AN ABSTRACT OF THE THESIS OF

Monchai Lertsutthiwong for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on December 1, 2010.

Title: Protocol Design and Optimization for Wireless Networks

Abstract approved: Thinh Nguyen

We investigate a number of techniques for increasing throughput and quality of media applications over wireless networks. A typical media communication application such as video streaming imposes strict requirements on the delay and throughout of its packets, which unfortunately, cannot be guaranteed by the underlying wireless network due inherently to the multi-user interference and limited bandwidth of wireless channels. Therefore, much recent research has been focused on the joint design of network layers in order to guarantee some pre-specified Quality of Service (QoS). In this thesis, we investigate three specific settings to address the general problem of media transmission over wireless networks. In the first setting, we propose a distributed admission control algorithm in one-hop wireless network to decide whether or not a new flow should be injected into the network, in order to guarantee the QoS of the current flows. Next, a novel medium access control protocol and a scheduling packet algorithm
are proposed for jointly optimizing the quality of video streaming applications. In the second setting, we extend the framework of the proposed admission control from a one-hop network to linear wireless networks, consisting of multiple nodes. In the third and final setting, we present an approach for increasing the throughput of wireless access networks by integrating network coding and beamforming techniques.
Protocol Design and Optimization for Wireless Networks

by

Monchai Lertsutthiwong

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented December 1, 2010
Commencement June 2011
Doctor of Philosophy thesis of Monchai Lertsutthiwong presented on December 1, 2010.

APPROVED:

__________________________
Major Professor, representing Electrical and Computer Engineering

__________________________
Director of the School of Electrical Engineering and Computer Science

__________________________
Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

__________________________
Monchai Lertsutthiwong, Author
ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor, Prof. Thinh Nguyen, for friendly motivation, enthusiastic attitude, and support throughout my graduate study at Oregon State University. This thesis would have not been possible without his scientific guidance and expertise. I also would like to express my sincere appreciation to Prof. Bella Bose, Prof. Alan Fern, and Prof. Bechir Hamdaoui for providing deeply insight of my research. I would like to thank Prof. Mario Magana for his advice. I would like to thank Prof. Raviv Raich for his friendly guidance and discussion while working with him. I would like to thank Prof. Sho Kimura for serving on my committee.

Second, I would like to thank my professional colleagues at Motorola Inc., Dr. Dragan Boscovic, Dr. Faramak Vakil, and Dr. Mehran Bagheri, for their guidance and insight of world-class technologies and business. Especially Dr. Faramak Vakil, I really appreciate his full support, friendly discussion on novel invention, and giving me opportunities to be with Motorola Inc.

Third, I would like to thank my current and former colleagues in Multimedia Information Processing and Networking (MAIN) Lab, Tuan Tran, Seungyun Cho, Dong Nguyen, and Kien Nguyen for their warm friendship and pleasant
discussions. I would like to thank my good friends, Nirut Chalainanont, Weetit Wanalertlak, Panupat Poocharoen, Sirikarn Woracheewan, and Hon Hung Tang, for their help and support in everything.

Finally, I would like to gratefully express my sincere gratitude to my family for their caring and encouragement.
TABLE OF CONTENTS

1 Introduction
   1.1 Overview of Wireless Networks .......................... 3
   1.2 Medium Access Control (MAC) Protocols .................. 5
   1.3 SDMA and Beamforming Technique ......................... 6
   1.4 Quality of Service ...................................... 8
   1.5 Thesis Contributions and Organization ................. 10

2 Scalable Video Streaming for Single-Hop Wireless Networks using a Contention-Based Access MAC Protocol
   2.1 Introduction ............................................. 13
   2.2 Related Work ............................................ 17
   2.3 MAC Protocol ............................................ 20
      2.3.1 Contention-Based Access Mechanism .................. 20
      2.3.2 Proposed MAC Protocol ............................... 22
   2.4 Admission Control Framework ............................ 31
      2.4.1 Architecture ........................................ 31
      2.4.2 Problem Formulation ................................ 32
   2.5 Computational Complexity of Layer Optimization ....... 34
      2.5.1 Computing Optimal Solutions ....................... 35
      2.5.2 Computing Approximate Solutions .................... 44
   2.6 Simulation Results ....................................... 46
      2.6.1 Protocol Evaluation .................................. 47
      2.6.2 Layer Allocation Algorithm Performance ............ 58
      2.6.3 Throughput Jitter Evaluation ....................... 61
      2.6.4 Overall Performance ................................ 66
   2.7 Conclusions .............................................. 68

3 On Throughput Guarantee of Aloha-Like Multi-Hop Wireless Networks 70
   3.1 Introduction .............................................. 70
   3.2 Preliminaries ............................................. 73
      3.2.1 Related Work ....................................... 73
      3.2.2 Characteristics of Wireless Networks ............... 74
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.3 Aloha Protocol</td>
<td>75</td>
</tr>
<tr>
<td>3.2.4 IEEE 802.11 Protocol</td>
<td>76</td>
</tr>
<tr>
<td>3.3 Proposed MAC Protocol</td>
<td>77</td>
</tr>
<tr>
<td>3.4 Feasibility Algorithm</td>
<td>79</td>
</tr>
<tr>
<td>3.5 Variable elimination technique</td>
<td>85</td>
</tr>
<tr>
<td>3.6 Simulations and experimental results</td>
<td>87</td>
</tr>
<tr>
<td>3.7 Conclusions</td>
<td>91</td>
</tr>
<tr>
<td>4 Efficient Wireless Broadcasting Through Joint Network Coding and Beam-</td>
<td>94</td>
</tr>
<tr>
<td>forming</td>
<td></td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>94</td>
</tr>
<tr>
<td>4.2 Related Work</td>
<td>98</td>
</tr>
<tr>
<td>4.3 Network Coding and Broadcasting</td>
<td>100</td>
</tr>
<tr>
<td>4.4 Joint Beamforming and Network Coding</td>
<td>102</td>
</tr>
<tr>
<td>4.5 Proposed Scheduling Algorithms</td>
<td>111</td>
</tr>
<tr>
<td>4.5.1 Technique 1: Greedy Selection-Based Scheduler</td>
<td>111</td>
</tr>
<tr>
<td>4.5.2 Technique 2: Codebook-Based Scheduler</td>
<td>114</td>
</tr>
<tr>
<td>4.6 Performance Evaluation</td>
<td>117</td>
</tr>
<tr>
<td>4.6.1 Uniform MS distribution</td>
<td>119</td>
</tr>
<tr>
<td>4.6.2 Clustered MS distribution</td>
<td>121</td>
</tr>
<tr>
<td>4.6.3 Impact of number of MSs</td>
<td>124</td>
</tr>
<tr>
<td>4.7 Conclusion</td>
<td>124</td>
</tr>
<tr>
<td>5 Conclusions and Future Work</td>
<td>127</td>
</tr>
<tr>
<td>5.1 Contributions</td>
<td>127</td>
</tr>
<tr>
<td>5.2 Future Work</td>
<td>130</td>
</tr>
<tr>
<td>Bibliography</td>
<td>131</td>
</tr>
<tr>
<td>Appendices</td>
<td>138</td>
</tr>
<tr>
<td>A.1 Approximation Bound of Double-Greedy Algorithms</td>
<td>139</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.2 Approximation Bound of Triple-Greedy Algorithms</td>
<td>143</td>
</tr>
<tr>
<td>A.3 Hardness Results of a BF-NC Scheduling Problem</td>
<td>145</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Timing diagram with RTS/CTS for our proposed framework</td>
<td>28</td>
</tr>
<tr>
<td>2.2</td>
<td>Performance comparisons for the proposed protocol versus IEEE 802.11 and IEEE 802.11e protocols; (a) Throughput for flow 1; (b) Throughput for flow 2; (c) Throughput for flow 3; (d) Overall collision; (e) Overall bandwidth usage.</td>
<td>51</td>
</tr>
<tr>
<td>2.3</td>
<td>Proposed protocol validation with 30 flows over 54Mbps bandwidth. (a) Throughput for flows 1 and 2; (b) Throughput for flows 3-10; (c) Throughput for flows 11-20; (d) Throughput for flows 21-30; (e) Overall bandwidth usage; (f) Transmission probability.</td>
<td>53</td>
</tr>
<tr>
<td>2.4</td>
<td>Performance results using the greedy algorithm. (a) Throughput; (b) Transmission probability; (c) Percentage of S slots; (d) Percentage of I, C, and S slots; (e) Distortion in MSE; (f) Rate-distortion characteristic.</td>
<td>55</td>
</tr>
<tr>
<td>2.5</td>
<td>Distortion performances of different algorithms for (a) Video profiles in Table 2.1; (b) video profiles in Table 2.2.</td>
<td>57</td>
</tr>
<tr>
<td>2.6</td>
<td>Throughput jitter for each flow for (a) Video profiles in Table 2.1; (b) Video profiles in Table 2.2.</td>
<td>62</td>
</tr>
<tr>
<td>2.7</td>
<td>Percentage of late packets as a function of streaming buffer length (in seconds); (a) 0% bandwidth safety margin; (b) 1% bandwidth safety margin; (c) 2% bandwidth safety margin; (d) 3% bandwidth safety margin. The bit rates of Akiyo, Coastguard, and Foreman are 192, 160, and 256kbps, respectively.</td>
<td>64</td>
</tr>
<tr>
<td>2.8</td>
<td>Normalized throughput standard deviation as a function of buffer size (T).</td>
<td>65</td>
</tr>
<tr>
<td>2.9</td>
<td>Distortions resulted from using various protocols for (a) Using video profiles I with channel capacity set to 1 Mbps; (b) Using video profiles II with channel capacity set to 2 Mbps.</td>
<td>69</td>
</tr>
<tr>
<td>3.1</td>
<td>A simple subnetwork ad hoc model when host i intends to send the packets to its neighbor j. We need to take into account hosts i, j, and c for successfully transmitting packets from host i to host j.</td>
<td>81</td>
</tr>
<tr>
<td>3.2</td>
<td>Pseudocode for a variable elimination technique.</td>
<td>84</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.3</td>
<td>Example for a variable elimination technique.</td>
<td>86</td>
</tr>
<tr>
<td>3.4</td>
<td>Example for a linear wireless ad hoc network. (a) Network with one</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>flow; (b) Network with two flows; (c) Network with ten flows.</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>The outgoing rates from each sender based on a network topology as</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>shown in Fig. 3.4(a). (a) Transmission probabilities; (b) Outgoing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rates</td>
<td></td>
</tr>
<tr>
<td>3.6</td>
<td>The outgoing rates from each sender based on a network topology as</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>shown in Fig. 3.4(b). (a) Transmission probabilities; (b) Outgoing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rates</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Illustrative example of NC technique.</td>
<td>97</td>
</tr>
<tr>
<td>4.2</td>
<td>An example of a pattern of lost packets at MSs $z_1$ and $z_2$.</td>
<td>101</td>
</tr>
<tr>
<td>4.3</td>
<td>NC example.</td>
<td>103</td>
</tr>
<tr>
<td>4.4</td>
<td>Retransmission techniques.</td>
<td>104</td>
</tr>
<tr>
<td>4.5</td>
<td>Example of MS grouping.</td>
<td>107</td>
</tr>
<tr>
<td>4.6</td>
<td>Broadcast transmission/retransmission schemes.</td>
<td>117</td>
</tr>
<tr>
<td>4.7</td>
<td>Simulated network topology scenarios with different MS distributions.</td>
<td>118</td>
</tr>
<tr>
<td>4.8</td>
<td>Simulation results for network topology scenario of Fig. 4.7(a) with</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>120 uniformly distributed MSs.</td>
<td></td>
</tr>
<tr>
<td>4.9</td>
<td>Simulation results for network topology scenario of Fig. 4.7(b) with</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>60 MSs formed into 3-clusters.</td>
<td></td>
</tr>
<tr>
<td>4.10</td>
<td>Simulation results for network topology scenario of Fig. 4.7(a) when</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>varying the number of MSs.</td>
<td></td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
</tr>
<tr>
<td>2.2</td>
</tr>
<tr>
<td>2.3</td>
</tr>
<tr>
<td>2.4</td>
</tr>
<tr>
<td>2.5</td>
</tr>
<tr>
<td>2.6</td>
</tr>
<tr>
<td>2.7</td>
</tr>
<tr>
<td>3.1</td>
</tr>
<tr>
<td>Figure</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>A.1</td>
</tr>
</tbody>
</table>
Chapter 1 – Introduction

