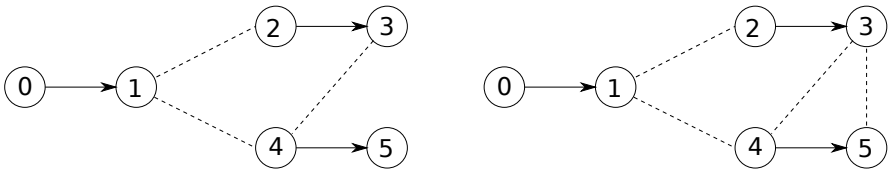


Figure 5.9: The diagram for overlapped gap.

independent of Contender 4. As a result, the start point of an arbitrary slot on the timeline of Contender 2 coincides with a start point of Contender 4's round with probability $1/(\bar{\mathcal{L}} + F)$. Thus, for a particular gap of Contender 2, there are $2\bar{\mathcal{L}} - 1$ overlapped cases, which are shown in Figure 5.9. It is simple to integrate over all $2\bar{\mathcal{L}} - 1$ cases. As a result, the expected overlapped gap length within a round is $\bar{\mathcal{L}}^2/(\bar{\mathcal{L}} + F)$. We substitute this expected length into Equation (5.2) as the parameter $\bar{\mathcal{L}}_C$. Next, by setting Equation (5.8) to $1/2$ (flow quantity ratio between clusters), the variable W could be easily quantified by W_0, F, and other constant system parameters.

Asymmetric Cluster. The two-flow cluster could be asymmetric as well. The two topologies in this category are identified in Figure 5.10. The topology in Figure 5.10 has the following two characteristics: every two flows form an asymmetric topology; if we order the flows based on which flow dominates which, we get a linear order. Based on these features, Equation (5.8) could be directly applied in the order of weak to dominant flows. More specifically, we fix W_0, set $\hat{\Re}_{02} = 1$, and derive W_2; to deduce W_4 given W_2, just simply apply the equation again.



(a) (b)

Figure 5.10: One flow competes with an asymmetric cluster.

5.5.2 Asymmetric Three-Flow Chain

Up to now, we have not addressed topologies where flows interact indirectly through other flows. We consider the simplest case of an *asymmetric chain topology* with three flows and six nodes as shown in Figure 5.11. Such a topology is different from previous cases because while $flow_{01}$ and $flow_{45}$ do not interact directly, their performances are affected by $flow_{23}$, and the interaction is indirect.

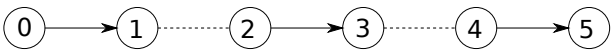


Figure 5.11: Asymmetric Chain Topology.

We outline the derivation and evaluate its accuracy in the evaluation section. First, since $flow_{45}$ dominates over $flow_{23}$, we assume that $flow_{45}$'s behavior can be modeled assuming that $flow_{23}$ and $flow_{01}$ can be considered as a single entity. We then consider the relationship between the window sizes of $flow_{01}$ and $flow_{23}$. By employing the method in [35] and our CW tuning mechanism, we can obtain the approximate optimal W_2 and W_4 given W_0 subject to the max-min fairness objective.

5.6 Evaluation

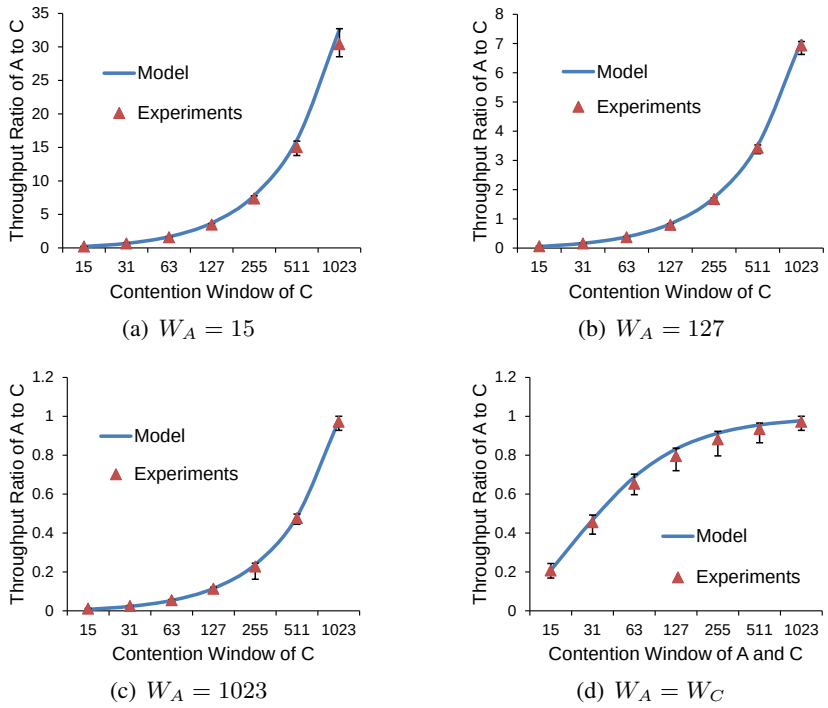
In this section, we perform a comprehensive set of experimental and simulation evaluations to validate our model. In particular, we intend to demonstrate that (i) our model can accurately predict the flow throughputs with asymmetric conditions, and (ii) our CW tuning mechanism can be used to achieve the performance objective. We perform two sets of evaluations. In the first set of experiments, we validate our models using a wireless mesh testbed. In the second set, we validate more complicated scenarios that are difficult to replicate in a testbed environment through simulation.

The default data rate for all evaluation is 12Mbps and we use UDP traffic operating at saturation mode. All data frames carry fixed payload size of 1470 bytes. RTS/CTS access mechanism is used. The default system parameters in 802.11a standard for all of the experiments and simulations are listed in Table 1.