It is hard to imagine a world without cell phones, laptops, GPS, and the like, but this was the reality of couple decades ago. Advances in wireless technology have lead to the recent proliferation of wireless networks, that in turn, revolutionized the way people work and play by providing reliable communications and data accesses from anywhere at anytime. Numerous applications, ranging from web browsing and Short Message Service (SMS) to video conferencing and streaming are now possible on many smart phones and laptops. From the stock prices to the latest news in the world, every bit of information is now readily and conveniently accessed at the tip of our fingers. All of these are made possible by wireless networks, the backbone that is responsible for all wireless communications.

Prominently, the IEEE 802.11 standard [1], a wireless Medium Access Control (MAC) protocol, popularly known as Wi-Fi, has become an integral part of both private home and enterprise networks. Since Wi-Fi networks are designed for indoor settings, they have limited communication range, typically 32 meters for indoors and 95 meters for outdoors. Another emerging wireless standard designed for long-ranged communication is the IEEE 802.16 [2], also known as the Worldwide Interoperability for Microwave Access (WiMAX) standard. The WiMAX communication range varies from 1 kilometer to 50 kilometers, with longer communication distances accompanied by lower transmission rates. Its aim is to provide wireless...
broadband access as an alternative to the existing wired access networks such as cable and Digital Subscriber Loop (DSL). A primary competitor of WiMAX is the emerging wireless standard LTE, abbreviated for Long Term Evolution which is set to launch in 2011, to replace the current 3G cellular networks of Verizon and AT&T. At the other extreme, the IEEE 802.15.1 standard or Bluetooth [3] is designed for short-ranged communications (within a few feet) that typically take place among wireless devices, e.g., cellular phones and external storage devices.

The challenges in transmitting data reliably over a wireless network are numerous. Each aforementioned wireless network has many unique challenges of its own. Therefore, it is not the aim of the dissertation to address many of the fundamental challenges in wireless communication. Rather, the dissertation is an investigation of some novel techniques for efficient data transmission in specific contexts and settings. In particular, the focus of the dissertation is on the joint design of video streaming applications and MAC protocols for Wi-Fi and WiMAX networks to provide a probabilistic guarantee of some pre-specified level of performance or Quality of Service (QoS) such as bandwidth and delay of a flow.

Instead of providing the details of the backgrounds and motivations for our contributions in the Introduction, we refer the readers the beginning of each chapter of the dissertation. That way, it is more focused, and thus makes it easier for the readers to contrast our contributions with the existing works. Having said, in what follows, we will also provide a brief overview of the wireless networks, its challenges, current solution approaches, and an outline of our contributions.
1.1 Overview of Wireless Networks

Broadly defined, a modern data communication network is a stratum of interconnecting nodes on which, the data can be sent from one node to another node. The Internet is the largest data communication network connecting hundreds of millions computers around the world. It is a wired network for the most part since data is transmitted in a physical wire, e.g., coaxial cable, twisted-copper pair, or fiber optic cable. In contrast, Wi-Fi and cellular networks are examples of wireless networks in which data transmission takes place in the form of electromagnetic waves. Two fundamental characteristics of a data communication network are bandwidth and delay. Bandwidth is defined as the amount of data that can be sent between two points in the network per unit time, while delay between two points in the network is the elapsed time for a data packet to travel from one point to another. A typical wired network has higher bandwidth than that of a wireless network. This is due to the inherent differences in the way data is transmitted in a wire and in a wireless medium. To increase bandwidth of between two points in a wired network, one can lay more cables between these two points. Thus, data can be sent in parallel, leading to a larger amount of data per unit time, assuming that these cables are sufficiently insulated so that cross-talk does not happen. On the other hand, in a wireless network, every communicating device that uses the same frequency band for communications, shares the same medium. This implies that, of all the devices in the proximity of the wireless range, only one pair of devices can be in communication at a time. Otherwise, signals from different senders may
interfere with each other at a receiver. This is analogous to the scenario of many people talking at the same time in a room, resulting in noise that no one can make out the message of any other. Arguably, proper and dynamic spectrum assignment for the wireless devices can alleviate the problem, but this approach is not only expensive, but difficult. Ultimately, the total wireless spectrum is rather small compared to what offered by the technologies for wired transmissions. Another major challenge in wireless transmission is fading, a phenomenon due to varying delay and amplitude attenuation of the arrival signal. The variations are caused by electromagnetic wave reflections and refractions, traveling along multiple paths of different lengths. The situation become worse with mobility that changes the patterns of arrival signals dynamically with time. Without proper considerations, fading will incur significant errors during transmissions, and thus retransmissions are needed. All of these factors directly reduce the bandwidth of a wireless network.

Due to these challenges, many wireless transmission techniques have been developed over the years. A class of popular techniques to mitigate the effect of fading is the use of forward error correcting codes. The core idea is to transmit the original data plus some redundancy in such a way that the receiver can recover the original data if the amount of erroneous data does not exceed a certain threshold. The error correcting techniques however do not fundamentally change the capacity of a wireless channel. Rather, the good error correcting techniques would allow the transmission rate to be near the channel capacity. Another approach to increase the channel capacity is to use multiple transmit and receive antennas or MIMO, short for Multiple-Input Multiple-Output. With the MIMO framework, it is possi-
ble to perform beamforming, i.e., to direct most of the transmission energy toward a certain receiver, and thus avoiding the interference to others while increasing the signal strength at the intended receiver. Yet, another approach is to use Network Coding (NC) approach to be discuss shortly.

The MIMO framework is not at all sufficient to mitigate the interference since the receivers can be co-located or in proximity of each other. To minimize the interference, a variety of wireless MAC protocols have been proposed to arbitrate the transmission of wireless devices. In what follows, we will briefly discuss the beamforming, network coding techniques and the MAC protocols as they are fundamental to the contributions of the dissertation. More details on these can be found in subsequent chapters.

1.2 Medium Access Control (MAC) Protocols

Due to the efficiency reason, the MAC protocols of different wireless networks vary significantly. For Wi-Fi networks, the MAC protocol is designed based on the concept of randomized access that enables the wireless devices to operate with minimal centralized coordination, and thus easy to deploy. If a collision occurs, i.e., two devices transmitting at the same time then the data will be retransmitted at some random time later. Specific details on the operation of the MAC protocol will be discussed at the beginning of each relevant chapter. This premise works because the cost of retransmissions in Wi-Fi networks is inexpensive relatively to that of a wireless long-ranged network such as cellular network. In a cellular
network, the transmission time slots for each user is centrally coordinated to ensure no interference. Such a centralized scheduler will be discussed shortly in Section 1.3. The first part of the dissertation will focus on the design and analysis of novel MAC protocols based on randomized access principle, and how it can be used to benefit video applications. In the second part of the dissertation, we describe a joint optimization of network coding and beamforming techniques to realize a centralized packet scheduler in a wireless long-ranged network such as WiMAX.

1.3 SDMA and Beamforming Technique

Improving the performance of wireless communication systems can be accomplished by using MIMO. The concept of MIMO is to use multiple antennas to transmit and/or receive signals. Assume the Base Station (BS) and the Mobile Station (MS) are equipped with $M$ and $N$ antennas, respectively. In this case, the channel capacity of a MIMO system increases linearly with $\min\{M, N\}$. Because of size and cost constraints at the MS, the performance is limited by $N$ [4],[5].

When MSs are located spatially apart, the BS may enable simultaneously transmissions to multiple MSs over downlink channels by using a technique called Spatial Division Multiple Access (SDMA) to achieve higher throughput [6]. Once the BS know which group of MSs that the BS intends to transmit the signal to, we can use a technique called Beamforming (BF) at the BS to control the radiation of transmitted signals into a desired direction [7],[8]. That means the BS can generate multiple signal beams to provide the services to multiple MSs at the same time.
The number of signal beams can be generated at once at the BS is at most $M$ where $M$ is the number of antennas at the BS. There are a number of BF techniques. The simple one, but efficient, is to use an antenna array which contains multiple antenna elements at the BS to physically combine signals from those antenna elements in such a way to generate a desired radiation pattern. This technique is called Linear Phased Array BF. Assume the BS know Channel State Information (CSI) of all MSs, controlling the shape and direction of transmitted signals can be done by mathematically multiplying the signal beam from antenna elements with a BF weight vector. Given a limited power constraint at the BS, the interference among transmitted beams reduces the reception of signals at intended MSs in terms of Signal-to-Interference plus Noise Ratio (SINR). By taking into account both SDMA and BF techniques, the optimal solution is to provide the service to a set of MSs whose SINR is maximized. The overview of scheduling algorithms to select a set of MSs to be serviced will be discussed in next paragraph.