5.6.1 Testbed Evaluation

The testbed is deployed in a college dormitory, over an area about 350m$\times$200m. This testbed consists of 20 Alix boards manufactured by PC Engines. Each board is equipped with a 500Mhz processor, 256 MB RAM, a 4GB CF card, and two Complex IEEE 802.11abg adapters featuring the Atheros AR5414 chipset. The board runs OpenWRT Kamikaze 7.09 with kernel version 2.6.25.16. The driver of the wireless adapter is MadWifi (0.9.4) with a default MAC retry limit of 10. Iperf (2.0.2) is used to generate UDP traffic.

In this section, we evaluate the accuracy of our model, in particular, Equations (5.8) and (5.10) for two flows with information, channel, and rate asymmetries using the topology in Figure 5.3(b) or Figure 5.4. Due to the limitations of the hardware, we vary the CW sizes of transmitters only with values of $2^k - 1$, where k is an integer. For each CW combination, we conducted 10 sets of runs, where each set consists of five 60-seconds run intervals. For each set, we randomly selected an asymmetric topology that fits the requirement and all other nodes are disabled.



(a) $W_A = 15$ (b) $W_A = 127$

(c) $W_A = 1023$ (d) $W_A = W_C$

Figure 5.12: The throughput ratio of $flow_{AB}$ to $flow_{CD}$, which is a function of W_A and W_C.

First, we validate the accuracy of $\hat{\Re}_{AC}$ under information asymmetric topology depicted in Figure 5.3(b). Our model is based on the perfect link assumption for RTS/CTS transmissions, which is reasonable as RTS/CTS is much shorter than data. We experimentally verified that the RTS/CTS frame loss rate on a link with RSSI of 10dB is less than 0.5%. In the evaluation, the wireless link conditions are as follows: (i) the RSSI of link AB and CD are both larger than 10dB; (ii) the RSSI of link BC and BD are higher than link AB.

The result for $\hat{\Re}_{AC}$ (equivalent to the throughput ratio of $flow_{AB}$ to $flow_{CD}$ since we use the same data length for both flows) computed using Equation (5.8) is shown in Figure 5.12 by varying W_C and for different values of W_A. The results show that our model is able to predict $\hat{\Re}_{AC}$ accurately in all the experiments. In addition, we note that severe unfairness arises in many scenarios. For example, when W_A is 15, $\hat{\Re}_{AC}$ varies from 0.21 (W_C = 15, A is almost starved) to 32 (W_C = 1023, C is

Table 5.1: $r(A) = 12Mbps$ and $r(C) = 12Mbps$

W_A	W_C	model			experiments		
		S_A	S_C	$\Re_{AC}$	S_A	S_C	$\Re_{AC}$
15	15	1.52	7.25	0.21	1.42	7.17	0.20
15	1023	8.54	0.26	32.62	8.03	0.25	32.12
127	127	3.41	4.10	0.83	3.32	4.09	0.81
1023	15	0.07	8.52	0.01	0.06	8.63	0.01
1023	1023	1.63	1.66	0.98	1.58	1.65	0.96

Table 5.2: $r(A) = 12Mbps$ and $r(C) = 6Mbps$

W_A	W_C	model			experiments		
		S_A	S_C	$\Re_{AC}$	S_A	S_C	$\Re_{AC}$
15	15	0.94	4.49	0.21	0.89	4.42	0.20
15	1023	8.36	0.26	32.62	7.92	0.25	31.68
127	127	2.53	3.04	0.83	2.41	3.03	0.80
1023	15	0.04	4.95	0.01	0.04	4.92	0.01
1023	1023	1.43	1.46	0.98	1.37	1.44	0.95

Table 5.3: $r(A) = 6Mbps$ and $r(C) = 12Mbps$

W_A	W_C	model			experiments		
		S_A	S_C	$\Re_{AC}$	S_A	S_C	$\Re_{AC}$
15	15	1.34	6.43	0.21	1.22	6.55	0.19
15	1023	4.95	0.15	32.62	4.84	0.15	32.27
127	127	2.65	3.18	0.83	2.52	3.22	0.79
1023	15	0.06	8.48	0.01	0.05	8.61	0.01
1023	1023	1.43	1.46	0.98	1.40	1.45	0.97

almost starved). Hence, it is notably difficult to find an appropriate CW combination to effectively eliminate unfairness without the prediction from the model. As shown in Figure 5.12(d), it is only when W_A is set to have the same value of W_C that the unfairness issue is less severe for window sizes above 127. However, the use of these larger window sizes leads to lower channel utilization.

Next, we evaluate the accuracy of Equation (5.10) using the channel asymmetric

topology (Figure 5.4). Each run set randomly picked a topology that satisfies (i) the weak link has RSSI larger than 10dB and (ii) the dominant link's RSSI is at least 10dB higher than the RSSI of the weak link.

The results in Table 5.1 show that our model is highly accurate. The average error over all 10 cases is 2.8%.

The experiment results presented in Tables 5.2 and 5.3 jointly consider channel asymmetry and rate asymmetry. Besides verifying the accuracy of our model, the *data length independent assumption* is also evaluated since with different rates, the data length is different as well.

We observe the following. Our model can predict the throughput ratio and throughput accurately even when there is asymmetric channel and rate. Next, even though the throughput ratio may be similar, the rate anomaly problem results in much lower total throughput in the case where the weaker flow (A) has a higher data rate. Finally, increasing the CW of lower rate flow improves fairness but may reduce throughput. There is an inherent tradeoff between fairness and throughput and our model can provide a mechanism to set the window sizes to meet a given performance objective.

In summary, the experimental results obtained show that our model is accurate for the topologies and values of contention window sizes considered.

5.6.2 Simulation Results

We now compare our model with existing approaches and validate the model under more complex topologies and scenarios using ns-2 simulation (version 2.35). In addition, since the simulator can set CW to any integer between 15 and 1023, we can also verify that tuning CW using our model can lead to a given throughput ratio or fairness objective.

First, we apply our model to the 802.11b standard and compare our CW-Tuning mechanism with two existing CW adaptive approaches: optimal CSMA (oCSMA) [38, 55] and G-Model [83]. The system configurations are exactly the same as that in [55]. With this configuration, CW-Tuning is able to achieve three fairness objectives. Ta-

Table 5.4: The throughput of A and C with information asymmetry in Figure 5.3(b), where $r(A) = r(C) = 2Mbps$.

	S_A	S_C	S_A/S_C	$S_A \times S_C$
oCSMA	0.524	0.853	0.614	0.447
G-Model	0.7	0.674	1.04	0.472
CW-Tuning	0.686	0.691	0.993	0.474

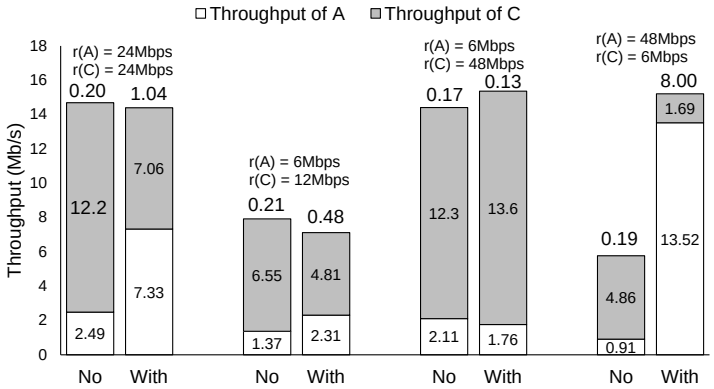


Figure 5.13: The comparison between *without* and *with* our CW Tuning mechanism (time-fairness) under information asymmetry in Figure 5.3(b). The value above the column bar represents the throughput ratio of A to C.

ble 5.4 shows the performance of these approaches. From the table, we can observe that our CW-Tuning significantly outperforms the oCSMA mechanism in terms of all three (max-min, time, and proportional) fairness objectives. The result also shows that our model performs slightly better than G-Model, which extensively search among possible CW combinations.

Second, we evaluate the accuracy of our model-based CW tuning mechanism with two flow asymmetric topology with and without rate asymmetry. The objective is time-fairness and the W_C is tuned based on Equation (5.11). Therefore, if the objective is achieved, $\hat{\Re}_{AC}$ should be the same as the ratio of the channel rate of A over channel rate of C. The result is depicted in Figure 5.13 for different data rate combinations. The result shows that our approach allows us to meet the performance objective by tuning CW.

Next, we evaluate the effectiveness of our CW tuning mechanism for achieving

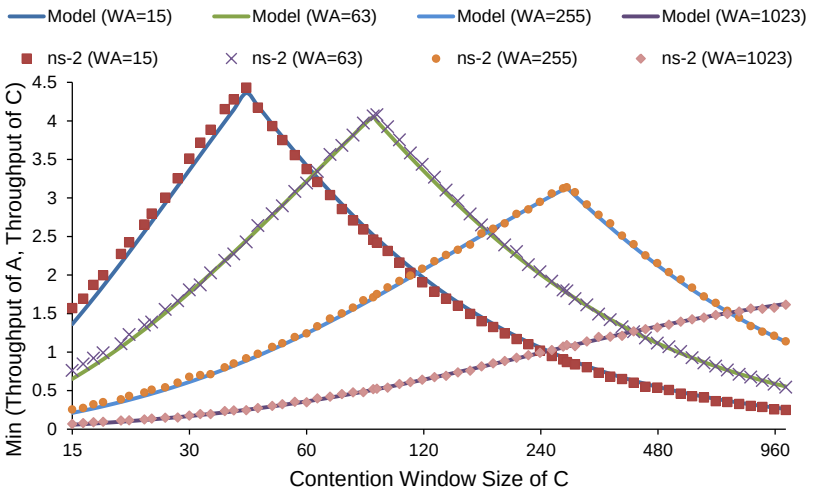


Figure 5.14: Min(S_A, S_C) under *IA* in Figure 5.3(b).

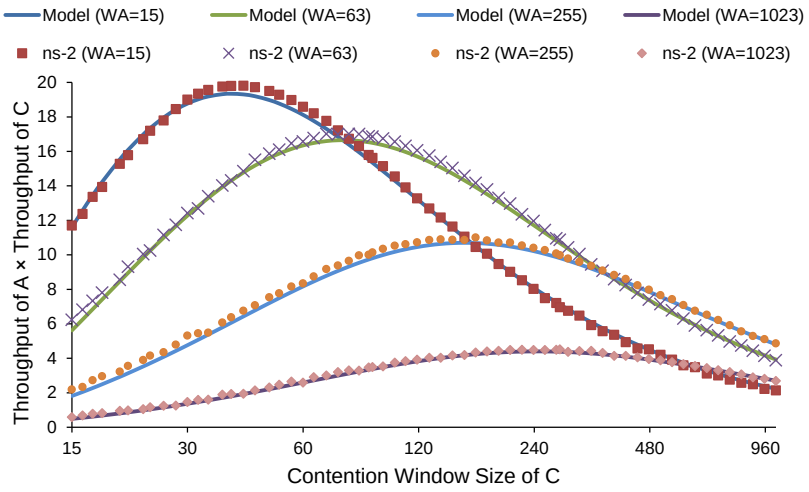


Figure 5.15: $S_A \times S_C$ under *IA* in Figure 5.3(b).

max-min and proportional fairness. The evaluation results are plotted in Figures 5.14 and 5.15, where we fix W_A and vary W_C over all possible values. For proportional fairness, the optimal point occurs when $S_A \times S_C$ is maximized. These CW tunning figures show that the model-based and simulation-based results match well. Using the result from Lemmas 1 and 2, we indicate the computed window sizes in Figures 5.14 and 5.15. The objective value of each scenario is maximized at the unique W_C as predicted.