**Scheduling Algorithm for SDMA-based Beamforming Techniques**

Providing services to multiple MSs on wireless downlink channels is difficult especially when the number of MSs is greater than the number of BF signal beams. In particular, the challenge is how to find a set of MSs to be serviced in each time slot while minimizing interference incurred by different active BF signal beams in order to maximize the overall throughputs. An exhaustive search can provide the best solution but it can be very costly and infeasible for a large number of MSs. There are a number of works on scheduling algorithms. For example, Shad et al. [9] proposed an algorithm for selecting the MSs for each time slot over SDMA/TDMA
networks by considering the largest SINR margin of scheduled MSs with respect to some SINR thresholds. Ravindran et al. [10] studied the performance of zero-forcing BF techniques under limited channel feedback in line-of-sight channels. Their results show that a zero-forcing BF technique performs well especially when the system has reliable CSI from MSs. Furthermore, Yoo and Goldsmith [11] considered the orthogonality of CSIs among the MSs to schedule the MSs while minimizing the overall interference. In this dissertation, we propose two scheduling algorithms. The former is based on the idea in [11]. The latter is based on the use of a given codebook (which contains a list of sets of considered BF beam signals) with proposed algorithms. The detailed explanation will be described in Section 4.

1.4 Quality of Service

Improving bandwidth with beamforming and network coding, and mitigating interference alone are not sufficient. When there are multiple users in a wireless networks, resources must be divided among them. If the number of users is large enough, the bandwidth per user will be smaller than a specified value, resulting in detrimental effect for certain applications. For example, if a video streaming application requires a constant bandwidth of 1 Mbps, but the wireless network only supports an average of 800kbps, the video playback will suffer many interruptions. Therefore, it is necessary to employ admission control mechanism that limits the number of users or flows in the network any one point in time.
Admission Control

Admission control is a mechanism to prevent a new user (or a new flow) from joining the network to maintain the quality of the existing flows above some threshold [12],[13]. The decision to accept or reject a new flow for a wireless link is hard compared to that of wireline link. For example, in a wireline link, the admission control technique may simply keep track of total used bandwidth. The available bandwidth in this case will be the difference between the used bandwidth and the link capacity. Thus, a new flow will be admitted only if its requested bandwidth is less than the available bandwidth. Otherwise, it will be rejected.

In a wireless link, when using a default mode, the AP which acts as a centralized control is able to precisely regulate the flows. Thus, it is possible to determine the available bandwidth. Therefore, given a requested throughput of a new flow and computed available bandwidth, the AP can precisely make a decision to accept or reject a new flow. On the other hand, the IEEE 802.11 is based on a random access approach for which one can show that the collision increases nonlinearly with the number of competing flows [14]. When adding a new flows into an existing system, this will reduce the throughputs of all existing users while increasing collisions. As a result, it is non-trivial for the existing IEEE 802.11 MAC protocol to make a decision whether or not to accept or reject a new flow while guaranteeing the QoS of all the flows. Our contribution will be a novel MAC protocol that enables each user to independently decide to inject or not to inject a new flow into the network in order to maintain the QoS of all the existing flows.
1.5 Thesis Contributions and Organization

The dissertation is organized into five chapters. Each chapter is self-contained with sufficient background and related work. In Chapter 2, we present an approach to enhancing the quality of video streaming applications in wireless home networks via a joint optimization of video layer-allocation technique, admission control algorithm, and MAC protocol. Using an Aloha-like MAC protocol, we propose a novel admission control framework, which can be viewed as an optimization problem that maximizes the average quality of admitted videos, given a specified minimum video quality for each flow. We present some hardness results for the optimization problem under various conditions, and propose some heuristic algorithms for finding a good solution. In particular, we show that a simple greedy layer-allocation algorithm can perform reasonably well, although it is typically not optimal. Consequently, we present a more expensive heuristic algorithm that guarantees to approximate the optimal solution within a constant factor. Simulation results demonstrate that our proposed framework can improve the video quality up to 26% as compared to those of the existing approaches. In Chapter 3, we extend the admission control framework to include multiple linear wireless networks. We propose a simple Aloha-like MAC protocol that enables each flow to maintain its requested bandwidth, and thus is suitable for multimedia traffic. In addition, we propose a bandwidth feasibility algorithm based on the Variable Elimination (VE) technique. The bandwidth feasibility algorithm determines whether or not a given network can support a set of flows of certain bit rates. Simulations indicate
that our solution can precisely control the bit rates over all hosts while providing the throughput guarantees. In Chapter 4, we develop a framework that exploits network coding and MIMO techniques, jointly together, to improve throughput of downlink broadcast channels. Specifically, we consider a BS equipped with multiple transmit antennas that serves multiple MSs simultaneously by generating multiple signal beams. Given the large number of MSs and the small number of transmit antennas, the BS must decide, at any transmission opportunity, which group of MSs it should transmit packets to, in order to maximize the overall throughput.

We propose two algorithms for grouping MSs that take advantage of network coding and the orthogonality of user channels to improve the overall throughput. Our results indicate that the proposed techniques increase the achievable throughput significantly, especially in highly lossy environments. Finally, the conclusions and future work are discussed in Chapter 5.
Limited bandwidth and high packet loss rate pose a serious challenge for video streaming applications over wireless networks. Even when packet loss is not present, the bandwidth fluctuation as a result of an arbitrary number of active flows in an IEEE 802.11 network, can significantly degrade the video quality. This paper aims to enhance the quality of video streaming applications in wireless home networks via a joint optimization of video layer-allocation technique, admission control algorithm, and Medium Access Control (MAC) protocol. Using an Aloha-like MAC protocol, we propose a novel admission control framework, which can be viewed as an optimization problem that maximizes the average quality of admitted videos, given a specified minimum video quality for each flow. We present some hardness results for the optimization problem under various conditions, and propose some heuristic algorithms for finding a good solution. In particular, we show that a simple greedy layer-allocation algorithm can perform reasonably well, although it is typically not optimal. Consequently, we present a more expensive heuristic algorithm that guarantees to approximate the optimal solution within a constant factor. Simulation results demonstrate that our proposed framework can improve the video quality up to 26% as compared to those of the existing approaches.
2.1 Introduction

Recent years have witnessed an explosive growth in multimedia wireless applications such as video streaming and conferencing [15]. One of the reasons for this tremendous growth is the wide deployment of the IEEE 802.11 wireless LANs (WLANs) in both private home and enterprise networks. Despite of these seemingly successes, many fundamental problems of transmitting multimedia data over wireless networks remain relatively unsolved. One of the challenges is how to efficiently guarantee a specified bandwidth for a video flow in a wireless network. The popular WLAN, particularly Distributed Coordination Function (DCF) in typical IEEE 802.11 [1] which operates under a contention-based channel access mechanism, does not provide a mechanism to guarantee minimum bandwidth for multiple concurrent flows. As a result, a video application may experience significant quality degradation due to free admission of an arbitrarily large number of flows. Nevertheless, Point Coordination Function (PCF) in typical IEEE 802.11 and HCF Controlled Channel Access (HCCA) in IEEE 802.11e [16] are able to provide a polled access mechanism to guarantee the minimum bandwidth. However, to take advantage of PCF and HCCA mechanisms, a scheduler and a queuing mechanism at the AP are needed to control to regulate the polling frequency in HCCA and PCF to provide flows with the requested throughputs. That said, this paper considers a contention-based approach to admission control, similar to the work of Banchs et al. [12] in which, the parameters of the IEEE 802.11e in the contention-based mode are set appropriately to enable flows to achieve their
requested throughputs or reduce the delay.

Admission control prevents a new flow from joining the network in order to maintain a reasonable quality of the existing flows. The decision to admit or reject a new flow that requests to enter a wireline link is arguably easier to make, compared to that of a wireless link. A simple admission control algorithm for a wireline link can keep track of the total used bandwidth. The available bandwidth is then equal to the difference between the link capacity and used bandwidth. A new flow is admitted if its requested bandwidth is smaller than the available bandwidth of the link by some threshold, otherwise it is rejected. Theoretically, the same algorithm can be applied to a wireless link if a Time Division Multiple Access (TDMA) scheme is used to allocate bandwidth for each flow. Using a TDMA scheme, each flow is assigned a set of exclusive time slots for transmitting its data, thus eliminating the multi-user interference associated with a wireless link. As a result, the admission control algorithm can determine its available bandwidth precisely and make the decision to admit or reject a new flow accordingly.

However, such a protocol may require a centralized scheduling algorithm, which may not be feasible in a distributed environment. Therefore, existing Medium Access Control (MAC) protocols such as the IEEE 802.11, employs a random access approach that allows the flows to compete for shared channel efficiently. That is, IEEE 802.11 protocol enables the flows to achieve high throughputs, while minimizing their collisions. Thus, characterizing the wasted bandwidth from collisions is specific to a MAC protocol.

The problem of MAC protocols such as the IEEE 802.11 is the multi-user in-
terference, i.e., the collisions between the new flow and the existing flows, which reduce all the flow’s throughputs. The number of these collisions increases non-linearly with the number of competing flows, making it harder for an admission control algorithm to determine the resulted throughputs of all the flows in order to make the right decision [14]. In particular, for a simple single-hop wireless network, to decide whether or not to admit a new flow, the admission control algorithm must ensure that the available bandwidth is at least $K + H$ kbps, where $K$ is the total requested bandwidth including that of the new flow, and $H$ is the incurred overhead from the collisions. While $K$ is given to the algorithm, determining $H$ is non-trivial when using a typical MAC protocol. Computing $H$ is even more difficult in a multi-hop wireless network.