The CW setting in our testbed is restricted to the values of $2^k - 1$, instead of any

Table 5.5: The optimal feasible settings in simulation and experiments subject to three fairness objectives.

	W_A^*	W_C^*	S_A	S_C	W_A	W_C	S_A	S_C
M-MF	15	41	4.45	4.43	31	63	4.56	4.07
TF	15	41	4.45	4.43	1023	1023	1.63	1.65
PF	15	38	4.21	4.68	15	31	3.56	5.35

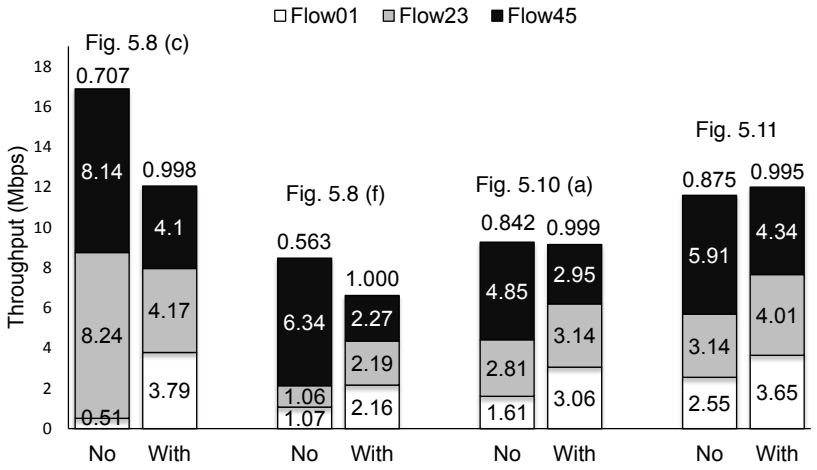


Figure 5.16: The comparison between *without* and *with* our CW Tuning (max-min) for three flows. The numbers above the column bars are the Jain's fairness index value.

integer values below 1024. To evaluate the impact of using these restricted values, we round the CW sizes obtained from our model to the nearest $2^k - 1$ for some integer k. The impact of this rounding is evaluated in Table 5.5. We observe that: (i) the performance degradation due to CW rounding is insignificant subject to either max-min fairness or proportional fairness; and (ii) without fine CW control, achieving time fairness can result in severe performance degradation.

Finally, we apply our CW tuning mechanism to obtain max-min fairness for three flow topologies from Figures 5.8, 5.10, and 5.11. The evaluation results are presented in Figure 5.16.

In the evaluations, we observe that our approach can achieve very good fairness among the three flows. The Jain's Fairness Index for all four scenarios are at least 0.995. On the other hand, the index varies from 0.563 to 0.875 without window tuning when

the CW size is always set to the default value of 15. For the topology in Figure 5.8(c), $flow_{01}$ is starved and the other two flows transmit at high throughput. By making the system fair, these two flows have to increase their window sizes substantially and the total throughput is significantly reduced. This scenario is similar for the topology in Figure 5.8(f), where $flow_{01}$ and $flow_{23}$ will be given much throughput while reducing throughput of $flow_{45}$. On the other hand, for the other 2 topologies, fairness can be achieved without sacrificing throughput.

5.7 Summary

In this chapter, we analytically characterize the impacts of CSMA contention windows under asymmetric conditions. The simplicity of the model clearly indicates and quantifies the paramount factors needed to optimize fairness metrics. Our extensive measurements in both testbed and simulation indicate that our CW tuning mechanism is effective in terms of both fairness and throughput. Our model lays the foundation for practical contention window tuning mechanisms in CSMA protocols, enabling fine control of throughputs among two or more heterogeneous multicast streaming flows.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The main objective of this thesis is to enhance the wireless video streaming system performance by addressing the inefficiencies in wireless transmission protocols. In this final chapter, we review the research contributions of our work, followed by the discussions on future work.

6.1 Conclusions

In conclusion, this thesis explored the adaptive mechanisms for wireless video streaming. In Chapter 1, we pointed out that streaming HD video over wireless channels to multiple users suffers from issues in fairness, stability, and efficiency of resource utilization. These ineffectivenesses are primarily attributed to the bandwidth limitations and the unpredictable nature of heterogeneous wireless networks. To address the observed inefficiencies, we focused on adaptive techniques which improve the streaming performance and system scalability by dynamically adapting wireless multicast data

rate (including application-layer and link-layer rates), AP association control, and contention window (CW) sizes.

One problem with streaming zoomable video to multiple users was that the wireless multicast is difficult to be applied to save the bandwidth consumption of transmissions. To resolve this limitation, we employed a new tiling approach that allows different tile resolutions to be mixed in a single video frame. As the perceptual quality of this approach was unclear, Chapter 3 conducted a psychophysical experiment to evaluate the perceptual quality impact. The experiments results demonstrated that in most cases, we can save more than 25% bandwidth consumption while the perceptual quality of mixed-resolutions tiled video is still acceptable.

Applying the new tiling format, Chapter 3 then jointly adapted the wireless multicast link rate and video quality to exploit the available network resources. In particular, we formulated the adaptive tiled video multicast streaming as an optimal resource allocation problem; an effective algorithm was designed to optimally solve this problem. The evaluation results show that our proposed optimal multicast mechanism has improved the average video quality by up to 12dB, 6dB, 3dB in terms of PSNR compared with three baseline schemes, *adaptive unicast*, *adaptive multicast*, and *approximation multicast*, respectively. Additionally, our designed algorithm can be applied to the allocation of multi-sessions video multicast as well, and has a lower, more practical, running time than the existing optimal allocation algorithm.

In Chapter 4, we applied zoomable video multicast allocation algorithm to the adaptive video streaming of general multi-sessions multicast allocation problem. Additionally, the wireless network has been extended from consisting of a single AP to multiple APs, where neighboring APs are operating on distinct non-overlapping channels. We presented JurCast, a joint user and rate allocation scheme for video multicast allocation over multiple APs. Our extensive measurements in both bestbed and large scale simulation demonstrated that JurCast significantly outperforms other compared adaptive schemes with up to 3dB and 55% enhancements in terms of PNSR and good-

put, respectively.