Even when an algorithm can determine precisely the collision bandwidth, it is not always beneficial to employ the traditional admission control framework in which, the decision to admit a new flow is solely based on the bandwidth and delay requirements of all the flows. Instead, with the advance in video coding techniques, we argue that the criterion for flow admission should be the visual quality of the video streams. That is, the inputs to the admission control algorithm are the minimum visual quality of the video streams, not their bandwidth and delay requirements. The former approach assumes that each video is coded at a certain bit rate, thus any lesser rate provided by the network, is unacceptable since the video playback will be interrupted frequently. On the other hand, with scalable video coding techniques, the video is encoded in a layered hierarchy. Thus, a video can be transmitted at different bit rates (or number of video layers), albeit at
different visual qualities. The advantage of this approach is that a larger number
of flows can be allowed to enter a network as long as the video quality of each
flow does not fall below a specified minimum threshold. The objective is then to
maximize the average quality of all the admitted videos, given a specified minimum
video quality for each stream, and the current available bandwidth.

That said, our paper aims to enhance the quality of video streaming applica-
tions in wireless home networks via a joint optimization of video layer-allocation
technique, admission control algorithm, and MAC protocol. While it is possible to
extend our framework to multi-hop wireless ad-hoc environment, for clarity, our
discussion is limited to a one-hop wireless network, e.g., the network of all the
wireless hosts (devices) within a home or a small building such that every host
can hear the transmissions of all other hosts. Using an Aloha-like MAC proto-
col [17], we present a novel admission control framework, which can be viewed as
an optimization problem that maximizes the average quality of admitted videos,
given a specified minimum video quality for each flow. In particular, using scalable
video streams, our framework allows more flows to enter the network, as long as
the video quality of each flow does not fall below a specified minimum threshold.
We then present some hardness results for the optimization problem under various
conditions, and propose two heuristic algorithms for obtaining a good solution. We
show that a simple greedy layer-allocation algorithm can perform reasonable well,
although it is typically not optimal. Consequently, we present a more expensive
heuristic algorithm that guarantees to approximate the optimal solution within a
constant factor.
The outline of paper is as follows. We first discuss a few related works on admission control for wireless networks and scalable video coding in Section 2.2. In Section 2.3, we describe a MAC protocol to be used in conjunction with the admission control algorithm. We then formulate the admission control framework as an optimization problem in Section 2.4. In Section 2.5, we provide some hardness results for the optimization problem, and corresponding heuristic algorithms for obtaining good solutions. Simulation results will be given in Section 2.6. We then summarize our contributions and conclude our paper with a few remarks in Section 2.7.

2.2 Related Work

Providing QoS for flows on the Internet is extremely difficult, if not impossible, due to its original design to scale with large networks. The current design places no limit the number of flows entering the network, or attempt to regulate the bandwidth of individual flows. As a result, bandwidth of multimedia applications over the Internet often cannot be guaranteed. To that end, many scalable coding techniques have been proposed for video transmission over the Internet. Scalable video coding techniques are employed to compress a video bit stream in a layered hierarchy consisting of a base layer and several enhancement layers [18]. The base layer contributes the most to the visual quality of a video, while the enhancement layers provide successive quality refinements. As such, using a scalable video bit stream, the sender is able to adapt the video bit rate to the current available
network bandwidth by sending the base layer and an appropriate number of enhancement layers [19],[20],[21],[22],[23],[24],[25]. The receiver is then able to view the video at a certain visual quality, depending on network conditions.

We note that scalable video coding techniques can mitigate the insufficient bandwidth problem, but the fundamental issue is the lack of bandwidth to accommodate all the flows. Thus, admission control must be used. While it is difficult to implement admission control on a large and heterogeneous network, e.g., the Internet, it is possible to implement some form of control or regulation in small networks, e.g., WLAN. Consequently, there have been many researches on providing some form of QoS for media traffic in WLANs [26],[27],[28],[29],[30],[31],[32]. Recently, the introduction of the new medium access control protocol of the IEEE 802.11e called the Hybrid Coordination Function (HCF), van der Schaar et al. [33] proposed an HCF Controlled Channel Access (HCCA) - based admission control for video streaming applications that can admit a larger number of stations simultaneously.

Many other existing admission control algorithms for WLANs have also been proposed. Gao et al. [34] provided an admission control by using a physical rate based scheme in IEEE 802.11e. They use the long-term average physical rates to compute the reservation of the channel for some amount of time called the Transmission Opportunity (TXOP) for each station then distribute TXOP’s to everyone. Their framework provides some certain level of admission control. Xiao and Li [35] used the measurements to provide flow protection (isolation) in the IEEE 802.11e network. Their algorithm is simple, yet effective. The algorithm
requires the Access Point (AP) to broadcast the necessary information to other wireless stations. In particular, the AP announces the budget in terms of the remaining transmission time for each traffic class (there are 4 traffic classes in the IEEE 802.11e) through the beacon frames. When the time budget for a class is depleted, the new streams of this class will not be admitted. Xiao and Li’s work set a fixed limit on the transmission time for the entire session, resulting in low bandwidth utilization when not every traffic class approaches its limit. Recently, Bai et al. [36] improved the bandwidth utilization of Xiao and Li’s work by dynamically changing the transmission time of each class based on the current traffic condition. There are also other admission control schemes implemented at different layers of the network stack. For example, Barry et al. [37] proposed to monitor the channel using virtual MAC frames and estimate the local service level by measuring virtual frames. Shah et al. [38] proposed an application layer admission control based on MAC layer measurement using data packets. Valae et al. [39] proposed a service curve based admission procedure using probe packets. Pong and Moors [13] proposed admission control strategy for QoS of flows in IEEE 802.11 by adjusting the contention windows size and the transmission opportunity. All these admission control schemes do not take quality of the traffic, particularly video quality in our framework, into consideration directly. On the other hand, we advocate a direct cross-layer optimization of video quality, admission control algorithm, and MAC protocol, simultaneously. Most similar to our work is that of Banchs et al. [12]. Since we will be using this scheme for performance comparisons, we delay the discussion until Section 2.6.
2.3 MAC Protocol

As discussed previously, the amount of wasted bandwidth from collisions in a wireless network is different when using different MAC protocol. In this section, we describe an Aloha-like MAC protocol [17] to be used in the proposed admission control framework that aims to maximize the average quality of admitted videos, given a specified minimum video quality for each flow. In order to contrast the advantages of the new MAC protocol, we first briefly describe the existing IEEE 802.11e protocols in the contention-based access mode.

2.3.1 Contention-Based Access Mechanism

The contention-based channel access scheme in existing IEEE 802.11e protocol called Enhanced Distributed Channel Access (EDCA), which defines a set of QoS enhancements for WLAN applications through modifications to the MAC layer. To access the channel, a host first senses the channel. If the channel is idle for more than the Arbitration Interframe Space (AIFS) time, it starts sending the data. Otherwise, it sets a backoff timer for a random number of time slots between $[0, CW_{\text{min}}]$ where $CW_{\text{min}}$ is the minimum contention window size. The backoff timer is decremented by one for each idle time slot after the AIFS time, and halts decrementing when a transmission is detected. The decrementing resumes when the channel is sensed idle again for an AIFS time. A host can begin transmission on the channel as soon as its backoff timer reaches zero. If a collision occurs, i.e., no acknowledgment packet is received after a short period of time, the backoff
The timer is chosen randomly between \([0, (CW_{min} + 1)2^i - 1]\) where \(i\) is the number of retransmission attempts. In effect, the contention window size is doubled for each retransmission in order to reduce the traffic in a heavily loaded network. Every time a host obtains the channel successfully, it can reserve the channel for some amount of time (TXOP). Unlike the IEEE 802.11b, IEEE 802.11e can tune the transmission parameters (i.e., \(CW_{min}, CW_{max}, TXOP, AIFS\)) to provide QoS support for certain applications. The EDCA is able to operate either in ad hoc or infrastructure modes.

The advantage of the existing IEEE 802.11 protocols is that it is bandwidth efficient. That is, based on the current traffic condition, each host adjusts its rate to achieve high throughput while minimizing the number of collisions. On the other hand, the rate of a flow cannot be controlled precisely unless we use PCF or HCCA. Often, this is problematic for video applications. In particular, without using PCF or HCCA, the fluctuation in achievable throughput would likely occur in the existing IEEE 802.11 protocols due to their best effort behaviors. However, PCF and HCCA require to have such a perfect scheduler in order to guarantee the achievable throughputs of all the flows. Thus, we argue for a different MAC protocol which, when used, would produce a stable throughput for a flow. Furthermore, it is preferable to implement the new MAC protocol with minimal hardware modification to the existing IEEE 802.11 devices. Indeed, this is possible.
2.3.2 Proposed MAC Protocol

In the new MAC protocol, the contention window size is not doubled after every unsuccessful retransmission attempt. Instead, depending on the rate requested by a host, it is assigned a fixed value. All other operations are exactly identical to those of the IEEE 802.11 protocol. We argue that when a proper admission control is employed, eliminating the doubling of $CW$ in the IEEE 802.11 protocol, helps to increase the bandwidth efficiency since the rate of each host is not reduced unnecessarily. We note that the existing works with EDCA cannot precisely control the rate due to best effort behavior. Furthermore, the transmission parameters (i.e., $CW_{\text{min}}$, $CW_{\text{max}}$, TXOP, AIFS) in EDCA are not designed to achieve the exact rate (on average). This would result in unnecessarily increasing in either the transmission rate comparing to requested rate or unexpected collision. However, our proposed protocol is able to solve such problems in contention based IEEE 802.11e.

Based on the above discussion, it is crucial for an admission control algorithm to determine whether or not there exists a set of $CW$’s for each host that satisfies their requested rates without doubling $CW$’s. To answer this question, we now proceed with an analysis of the new MAC protocol.

We assume the use of reservation packets, i.e., Request-To-Send/Clear-To-Send (RTS/CTS) packets. RTS/CTS packets are employed to reduce the collision traffic as well as eliminating the hidden terminal problem [40]. The main idea is to send small packets to reserve the channel for the actual data transmission. By doing so,
collisions only occur with the small packets, hence reducing the amount of wasted bandwidth. Since we assume that all the hosts can hear each other’s transmissions, we do not have the hidden terminal problem. Our use of RTS/CTS is simply to reduce the collision bandwidth.

Our analysis is based on time-slotted, reservation based protocols similar to the Aloha protocol, where the time taken to make a reservation is a geometrically distributed random variable with parameter $p$. The significant difference between our protocol and the Aloha protocol is that all the hosts in our network are assumed to be able to hear each other transmissions. Therefore, a host will not attempt to transmit if it determines that the channel is busy, i.e., some host is sending. Thus, a host will attempt to send an RTS packet with probability $p$ only if it determines that the channel is idle.