Chapter 5 concentrated on the heterogeneous topologies where two or more APs are operating on the same wireless channel. Both the existing studies and our experiment results identified that in such scenarios, wireless transmissions from different APs could suffer from severe unfairness or even starvation, especially for saturated traffic conditions, such as the HD video streaming applications. Unfortunately, the state of art in solving the unfairness was rather primitive, which is attributed to the complexity of the analytical model. In this chapter, we suggested a novel model to analyze the performance of each transmission flow given the corresponding contention window sizes. Applying the analytical model, we derived a simple and effective dynamic adaptive CW tuning approach. The evaluation results indicated that our model-based adaptive CW tuning mechanism is able to achieve optimal fairness.

6.2 Future Work

This study in adaptive wireless delivery solutions has contributed toward effective wireless transmissions to support satisfactory service for HD videos. A number of open problems remaining to be addressed to allow the development of a general and practical large-scale wireless streaming system. These problems suggest a variety of research directions that need to be pursued to make such a system feasible.

6.2.1 Integrating Multicast to DASH

Most recently, *Dynamic Adaptive Video Streaming over HTTP* (DASH) [65, 10, 52] becomes one of the most popular adaptive streaming techniques. Our work, however, concentrates on adapting last-mile multicast delivery (at link layer), which is a UDP-based framework. To develop a general and practical adaptive streaming system, a problem that needs to be solved is integrating our adaptive multicast transmission to DASH streaming.

One feasible approach is to leverage hybrid FLUTE (File Delivery over Unidirectional Transport)/DASH video delivery scheme, which is proposed in several recent

3GPP specifications [1, 2] and studies [66, 43, 13]. In MBMS (Multimedia Broadcast Multicast Services) broadcast, the formats defined by DASH can be utilized for the multicast delivery over non-HTTP/TCP networks. Moreover, the FLUTE protocol defined in RFC3926 [5] permits to deliver video segments over MBMS such that the client observes them being delivered over HTTP/TCP. Thus, FLUTE and DASH can be seamlessly integrated into a hybrid broadcast/multicast streaming scheme. In addition to defining the segment format and metadata, the DASH session can be utilized for frame loss recovery over unicast (suggested in [13]).

To apply our adaptive multicast algorithm, we can leverage the deployed gateway (shown in Figure 4.1) as a proxy between wireless nodes and the Internet. Apart from running the allocation algorithm to determine the video frames to be multicast, this proxy consists of the following components: (i) aggregating the HTTP requests from clients and generating the minimum necessary requests to fetch video data from video servers; (ii) maintaining unicast TCP connections to clients for DASH MPD (media presentation description) and frame loss recovery; (iii) multicast video data to mobile clients as a FLUTE server.

By aggregating and regenerating the HTTP requests, we could significantly reduce the amount of streaming traffic over the Internet. On the other hand, the hybrid FLUTE/DASH delivery scheme is able to considerably save wireless bandwidth consumption than conventional DASH streaming.

6.2.2 Exploiting Mixed Resolutions Tiling

Another possible avenue of future work would be to exploit the scheme of *mixing tile resolutions*, which includes the following potential extensions over our existing approach.

Our tile resolution allocation scheme employs the metrics, bandwidth or tile quality (PSNR), regardless of the video content. The psychophysical assessment in Chapter 3 clearly demonstrates that the perceived video quality closely depends on video motion density. Therefore, if we can allocate tile resolutions more intelligently (e.g.,

lower resolutions on regions with low motion), we can further reduce the bandwidth consumption without considerably impairing the perceived quality.

We also plan to study how mixing tile resolutions can be applied to benefit the general multi-sessions multicast streaming (Chapter 4) and DASH streaming. Applying mixing tile resolutions improves the loss resilience of streaming from the following two aspects: less severe video quality impairment as the tiles are coded independently; and less overhead for loss recovery as a finer region size (tile) is used rather than the entire frame. Although this tiling scheme will increase the video coding overhead, using an appropriate tile size may still benefit time-constraint live video streaming.

6.2.3 Multicast Link Rate Selection

Multicast link rate selection is an essential factor that determines the performance of our adaptive multicast streaming approach. Under the same network condition, higher transmission rate leads to higher packet loss rate. To preserve the low frame loss rate, conservative threshold parameters are used in our current rate adaptation mechanism, which may underutilize our wireless channels.

The prior work [17] on unicast transmission selection shows that the throughput curve increases with higher transmission rate as long as the packet loss rate under the currently selected rate is not substantial increased. In multicast video streaming system, if a more aggressive multicast link rate is used for delivery, more redundant FEC packets are required to protect from packet loss. In the future work, we plan to incorporate this trade-off into our optimal allocation scheme.

6.2.4 Contention Window Adaptation with Multicast Scheduling

In Chapter 5, we present a model to optimally quantify the contention window adaptation under asymmetric topology. Our CW tunning mechanism can be directly applied to the asymmetric multicast flows, as shown in Figure 6.1 (when client 6 is absent). More specifically, clients 1, 2, and 3 can be clustered as one super node, and clients 4 and 5 can be clustered as another node.

The practical general networks, however, is a hybrid asymmetric/symmetric topol-

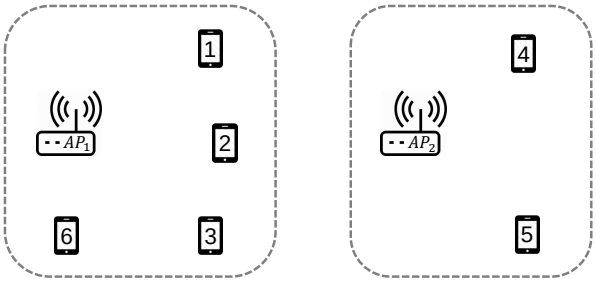


Figure 6.1: An asymmetric multicast topology. In this topology, four clients 1, 2, 3, and 6 are associated with AP_1, the other three clients are associated with AP_2. Clients 1, 2, and 3 can observe transmissions from both APs, while clients 4 and 5 are within transmission range of AP_2, and client 6 is within transmission range of AP_1.

ogy when multiple mobile clients are present. Our current CW tuning mechanism cannot solve such hybrid topologies, which is the major limitation of our CW tuning approach. To overcome this limitation, one possibility would be to extend the CW tuning with the multicast scheduling. In particular, we can first cluster the mobile clients to several super nodes and build a network topology accordingly, and then decide how to schedule these multicast flows to achieve the optimal system performance. For instance, an adaptive multicast scheduling approach can be suggested to determine client 6 to be scheduled simultaneously with clients 4 and 5 (avoid collision) or with clients 1-3 (exploit multicast).

6.3 Summary

In summary, we have designed a framework that dynamically adapts wireless transmission for video streaming applications. The framework can be applied to solve the well-known challenging problems in wireless networks. The extensive experiments demonstrated that applying our suggested adaptive schemes have both improved the performance and increased the scalability of our wireless streaming systems.

BIBLIOGRAPHY

[1] 3GPP TS 26.247. Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH). 106

[2] 3GPP TS 26.346. Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs. 106

[3] ONOE Rate Control. http://madwifi.org/browser/trunk/ath_rate/onoe. 16

[4] OpenRQ: an open-source RaptorQ implementation. http://www.lasige.di.fc.ul.pt/openrq/. 65

[5] RFC 3926. FLUTE-File Delivery over Unidirectional Transport. 2004. 106

[6] Apple Inc. Mac OS X Server QuickTime Streaming and Broadcasting Administration. 2007. 13

[7] IEEE 802.11-2007. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE 802.11 LAN Standards 2007*, 2007. 40

[8] Move Networks. Internet television: Challenges and opportunities. Technical report, Move Networks, Inc. 2008. 13

[9] A. Zambelli. IIS Smooth Streaming technical overview. Technical report, Microsoft Corporation. 2009. 13

[10] S. Akhshabi, A. C. Begen, and C. Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP. In *Proceedings of the 2nd Annual ACM Conference on Multimedia Systems (MMSys)*, pages 157–168, San Jose, California, USA, 2011. ACM. 3, 12, 105

[11] A. Balachandran, P. Bahl, and G. M. Voelker. Hot-spot congestion relief and service guarantees in public-area wireless networks. *ACM SIGCOMM Computer Communication Review*, 32(1):59–59, 2002. 19

[12] Y. Bejerano, S.-J. Han, and L. E. Li. Fairness and load balancing in wireless LANs using association control. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 315–329, Philadelphia, Pennsylvania, USA, 2004. ACM. 19

[13] R. Belda, I. Fez, F. Fraile, P. Arce, and J. C. Guerri. Hybrid FLUTE/DASH video delivery over mobile wireless networks. *Transactions on Emerging Telecommunications Technologies*, 25(11):1070–1082, 2014. 106

[14] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless LANs. In *ACM SIGCOMM Computer Communication Review*, 1994. 4, 21, 76, 77, 80, 82

[15] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on*, 18(3):535–547, 2000. 20

[16] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello. Experimental assessment of the backoff behavior of commercial IEEE 802.11b network cards. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1181–1189, Anchorage, Alaska, USA, 2007. IEEE. 82

[17] J. C. Bicket. *Bit-rate selection in wireless networks*. PhD thesis, Massachusetts Institute of Technology, 2005. 2, 16, 44, 107

[18] F. Calì, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *Networking, IEEE/ACM Transactions on*, 8(6):785–799, 2000. 20, 21

[19] J. Camp, E. Aryafar, and E. Knightly. Coupled 802.11 flows in urban channels: model and experimental evaluation. *Networking, IEEE/ACM Transactions on*, 20(5):1452–1465, 2012. 8, 21, 76, 77, 81

[20] J. Camp and E. Knightly. Modulation rate adaptation in urban and vehicular environments: cross-layer implementation and experimental evaluation. *Networking, IEEE/ACM Transactions on*, 18(6):1949–1962, 2010. 2, 16

[21] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee, and L. Ravindranath. Dircast: A practical and efficient Wi-Fi multicast system. In *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP)*, pages 161–170, Princeton, New Jersey, USA, 2009. IEEE. 6, 17, 20, 46, 56, 63, 66

[22] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 72–83. Springer, 2004. 60

[23] A. Chen, D. Lee, and P. Sinha. Optimizing multicast performance in large-scale WLANs. In *Proceedings of the 27th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 17–17, Toronto, Ontario, Canada, 2007. IEEE. 19, 56, 60, 63

[24] S. Das, H. Viswanathan, and G. Rittenhouse. Dynamic load balancing through coordinated scheduling in packet data systems. In *Proceedings of the 22nd IEEE International Conference on Computer Communications (INFOCOM)*, pages 786–796, San Francisco, California, USA, 2003. IEEE. 19

[25] S. Deb, S. Jaiswal, and K. Nagaraj. Real-time video multicast in WiMAX networks. In *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM)*, Phoenix, Arizona, USA, 2008. IEEE. 17, 18

[26] W. C. Feng, T. Dang, J. Kassebaum, and T. Bauman. Supporting region-of-interest cropping through constrained compression. In *Proceedings of the 16th ACM International Conference on Multimedia*, pages 745–748, Vancouver, BC, Canada, 2008. ACM. 14

[27] W. C. Feng, T. Dang, J. Kassebaum, and T. Bauman. Supporting region-of-interest cropping through constrained compression. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 7(3):17, 2011. 14

[28] M. Garetto, J. Shi, and E. Knightly. Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 200–214, Cologne, Germany, 2005. 4, 8, 21, 76, 77, 80, 82, 84