Assume the host transmits the packets with some probability $p$. To translate the transmission probability $p$ back to the contention window size used in IEEE 802.11 protocol, $CW$ can be set to $2/p$. We note that this is only an approximation since $CW$ in the IEEE 802.11 protocol is not reset at every time slot. To simplify the analysis, we further assume that every host can start at most one flow at any point in time. A straightforward generalization to support multiple flows per host is to consider all the flows from one host as one single large flow with the transmission probability $p$. Whenever a host successfully obtains the channel, it selects a packet from one of its flows to send. The probability of a packet selected from a particular flow then equals to the ratio of that flow’s throughput to the total throughput of all the flows on the same host. This approach would result in
the correct average required throughputs for all the flows.

For a network with \( N \) flows, our objective is to determine whether or not there exists a set of \( p_1, p_2, \ldots, p_N \) and for each flow such that all the flows achieve their specified throughputs \( R_1, R_2, \ldots, R_N \), taking into account of collisions. Since the rates \( R_i \)'s depend on the percentages of successful slots, we first characterize the percentages of collided, successful, and idle slots, given \( p_i \)'s for each flow \( i \). To that end, let us denote

- \( I \): percentage of idle slots
- \( S_i \): percentage of successful RTS slots for a flow \( i \)
- \( C \): percentage of collided slots
- \( R'_i \): throughput of flow \( i \) as a fraction of the channel capacity.

Note that \( I + C + \sum_i S_i = 1 \). Suppose the transmission probability for a new flow is \( p \), then for \( C - type \) slots, in which collisions occur, the new traffic would have no impact on it. For \( S - type \) slots, with probability \( p \), it may cause a collision. For an \( I - type \) slots, with probability \( p \), it would become a \( S - type \) slot. Otherwise it stays the same. Using the above argument, we can calculate \( I, S, \) and \( C \) after the new flow starts. In particular, the new idle, collided, and successful probabilities
can be calculated using the current $I$, $C$, $S$, and $p$ as:

\[
S_{\text{new}} = S_{\text{current}}(1 - p) + I_{\text{current}}p \tag{2.1}
\]

\[
I_{\text{new}} = I_{\text{current}} - I_{\text{current}}p \tag{2.2}
\]

\[
C_{\text{new}} = 1 - I_{\text{new}} - S_{\text{new}}. \tag{2.3}
\]

Here, we denote $S = \sum_i S_i$. Similarly, we can calculate the successful probability $S_i$ as

\[
S_{i,\text{new}} = S_{i,\text{current}}(1 - p), \tag{2.4}
\]

for any existing flow $i$, and the successful probability for the new flow ($S_{N+1}$) as

\[
S_{N+1} = I_{\text{current}}p. \tag{2.5}
\]

Using the equations above, one can compute the $I$’s, $C$’s, and $S_i$’s for $N$ flows, given the transmission probabilities $p_1, p_2, ..., p_N$. In particular, the following algorithm can be used to compute the collision probability $C$, which will be used in the admission control algorithm later.

**Algorithm 1:** Computing $C$, given the transmission probabilities $p_i$’s

\[
C = \text{Compute}_C(p_1, p_2, ..., p_N, N)
\]

$I = 1$

$C = 0$

$S = 0$
for $i = 1$ to $N$ do
\begin{align*}
S &= S \times (1 - p_i) + I \times p_i \\
I &= I - I \times p_i \\
C &= 1 - I - S
\end{align*}
end for

return $C$

Algorithm 1 enables us to compute the successful probability precisely based on the given transmission probabilities $p_i$’s. On the other hand, one typically wants to determine the transmission probabilities $p_i$’s, given the requested rates $R'_i$’s from each flow $i$. Since the rate $R'_i$ is proportional to the corresponding successful probability $S_i$, we now show how to compute $p_i$’s based on $S_i$’s. We then show how to relate $S_i$’s to $R'_i$’s, completing our objective.

In principle, (2.1)-(2.5) enable us to write down a set of $N$ equations with $N$ unknown variables $p_1, p_2, \ldots, p_N$ in terms of the known variables $S_i$’s, and solve for $p_i$’s. Unfortunately, these equations are not linear, and therefore difficult to solve. We propose an algorithm to find the $p_i$’s given $S_i$’s based on the following observation: When a flow $i$ stops, $I$ will increase by $S_i$. If flows $i$ starts again with the same transmission probability $p_i$ as before, its successful probability remains $S_i$ as before. Hence, the following equations hold:

\begin{align*}
(I + S_i)p_i &= S_i \\
p_i &= \frac{S_i}{I + S_i}
\end{align*}

(2.6)
This is true because \( I + S_i \) is the probability of idle slots without flow \( i \). Hence, after the flow \( i \) starts, its successful probability is \( (I + S_i)p_i \) which should also equal precisely to \( S_i \), the successful probability before it stops. Thus, we have \( N \) such equations corresponding to \( N \) flows. We also have the constraint:

\[
I + C + S = 1 \tag{2.7}
\]

where \( C \) and \( S \) are the collision and successful probabilities for all the flows. We note that \( I \) is the same for every equation since it is the probability of idle slots when all flows are active. Now, we can solve for \( N + 1 \) unknowns, i.e. \( N \) for \( p_i \)’s and one for \( I \). Solving this set of \( N + 1 \) equations is simple since each equation is linear except (2.7). Equation (2.7) is non-linear in \( p_i \) because \( C \) and \( S \) are polynomials in \( p_i \) which are the results from (2.1)-(2.5). However, (2.7) will be used as a constraint. Since \( I \in [0,1] \), one can systematically try different values of \( I \) from large to small, i.e., 1 to 0. For each value of \( I \), we compute \( p_i \)'s according to (2.6). All the \( p_i \)'s are then input to Algorithm 1 to compute \( C \). We then test to see whether or not \( I + C + S \) approximately equals to 1. If so, we have an admissible set of solutions. If not, we increase \( I \) by a small value and repeat the procedure. If the algorithm cannot find such \( I \) for the entire range of \( I \in [0,1] \), then the solution does not exist. This indicates invalid \( S_i \)'s.

Typically, \( R'_i \)'s, not \( S_i \)'s, are given. Therefore, to use the procedure above, we first calculate the \( S_i \)'s in terms of \( R'_i \)'s. With minimal modification from IEEE 802.11e standard, our framework uses IEEE 802.11e frame formats and the timing
diagram as shown in Fig. 2.1. After the channel is idle for a period time equal to a Distributed Interframe Space (DIFS) time slots, instead of counting down $CW$ before beginning a new transmission, the host sends RTS with probability $p$ to reserve the shared channel. That means we set $\text{AIFS} = \text{DIFS}$ as same as the one in typical IEEE 802.11 standard [1]. Because all hosts can listen each other transmissions, the collision will occur only if there is more than one host initiating RTS’s at exactly the same time slot. Otherwise, the host successfully reserves the channel then that host can begin the transmission for $TXOP$ time slots without further collision. A host detects unsuccessful transmission of an RTS if none of the CTS arrives within DIFS time slots. Note that a host needs to wait for a short period of time called Short Interframe Space (SIFS) where $\text{SIFS} < \text{DIFS}$ before sending an ACK as shown in Fig. 2.1. To be fair among all the flows with the same traffic class [16], i.e. video streams, everyone uses the same $TXOP$ where $TXOP = \text{CTS} + \text{PHY}_{\text{hdr}} + \text{MAC}_{\text{hdr}} + \text{PAYLOAD} + \text{ACK} + 3\text{SIFS} + \text{DIFS}$.

Suppose after $T$ time slots where $T$ is large, we observe that there are $K_i$
successful transmissions of RTS and $K_i \times TXOP$ slots of data transmission for each flow $i$. Then by definition, we have:

$$S_i = \frac{K_i}{T - \sum_{i=1}^{N} K_i \times TXOP} = \frac{K_i \times TXOP / T}{TXOP \times (1 - \sum_{i=1}^{N} K_i \times TXOP / T)} = \frac{R'_i}{TXOP \times (1 - \sum_{i=1}^{N} R'_i)}$$

(2.8)

where $R'_i = K_i \times TXOP / T$ can be thought of as the host $i$’s requested bandwidth in terms of a fraction of the channel capacity and $N$ is the number of flows. If the channel capacity is $BW$, then the transmission rate $R_i$ can be computed such that $R_i = R'_i \times BW$. For example, if channel capacity ($BW$) is 54 Mbps, and host $i$ requests the rate ($R_i$) of 27 Mbps, then $R'_i = 0.5$.

Using (2.8), given the specified rates $R'_i$’s, one can compute the corresponding $S_i$’s, which are then used in the following algorithm to determine the transmission probabilities $p_i$’s, if there are such $p$’s.

**Algorithm 2:** Compute $p_i$’s given all $R'_i$’s

$[p_1, p_2, ..., p_N, success] = Compute_p(R'_1, R'_2, ..., R'_N, N)$

$\epsilon = 0.01$

$I' = 1$

{$I'$ is the percentage of idle slots}
search_step = 0.01
success = 0

for $i = 1$ to $N$ do
    $S_i = \frac{R'_i}{TXOP \times (1 - \sum_{i=1}^{N} R'_i)}$
end for

while $I' < 1$ do
    for $i = 1$ to $N$ do
        $p_i = \frac{S_i}{I' + S_i}$
    end for
    {run Algorithm 1 to compute collision probability $C$
    $C = Compute_C(p_1, p_2, ..., p_N, N)$
    $total = I' + C(RTS + DIFS) + \sum_{i=1}^{N} S_i(RTS)$
    {check for boundary condition smaller $\epsilon$ results in higher accuracy
    if ($abs(total - 1) < \epsilon$) then
        success = 1
        return $[p_1, p_2, ..., p_N, success]$
    end if
    $I' = I' - search_step$
end while
{fail to find $p$, success = 0
return $[0, 0, ..., 0, success]$

We note that for each unsuccessful RTS transmission, we waste the channel equal to
RTS+DIFS time slots. On the other hand, each successful RTS transmission uses
only RTS time slots. Furthermore, Algorithm 2 explicitly considers the percentage
of collided, successful, and idle slots with respect to RTS transmissions to reserve the channel. This results in \( \text{total} = I' + C(RTS + DIFS) + \sum_{i=1}^{N} S_i(RTS) \) where \( \text{total} \) is close to (or equal to) 1. We now describe our proposed admission control framework.