[29] G. A. Gescheider. *Psychophysics: the fundamentals*. Psychology Press, 1997. 25, 31

[30] J. Goodman, A. Greenberg, N. Madras, and P. March. Stability of binary exponential backoff. *Journal of the ACM (JACM)*, 35(3):579–602, 1988. 78

[31] S. Halawa, D. Pang, N.-M. Cheung, and B. Girod. ClassX: an open source interactive lecture StreamingSystem. In *Proceedings of the 19th ACM International Conference on Multimedia*, pages 719–722, Scottsdale, Arizona, USA, 2011. ACM. 3, 23

[32] B. Han, A. Schulman, F. Gringoli, N. Spring, B. Bhattacharjee, L. Nava, L. Ji, S. Lee, and R. Miller. Maranello: practical partial packet recovery for 802.11. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, pages 205–218, San Jose, California, USA, 2010. 82

[33] J. Håstad, T. Leighton, and B. Rogoff. Analysis of backoff protocols for multiple access channels. *SIAM Journal on Computing*, 25(4):740–774, 1996. 78

[34] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11 b. In *Proceedings of the 22nd IEEE International Conference on Computer Communications (INFOCOM)*, pages 836–843, San Francisco, California, USA, 2003. 3, 20, 76, 81

[35] M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs. *ACM SIGCOMM Computer Communication Review*, 35(4):121–132, 2005. 8, 20, 21, 77, 81, 94

[36] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 236–251, Rome, Italy, 2001. ACM. 2, 16

[37] C. V. N. Index. Global mobile data traffic forecast update, 2014-2019. *White Paper, February*, 2015. 1

[38] L. Jiang and J. Walrand. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. *Networking, IEEE/ACM Transactions on*, 18(3):960–972, 2010. 21, 83, 98

[39] L. Kleinrock, F. Tobagi, et al. Packet switching in radio channels: Part I–Carrier sense multiple-access modes and their throughput-delay characteristics. *Communications, IEEE Transactions on*, 23(12):1400–1416, 1975. 77

[40] A. Kochut, A. Vasan, A. U. Shankar, and A. Agrawala. Sniffing out the correct physical layer capture model in 802.11 b. In *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP)*, pages 252–261, Berlin, Germany, 2004. IEEE. 20, 76, 81

[41] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click modular router. *ACM Transactions on Computer Systems (TOCS)*, 18(3):263–297, 2000. 43, 64

[42] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed point analysis of single cell IEEE 802.11 WLANs. In *Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1550–1561, Miami, Florida, USA, 2005. IEEE. 20

[43] D. Lecompte and F. Gabin. Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: overview and Rel-11 enhancements. *Communications Magazine, IEEE*, 50(11):68–74, 2012. 106

[44] P. Li, H. Zhang, B. Zhao, and S. Rangarajan. Scalable video multicast in multi-carrier wireless data systems. In *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP)*, pages 141–150, Princeton, New Jersey, USA, 2009. IEEE. 18, 24, 41

[45] P. Li, H. Zhang, B. Zhao, and S. Rangarajan. Scalable video multicast with joint layer resource allocation in broadband wireless networks. In *Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP)*, pages 295–304, Kyoto, Japan, 2010. IEEE. 18, 41, 46

[46] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer. RaptorQ Forward Error Correction Scheme for Object Delivery. Technical report, RFC 6330, August, 2011. 65

[47] E. Magistretti, O. Gurewitz, and E. Knightly. 802.11 ec: collision avoidance without control messages. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 65–76, Istanbul, Turkey, 2012. ACM. 87

[48] D. Malone, K. Duffy, and D. Leith. Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions. *Networking, IEEE/ACM Transactions on*, 15(1):159–172, 2007. 20

[49] A. Mavlankar, P. Agrawal, D. Pang, S. Halawa, N.-M. Cheung, and B. Girod. An interactive region-of-interest video streaming system for online lecture viewing. In *18th International Packet Video Workshop (PV)*, pages 64–71, Hong Kong, China, 2010. IEEE. 3, 23

[50] A. Mavlankar, P. Baccichet, D. Varodayan, and B. Girod. Optimal slice size for streaming regions of high resolution video with virtual pan/tilt/zoom functionality. In *Proceedings of 15th European Signal Processing Conference (EUSIPCO)*, Poznan, Poland, 2007. 13

[51] A. Mavlankar, D. Varodayan, and B. Girod. Region-of-interest prediction for interactively streaming regions of high resolution video. In *16th International Packet Video Workshop (PV)*, pages 68–77, Lausanne, Switzerland, 2007. IEEE. 13

[52] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang. QDASH: a QoE-aware DASH system. In *Proceedings of the 3rd Annual ACM Conference on Multimedia Systems (MMSys)*, pages 11–22, Chapel Hill, North Carolina, USA, 2012. ACM. 3, 12, 105

[53] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill. Designing high performance enterprise Wi-Fi networks. In *Proceedings of the 5th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, pages 73–88, San Francisco, California, USA, 2008. 19, 56

[54] B. Nardelli and E. Knightly. Closed-form throughput expressions for CSMA networks with collisions and hidden terminals. In *Proceedings of the 31st IEEE International Conference on Computer Communications (INFOCOM)*, pages 2309–2317, Orlando, Florida, USA, 2012. IEEE. 8, 21, 77, 83

[55] B. Nardelli, J. Lee, K. Lee, Y. Yi, S. Chong, E. Knightly, and M. Chiang. Experimental evaluation of optimal CSMA. In *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1188–1196, Shanghai, China, 2011. IEEE. 21, 77, 80, 82, 83, 98

[56] I. Pefkianakis, S. Wong, H. Yang, S. Lee, and S. Lu. Towards history-aware robust 802.11 rate adaptation. *Mobile Computing, IEEE Transactions on*, 12(3):502–515, 2013. 44, 64