2.4 Admission Control Framework

2.4.1 Architecture

Due to a typical small size of a single-hop network, our admission control algorithm runs at the AP or an elected host. We assume that all hosts are collaborative. That is, each host obeys the admission protocol which operates as follows.

For simplicity, in this paper, we assume that there is no cross-traffic of any kinds except videos. In general, to accommodate other non-time sensitive traffic in the proposed framework, one can perhaps set the minimal throughput requirements for the traffic. Each host can send videos to and receive from the Internet, or they can send videos among each other. When a host wants to inject a new video stream into the network, it first requests to join the network by sending a message to the AP. For video streaming applications, the message may contain the rate distortion profile of the video, which specifies the different distortion amounts and the corresponding number of layers used. The message also contains the maximum allowable distortion for a video. Note that for live video applications, the rate-distortion profile is typically not known ahead of time, but can be estimated. That
said, the focus of this paper will be on streaming applications. Upon receiving the request, the AP (or some elected host) will run the admission control algorithm to produce a set transmission probabilities $p_i$’s for each flow $i$ that maximizes the average visual quality of all the flows, given the maximum distortion levels for each video and overall bandwidth constraint. If such transmission probabilities exist, the AP will broadcast the transmission probabilities $p_i$’s to all the hosts.

Upon receiving AP’s instructions, each host $i$ begins to transmit its packets with probability $p_i$ (or roughly setting its contention window to $2/p_i$) when it observes that the channel is idle. Each transmission probability $p_i$ corresponds to a particular rate (or number of layers). If there is no feasible set of transmission probabilities, the AP will inform the new flow that it cannot join the network at this moment.

2.4.2 Problem Formulation

We are now at the position to formulate a rate-distortion optimization problem for multiple layered video streams under bandwidth and distortion constraints. We note that the average throughput per unit time or transmission rate $R_i$ for a flow $i$ can be achieved by setting its transmission probability $p_i$. When there is enough bandwidth for everyone, $p_i$’s are set to large values so that all the layers of all the video streams would be sent. When there is not enough bandwidth, e.g. due to too many flows, the layers from certain videos are dropped resulting in the least average distortion over all the videos. For a simple scenario, we assume that there
is no packet loss during transmission. The transmission rate $R_i(l_i)$ for flow $i$ is proportional to the number of transmitted video layers $l_i$.

The optimization problem studied in this paper is to select the optimal number of video layers to transmit for each of $N$ hosts (or $N$ flows) while maximizing the overall video quality. Furthermore, the inclusion of the bandwidth overhead term $H = C + S$ due to channel contention access (collision and reservation bandwidth used for RTS/CTS packets) makes our optimization problem distinct from other optimization problems. In particular, the problem is specified by giving: for each host, a function $D_i(l_i)$, that gives the reduction in distortion when using $l_i$ layers at host $i$; a rate function $R_i(l_i)$, that gives the required bandwidth for transmitting $l_i$ layers from host $i$; an overhead function $H(l_1, ..., l_N)$, that gives the amount of bandwidth consumed by overhead (e.g., due to the channel contention) for a given assignment of layers to hosts; lower bounds on the reduction in distortion for each $i$ denoted by $Z_i$; and finally a bound on the total bandwidth $BW$. Given these quantities, the optimization problem is as follows:

$$\text{maximize} \quad \sum_{i=1}^{N} D_i(l_i)$$

$$\text{over} \quad l_i$$

$$\text{subject to} \quad D_i(l_i) \geq Z_i$$

$$\sum_{i=1}^{N} R_i(l_i) + H(l_1, ..., l_N) \leq BW$$

That is, we must find the optimal assignment of layers to each host that max-
imizes the reduction in total distortion subject to bandwidth and local minimum reduction in distortion constraints. In particular, there exists a solution iff we are able to compute a set of transmission probabilities \( p_i \)'s corresponding to an optimal assignment of layers for everyone. A necessary condition is that each flow \( i \) is required to maximize its total reduction in distortion \( D_i \) at least \( Z_i \). Nevertheless, the way to select the layer depends on what layer-selection strategies we use (e.g., greedy algorithm, exhaustive search). Note that propagation delay and processing delay can be negligible due to operating in a single-hop network. However, the delay variation or jitter, would likely affect the performance of the protocol. The detail analysis of throughput jitter is discussed in Section 2.6.3. Next, we will study the computational properties of the layer-selection problem and show that while in general the problem is computationally hard, under certain reasonable conditions, a simple greedy layer-allocation algorithm can be guaranteed to perform close to optimal.

### 2.5 Computational Complexity of Layer Optimization

In this section, we study the computational complexity of the layer allocation problem described above, showing both hardness results and conditions under which optimal and approximate solutions can be guaranteed in polynomial time. Our optimization problem is distinct from most other bandwidth optimization problems by its inclusion of the overhead term \( H \) in the bandwidth constraint. Thus, existing algorithms and complexity proofs do not directly apply to our problem.
Below we first consider the complexity of solving the problem optimally and then we consider efficient approximation algorithms.

2.5.1 Computing Optimal Solutions

Here we analyze the computational complexity of problem classes with the form given in the previous section. We begin by stating three assumptions about the optimization problem and consider the complexity under various subsets of these assumptions.

Assumption 1: Uniform rate increase per level

\[ R_i(l + 1) - R_i(l) = R_j(l' + 1) - R_j(l') \; \text{for any } i, j, l, \text{ and } l' \quad (2.10) \]

Assumption 2: Diminishing returns

\[ D_i(l + 1) - D_i(l) \leq D_i(l) - D_i(l - 1) \; \text{for any } i \text{ and } l \quad (2.11) \]

Assumption 3: Invariant overhead

\[ H(..., l_i + 1,...) = H(..., l_j + 1,...) \; \text{for any } l_1, ..., l_N \quad (2.12) \]

Below we will also refer to the property of additive overhead which means that \( H(l_1, ..., l_N) \) can be factored as a sum of the individual overhead function \( H_i \). That
is,

\[ H(l_1, ..., l_N) = \sum_{i=1}^{N} H_i(l_i) \]  \hspace{1cm} (2.13)

Intuitively the first assumption states that the amount by which the rate function increases is constant across all layers of all streams. The second assumption states that within a particular stream, higher layers may never reduce distortion more than the lower layers. Thus, it will never be the case that a stream must include many lower layers with low distortion reduction in order to get a big distortion reduction at a higher layer. The third assumption states that given a particular layer allocation across layers, incrementing any layer by one produces the same increase in the overhead function. This means that the overhead function is impartial to both the particular stream that is incremented and the current number of layers allocated to that stream.

Our first result is that given the above three assumptions (2.10)-(2.12), we can solve the optimization problem using an efficient greedy layer-allocation algorithm. The algorithm proceeds as follows:

1. For each stream \( i \), we initialize the layer count \( l_i \) to the smallest \( l_i \) such that \( D_i(l_i) \geq Z_i \). If for some \( i \) this is not possible, then return “no solution”.

2. If it is not possible to increment the layer count of any stream without violating the bandwidth constraints then terminate and return the current layer counts. In other words, it is not possible to find a feasible set of transmission probabilities for each host using the Algorithm 2 in Section 2.3.

3. Increment the layer count of stream \( i \) by 1, where stream \( i \) is the stream that
when incremented produces the greatest reduction in distortion without violating bandwidth constraints.

**Proposition 1** The greedy layer-allocation algorithm is optimal for any problem where Assumptions 1, 2, and 3 hold.

**Proof 1** We first introduce some notation. We will use an allocation vector $L = (l_1, \ldots, l_N)$ to specify the layer allocation $l_i$ to each host $i$ where $N$ is the total number of hosts. We will denote by $D(L)$ the reduction in distortion resulting from allocation vector $L$. A layer increment sequence is a sequence of host indices $(i_1, \ldots, i_k)$, indicating the order of layers to increment finally arriving at a final allocation vector where $k$ is the total increments in layers.

Note that with invariant overhead and uniform rate increase, each increment in the layer counts results in exactly the same increase in bandwidth. This means that all optimal layer allocations will satisfy $\sum_i l_i = k$ for some value $k$. That is, all optimal layer allocations will be a result of exactly $k$ increments to layer counts. Thus, finding the optimal layer allocation is equivalent to finding a length $k$ layer increment sequence that results in the best layer allocation starting from the null allocation.

Now consider any layer allocation $L$ and let $i^*$ be the index of the host that would be selected by the greedy algorithm starting from $L$ and let $\Delta^*$ be the reduction in distortion resulting from the greedy step. Now consider any layer increment sequence $(i_1, \ldots, i_v)$ starting at $L$ resulting in an allocation vector $L_v$. We say
that the sequence is an optimal $v$-step completion of $L$ if the value of $D(L_v)$ is the maximum possible when starting from $L$ and incrementing $v$ layers.

Our key claim is that there is always an optimal $v$-step completion to $L$ that includes an increment to $i^*$. Assume that this was not the case and that the above sequence was an optimal completion, implying that it does not contain an increment to $i^*$. We show that this leads to a contradiction. First, let $\Delta_j$ equal the reduction in distortion resulting after adding the $j$’th layer increment and note that $D(L_v)$ is equal to the sum of this sequence. By the diminishing returns assumption we have that $\Delta^* \geq \Delta_j$ for all $j$. This is true because the greedy algorithm selected the index $i^*$ with the largest decrease in distortion across all layers and thus any further decreases resulting from incrementing any layer must not be greater than that, otherwise this would violate diminishing returns. Given this fact consider the new layer increment sequence $(i^*, i_1, \ldots, i_{v-1})$ and let $L^*$ equal the result of applying this sequence starting at $L$. It can be easily verified that this is a legal sequence and that the corresponding sequence of reductions in distortion is equal to $(\Delta^*, \Delta_1, \ldots, \Delta_{v-1})$. Since $D(L^*)$ is simply the sum of this sequence and we know that $\Delta^* \geq \Delta_v$ this implies $D(L^*) \geq D(L_v)$. Thus, we have shown an optimal $k$-step completion that includes an increment to $i^*$, which gives a contradiction.

Using the above fact, it is straightforward to show by induction on the number of greedy steps $k$ that the greedy algorithm always maintains an optimal $k$-step completion of the null set, which completes the proof.

We now consider in Propositions 2-4, the complexity of this problem when each one of the above assumptions is lifted.
Proposition 2  The class of decision problems for which Assumptions 2 and 3 hold but Assumption 1 does not is NP-complete even if we restrict the overhead to be the constant zero function.

Proof 2  Our problem is clearly in NP, as it is possible to enumerate possible layer allocations and check them in polynomial time. Each layer- allocation certificate is polynomial-size implying the decision problem is in NP. To show NP-hardness we reduce from 0-1 knapsack problem. More formally, an instance of 0-1 knapsack is a 4-tuple as shown in (2.14),

\[ \langle \{v_1, ..., v_N\}, \{c_1, ..., c_N\}, V, C \rangle \]  \hspace{1cm} (2.14)

where \( v_i \) and \( c_i \), and give the value and cost of the \( i \)'th item, \( V \) is the value goal and \( C \) is the cost limit. We will form the following version of our problem as in (2.15),

\[ \langle \{1, ..., 1\}, \{D_1(l_1), ..., D_N(l_N)\}, \{R_1(l_1), ..., R_N(l_N)\}, H(l_1, ..., l_N), V, C \rangle \]  \hspace{1cm} (2.15)

where \( D_i(l_i) = v_i \), \( R_i(l_i) = c_i \), and \( H(l_1, ..., l_N) = 0 \) for all inputs. That is, we have only one layer to allocate. The reduction in distortion and the rate function for that layer are equal to \( v_i \) and \( c_i \) of the 0-1 knapsack problem, respectively. Note that this problem does not satisfy the constant rate increase since \( c_i \) can be different for each \( i \). However, it does satisfy Assumptions 2 and 3 trivially. It is straightforward to show that the answer to the layer-allocation problem given by
Proposition 3 The class of problems for which Assumptions 1 and 3 hold but Assumption 2 does not is NP-complete even if we restrict the overhead to be the constant zero function.

Proof 3 The problem is in NP for the same reasons as above. For the purposes of this problem we will only consider the 0-1 knapsack problem with integer values of $v_i$, $c_i$, $V$, and $C$. We can do this without loss of generality since we can always multiply all numbers by the appropriate power of 10. Given an instance of the 0-1 knapsack problem as specified in (2.14), our reduction constructs the following layer allocation problem,

$$\langle \{c_1, ..., c_N\}, \{D_1(l_1), ..., D_N(l_N)\}, \{R_1(l_1), ..., R_N(l_N)\}, H(l_1, ..., l_N), V, C \rangle$$ (2.16)

where $D_i(l_i) = 0$ for $l_i < c_i$, $D_i(c_i) = v_i$, $R_i(l_i) = l_i$, and $H(l_1, ..., l_N) = 0$. The intuition here is that we have many layers in each stream, but only the last layer actually reduces the distortion. This type of behavior violates Assumption 2. Each layer adds exactly one unit of bandwidth which satisfies Assumption 1 and the overhead is zero, which satisfies Assumption 3. Note also that the number of layers for flow $i$ is $c_i$. So in order to get a reduction in distortion of $v_i$, we must pay a bandwidth of $c_i$, which aligns with the 0-1 knapsack problem.

Given this reduction, it shows that there is a “yes” answer for the constructed layer allocation problem iff there is a “yes” answer for the 0-1 knapsack instance.
Finally, we show that Assumption 3 is also necessary in some sense. In particular, when it is lifted the problem becomes computationally hard even when restricted to the class of problems with additive overhead.

**Proposition 4** The class of problems for which Assumptions 1 and 2 hold but Assumption 3 does not is NP-complete even if we restrict to additive overhead.

**Proof 4** Again here we will only consider the integer knapsack problem. Given an instance of an integer 0-1 knapsack problem, we construct the following instance of the layer allocation problem,

$$\langle \{v_1, ..., v_N\}, \{D_1(l_1), ..., D_N(l_N)\}, \{R_1(l_1), ..., R_N(l_N)\}, H(l_1, ..., l_N), V, C \rangle$$  \hspace{1em} (2.17)

where \( D_i(l_i) = l_i \), \( R_i(l_i) = 0 \), and \( H(l_1, ..., l_N) = \sum_i H_i(l_i) \) where \( H_i(l_i) = c_i \) for \( l_i > 0 \) and \( H_i(0) = 0 \). So here we have the diminishing return property since for each layer we add a single unit of reduction in distortion. We have the constant bandwidth property trivially. But we do not have invariant overhead since when we move a layer \( l_i \) from 0 to 1 we get an increase in \( c_i \). Only the overhead function occupies the bandwidth since all the rates are equal to zero. Given this reduction it is easy to verify that there is a “yes” answer to the 0-1 knapsack instance iff there is a “yes” answer to the constructed layer allocation problem.

Together these complexity results show that if we remove any one of three assumptions, the problem becomes NP-hard and hence is not likely to be solved by an efficient algorithm, in particular the greedy algorithm. The results also show that
this is true even if we place strict restrictions on the form of the overhead function. Even if the overhead is additive the problem is hard as shown by Proposition 4.

In practice, it may often be possible to satisfy assumptions 1 and 2. Unfortunately, we can show that the overhead function arising in our protocol is not invariant as required by Assumption 3 and hence an efficient optimal solution is still unlikely.

**Proposition 5** The overhead of the proposed MAC protocol is not invariant.

**Proof 5** We will show that the overhead of proposed MAC protocol is not invariant by contradiction. Assuming that the bandwidth overhead $H = C + S$, bandwidth involved in reservation, is invariant, then:

$$H'(i) = H'(j) \quad (2.18)$$

where $H'(i)$ and $H'(j)$ denote the overhead resulted from adding $\delta$ bps (one layer) into flow $i$ and flow $j$, respectively. Since $I + C + S = I + H = 1$, we have

$$I'(i) = I'(j) \quad (2.19)$$

where $I'(i)$ and $I'(j)$ denote the idle slots resulted from adding $\delta$ bps into flow $i$ and flow $j$, respectively. In particular, adding $\delta$ bps into flow $i$ results in the increase
of $S_i$ by $\Delta$. We can represent $I$ in terms of $S_i$ as:

$$I = \prod_i (1 - p_i)$$

$$= \prod_i \left(1 - \frac{S_i}{I+S_i}\right)$$

$$= \prod_i \left(\frac{I}{I+S_i}\right)$$

(2.20)

That is,

$$\left(\frac{I'(i)}{I'(i) + (S_i + \Delta)}\right) \prod_{k \neq i} \left(\frac{I'(i)}{I'(i) + S_k}\right) = \left(\frac{I'(j)}{I'(j) + (S_j + \Delta)}\right) \prod_{k \neq j} \left(\frac{I'(j)}{I'(j) + S_k}\right)$$

(2.21)

Expand the product on both sides of (2.21). All product terms, except the one with either $S_i$ or $S_j$, will be canceled out, leading to:

$$(I'(i) + (S_i + \Delta))(I'(i) + S_j) = (I'(j) + S_i)(I'(j) + (S_j + \Delta))$$

(2.22)

Since $I'(i) = I'(j)$, (2.22) is true iff $S_i = S_j$. Since $S_i$ is directly proportional to $R_i$, this implies that in order to achieve $H'(i) = H'(j)$, the current sending rate of any two video stream must be equal to each other. This is not true in general, therefore the bandwidth overhead of the proposed MAC protocol is not invariant.
2.5.2 Computing Approximate Solutions

Of the three assumptions above, Assumption 2, diminishing returns, is the one that is most likely to be satisfied in application settings, since most coding schemes exhibit diminishing returns behavior. Assumption 1, uniform rate increase, will also often be satisfied, though it will rule out non-uniform rate coding schemes. Assumption 3, invariant overhead, as we have seen is violated by the our protocol and we are unaware of other protocols that satisfy the assumption. In this light, it is interesting to consider what can be said theoretically when only Assumption 2 is satisfied. We know from Propositions 2 and 3 that in general the problem is NP-hard to solve optimally with only Assumption 2, however, this does not rule out the existence of efficient approximation algorithms.

In fact, using a slight modified the greedy layer-allocation algorithm, we can obtain a solution guaranteed within a factor of \((1 - e^{-1}) \geq 0.63\) from the optimal solution. We now describe the first approximate algorithm.

**Double-Greedy Algorithm.** This algorithm is based on the recent result in [41]. We consider a modified version of the original greedy algorithm, where at each step instead of adding the video layer \(x\) that most increases in the reduction in distortion, we add the layer that achieves the largest ratio of the reduction in distortion to the increases in bandwidth, provided that adding \(x\) does not violate the total bandwidth constraint \(BW\). That is, for each iteration, we add the layer \(x\) with maximum ratio of the reduction in distortion to the increases in bandwidth, such that the total usage bandwidth is less than \(BW\). The iteration repeats un-
til no video layer can be added to the network without violating the bandwidth constraint. Thus, it can be shown that even this modified algorithm can yield arbitrarily poor results compared to the optimal solution. However, if one returns the best solution found by either the original greedy algorithm or the modified greedy algorithm then one can guarantee an approximation factor of \(0.5(1 - e^{-1})\). We will call this algorithm the *Double-Greedy algorithm*.

This shows that by increasing the computation time by a factor of two over the original greedy algorithm (i.e. we must now run both greedy and modified greedy) it is possible to achieve a non-trivial approximation bound.\(^1\)

**Triple-Greedy Algorithm.** It turns out that by increasing the computation further, but remaining polynomial time, it is possible to improve the approximation bound to \((1 - e^{-1})\) for the constrained optimization problem, which matches the result for the unconstrained problem, as proven in [41],[42],[43]. The new algorithm simply enumerates all triples of consecutive layers, denoted by \(T\), in each flow that do not violate the bandwidth constraint \(BW\). For each triple \(T\), the algorithm runs the modified greedy algorithm initialized to the set \(T\) then returns the set \(T'\) among all of the triples that achieved the highest ratio of the reduction in distortion to the increases in bandwidth. This algorithm increases the runtime by a factor of \(O(|E|^3)\) over the original greedy algorithm, but yields a much stronger

\(^1\)Technically it is not necessary to run the full greedy algorithm in order to achieve the approximation bound. Rather one need only run the original greedy algorithm for one iteration (i.e. selecting the best single element of the set) and then return the maximum of the best single element and the result of the modified greedy algorithm. The results of *Double-Greedy* are guaranteed to be at least as good as this and often better, but with an increase in computation time.
approximation result. We have the following result:

**Theorem 1** The Double-Greedy* and Triple-Greedy* algorithms achieve a constant factor approximation bounds of $0.5(1 - e^{-1})$ and $(1 - e^{-1})$, respectively.