[57] K. Piamrat, A. Ksentini, J.-M. Bonnin, and C. Viho. Q-DRAM: QoE-based dynamic rate adaptation mechanism for multicast in wireless networks. In *Global Telecommunications Conference, GLOBECOM 2009.*, pages 1–6, Honolulu, Hawaii, USA, 2009. IEEE. 17

[58] L. Qiu, Y. Zhang, F. Wang, M. Han, and R. Mahajan. A general model of wireless interference. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 171–182, Montreal, Quebec, Canada, 2007. ACM. 21

[59] N. Quang Minh Khiem, G. Ravindra, A. Carlier, and W. T. Ooi. Supporting zoomable video streams with dynamic region-of-interest cropping. In *Proceedings of the 1st Annual ACM Conference on Multimedia Systems (MMSys)*, pages 259–270, Scottsdale, Arizona, USA, 2010. ACM. 13, 14, 23, 47

[60] N. Quang Minh Khiem, G. Ravindra, and W. T. Ooi. Adaptive encoding of zoomable video streams based on user access pattern. In *Proceedings of the 2nd Annual ACM Conference on Multimedia Systems (MMSys)*, pages 211–222, San Jose, California, USA, 2011. ACM. 14

[61] S. Sen, N. K. Madabhushi, and S. Banerjee. Scalable WiFi media delivery through adaptive broadcasts. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, pages 13–13, San Jose, California, USA, 2010. USENIX Association. 17, 66

[62] A. Shafiei, Q. M. K. Ngo, R. Guntur, M. K. Saini, C. Pang, and W. T. Ooi. Jiku live: a live zoomable video streaming system. In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 1265–1266, Nara, Japan, 2012. ACM. 3, 23

[63] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. Knightly. Measurement and modeling of the origins of starvation in congestion controlled mesh networks. In *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM)*, Phoenix, Arizona, USA, 2008. IEEE. 21

[64] A. Shokrollahi. Raptor codes. *Information Theory, IEEE Transactions on*, 52(6):2551–2567, 2006. 65

[65] T. Stockhammer. Dynamic adaptive streaming over HTTP–: standards and design principles. In *Proceedings of the 2nd Annual ACM Conference on Multimedia Systems (MMSys)*, pages 133–144, San Jose, California, USA, 2011. ACM. 3, 12, 105

[66] T. Stockhammer and M. G. Luby. Dash in mobile networks and services. In *Visual Communications and Image Processing (VCIP), 2012 IEEE*, pages 1–6, San Diego, California, USA, 2012. IEEE. 106

[67] Y. Tay. Analytical Performance Modeling for Computer Systems. *Synthesis Lectures on Computer Science*, 4(3):1–141, 2013. 77, 85

[68] Y. Tay and K. Chua. A capacity analysis for the IEEE 802.11 MAC protocol. *Wireless networks*, 2001. 20, 85

[69] F. Tobagi, L. Kleinrock, et al. Packet switching in radio channels: Part II–The hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *Communications, IEEE Transactions on*, 23(12):1417–1433, 1975. 79

[70] A. Tsertou and D. Laurenson. Revisiting the hidden terminal problem in a CSMA/CA wireless network. *Mobile Computing, IEEE Transactions on*, 7(7):817–831, 2008. 8, 77

[71] R. van Brandenburg, O. Niamut, M. Prins, and H. Stokking. Spatial segmentation for immersive media delivery. In *Proceedings of the 15th International Conference on Intelligence in Next Generation Networks (ICIN)*, pages 151–156, Paris, France, 2011. IEEE. 13

[72] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. *ACM SIGCOMM Computer Communication Review*, 39(4):3–14, 2009. 2, 16

[73] H. Wang, M. C. Chan, and W. T. Ooi. Wireless multicast for zoomable video streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) (to appear)*, 2015. 66

[74] H. Wang, V.-T. Nguyen, W. T. Ooi, and M. C. Chan. Mixing tile resolutions in tiled video: a perceptual quality assessment. In *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop (NOSSDAV)*, pages 25–30, Singapore, 2014. ACM. 15

[75] H. Wang, W. T. Ooi, and M. C. Chan. Modeling CSMA/CA network under asymmetric conditions. In *Proceedings of the 8th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, pages 67–74, Barcelona, Spain, 2013. ACM. 4

[76] W. Wang, B. Leong, and W. T. Ooi. Mitigating unfairness due to physical layer capture in practical 802.11 mesh networks. *Mobile Computing, IEEE Transactions on*, 14(1):99–112, 2015. 4, 8, 21, 77

[77] X. Wang and K. Kar. Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks. In *Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM)*, pages 23–34, Miami, Florida, USA, 2005. 21

[78] S. H. Wong, R. Raghavendra, Y. Song, and K.-W. Lee. X-wing: A high-speed wireless broadcasting framework for ieee 802.11 networks. In *Proceedings of the 10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 344–352, New Orleans, USA, 2013. IEEE. 17

[79] S. H. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 146–157, Los Angeles, California, USA, 2006. ACM. 2, 16, 44

[80] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma. Performance of reliable transport protocol over IEEE 802.11 wireless LAN: analysis and enhancement. In *Proceedings of the 21st IEEE International Conference on Computer Communications (INFOCOM)*, New York, NY, USA, 2002. 20

[81] J. Yoon, H. Zhang, S. Banerjee, and S. Rangarajan. MuVi: a multicast video delivery scheme for 4G cellular networks. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 209–220, Istanbul, Turkey, 2012. ACM. 18, 24, 66

[82] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A practical SNR-guided rate adaptation. In *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM)*, Phoenix, Arizona, USA, 2008. IEEE. 2, 16

[83] A. Zhou, M. Liu, Z. Li, and E. Dutkiewicz. Modeling and Optimization of Medium Access in CSMA Wireless Networks with Topology Asymmetry. *Mobile Computing, IEEE Transactions on*, 11(9):1559–1571, 2012. 8, 21, 77, 83, 98