**Proof 6** See the Appendix for the proof.

### 2.6 Simulation Results

In this section, we provide a comprehensive evaluation of the proposed optimized framework for video streaming in single-hop networks. In particular, the simulations provide the visual quality of video streams, as measured by Mean Square Error (MSE), when admission control is employed in conjunction with different layer allocation algorithms and the proposed MAC protocol. The layer allocation algorithms of interest are the optimal, the equal rate, the greedy, and the double greedy algorithms. We intentionally omit the results for the triple greedy algorithm since in our simulations, they are observed to be identical to those of the double greedy algorithm. This suggests that perhaps for the bit rates and distortion levels of typical video layers, the double greedy algorithm is sufficient to obtain a good solution.

We note that the optimal algorithm employs an exhaustive search scheme. That is, it examines all the possible combinations of video layers and chooses the one that results in the lowest distortion, i.e., smallest MSE that satisfies the bandwidth and distortion constraints. Thus, the optimal algorithm is prohibitively expensive when the numbers of video layers and hosts are large. The optimal algorithm, however
produces the smallest MSE, and is thus used to evaluate the goodness of other algorithms. The equal rate algorithm allocates an equal amount of bandwidth to every video (hosts), layer by layer in a round robin fashion until the constraint on total used bandwidth is no longer satisfied. The greedy and double greedy algorithms are previously described in Section 2.5.1 and 2.5.2, respectively.

In all our simulations, we use two sets of standard video profiles, each set consists of three layered videos, as shown in Tables 2.1 and 2.2 [18],[44],[45],[46],[47]. Depending on the scenarios, a simulation may use either one or both sets of the video profiles.

Table 2.1: Standard video profiles - set I

<table>
<thead>
<tr>
<th>Profile</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akiyo</td>
<td>64</td>
<td>128</td>
<td>192</td>
<td>256</td>
<td>320</td>
<td>384</td>
<td>448</td>
<td>512</td>
</tr>
<tr>
<td>Distortion (MSE)</td>
<td>83.77</td>
<td>63.54</td>
<td>50.48</td>
<td>38.29</td>
<td>32.59</td>
<td>27.74</td>
<td>23.61</td>
<td></td>
</tr>
<tr>
<td>Reduction in Distortion (MSE)</td>
<td>20.23</td>
<td>13.06</td>
<td>12.19</td>
<td>5.70</td>
<td>4.85</td>
<td>4.13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coastguard</td>
<td>123.90</td>
<td>103.06</td>
<td>87.72</td>
<td>78.18</td>
<td>71.30</td>
<td>65.03</td>
<td>57.99</td>
<td>51.65</td>
</tr>
<tr>
<td>Distortion (MSE)</td>
<td>20.84</td>
<td>15.34</td>
<td>9.54</td>
<td>6.88</td>
<td>6.27</td>
<td>7.08</td>
<td>6.30</td>
<td></td>
</tr>
<tr>
<td>Foreman</td>
<td>64</td>
<td>128</td>
<td>192</td>
<td>256</td>
<td>320</td>
<td>384</td>
<td>448</td>
<td>512</td>
</tr>
<tr>
<td>Distortion (MSE)</td>
<td>71.30</td>
<td>56.63</td>
<td>46.03</td>
<td>39.18</td>
<td>33.35</td>
<td>29.09</td>
<td>23.89</td>
<td></td>
</tr>
</tbody>
</table>

2.6.1 Protocol Evaluation

We first compare the performance of the proposed MAC protocol against the standard IEEE 802.11 without admission control [1] and the IEEE 802.11e with admission control. In particular, for the IEEE 802.11e with admission control, we use the mechanism proposed by Banchs et al. [12]. This mechanism is somewhat
similar to ours, in the sense that each flow $i$ achieves its throughput by setting the contention window $CW_i$ to an appropriate size. For our proposed MAC, we can approximate $CW_i = 2/p_i$. The fundamental difference, however is in the formulation which leads to two different algorithms, and consequently different behaviors. In the IEEE 802.11e with admission control, Banchs et al. formulated the admission control process as maximizing the total throughput from all the flows subject to the constraint on the relative throughput for each flow. In particular, the algorithm tries to set the values of the contention window for each flow in such a way to maximize $R = R_1 + R_2 + ... R_N$, while ensuring that $R_1/R_1 = n_1$, $R_2/R_1 = n_2$, ... $R_N/R_1 = n_N$ where $N$, $R_i$, and $\{n_1, ..., n_N\}$ are the number of flows, the throughput rate for flow $i$, and a set of given requirements, respectively. Note that we randomly choose $R_1$ as a reference flow. As a direct result, the throughput obtained by each flow might be higher than what is specified, especially when the specified aggregate throughput is much smaller than the network capacity. On the other hand, our algorithm produces precisely the specified rate for each flow.

Table 2.2: Standard video profiles - set II

<table>
<thead>
<tr>
<th>Profile</th>
<th>Layer</th>
<th>Bit Rates (kbps)</th>
<th>Distortion (MSE)</th>
<th>Reduction in Distortion (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(FGS-Temporal)</td>
<td>1</td>
<td>140</td>
<td>57.03</td>
<td>23.98</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>240</td>
<td>33.05</td>
<td>11.16</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>340</td>
<td>21.89</td>
<td>5.59</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>440</td>
<td>16.30</td>
<td>4.84</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>540</td>
<td>11.46</td>
<td>2.79</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>640</td>
<td>8.67</td>
<td>1.43</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>740</td>
<td>7.24</td>
<td>1.37</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>840</td>
<td>5.87</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coastguard</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(FGS)</td>
<td>1</td>
<td>110</td>
<td>60.68</td>
<td>9.03</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>160</td>
<td>51.65</td>
<td>7.69</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>240</td>
<td>43.96</td>
<td>5.67</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>300</td>
<td>38.29</td>
<td>4.63</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>360</td>
<td>31.12</td>
<td>3.63</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>420</td>
<td>26.43</td>
<td>2.33</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>490</td>
<td>24.16</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>560</td>
<td>20.56</td>
<td>1.74</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>630</td>
<td>17.50</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foreman 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(FGS-AFP)</td>
<td>1</td>
<td>384</td>
<td>42.96</td>
<td>11.84</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>512</td>
<td>31.12</td>
<td>9.69</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>640</td>
<td>26.43</td>
<td>7.69</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>768</td>
<td>22.55</td>
<td>5.67</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>896</td>
<td>19.19</td>
<td>4.63</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1024</td>
<td>17.91</td>
<td>3.63</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1152</td>
<td>16.96</td>
<td>2.95</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1280</td>
<td>14.22</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
As will explained shortly, the ability to precisely control the rate will enable an efficient cross-layer optimization.

Our simulator is a time-driven, packet-based simulator written in MatLab. It is designed to mimic as close as possible to the real operations using all the critical parameters in the IEEE 802.11e protocol. For all the simulations, the parameters specified for Frequency Hopping Spread Spectrum (FHSS) PHY layer with the channel capacity of 1 Mbps are shown in Table 2.3. We note that the MAC_{hdr} for IEEE 802.11e contains 2 bytes for a QoS field in addition to that of IEEE 802.11. We also assume that the processing and propagation delays are zero.

Table 2.3: FHSS system parameters for IEEE 802.11e used to obtain numerical results

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet payload</td>
<td>1500 bytes</td>
</tr>
<tr>
<td>MAC header (MAC_{hdr})</td>
<td>36 bytes</td>
</tr>
<tr>
<td>PHY header (PHY_{hdr})</td>
<td>16 bytes</td>
</tr>
<tr>
<td>RTS</td>
<td>20 bytes+PHY_{hdr}</td>
</tr>
<tr>
<td>CTS</td>
<td>14 bytes+PHY_{hdr}</td>
</tr>
<tr>
<td>ACK</td>
<td>14 bytes+PHY_{hdr}</td>
</tr>
<tr>
<td>Channel capacity (BW)</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>Slot time</td>
<td>50 $\mu$s</td>
</tr>
<tr>
<td>SIFS</td>
<td>28 $\mu$s</td>
</tr>
<tr>
<td>DIFS</td>
<td>128 $\mu$s</td>
</tr>
<tr>
<td>RTS timeout</td>
<td>(RTS/BW×8×10^6)+DIFS $\mu$s</td>
</tr>
</tbody>
</table>

We first show the simulation results for a single-hop wireless network consisting of 3 hosts. Each host sends exactly one video to other host over a limited channel capacity ($BW$) of 1 Mbps. These flows are assumed to be in the same traffic
class. The minimum throughput requirements for flows 1 ($R_1$) and 2 ($R_2$) are set to 200 kbps and 300 kbps, respectively. The minimum throughput of flow 3 ($R_3$) increases linearly from 115 kbps to 370 kbps with a step size of 15 kbps. For the IEEE 802.11e with admission control, $CW_i$'s are set according to the admission control algorithm while for the standard IEEE 802.11 without admission control, $CW_{min}$ and $CW_{max}$ are set to 15 and 1023 respectively.

Fig. 2.2(a)-(c) show the observed throughputs for IEEE 802.11 without admission control, IEEE 802.11e with admission control, and our proposed admission control as a function of the flow 3’s throughput. As seen, the standard IEEE 802.11 performs well when the total requested throughput is smaller than the network capacity. Without admission control, however, flow 3 cannot achieve its requested throughput of greater than 320 kbps. On the other hand, the IEEE 802.11e with admission control performs very well as the throughput for each flow is consistently above its specified minimum requirement. Unlike the IEEE 802.11e, our proposed admission control produces a precise requested throughput for each flow. As a result, the collision rate (wasted bandwidth) is much smaller than that of the standard and the IEEE 802.11e as shown in Fig. 2.2(d), even when the total useful throughputs in both schemes are approximately the same. This is an advantage of using the proposed MAC protocol. Not surprisingly, the total bandwidth usage for our algorithm is much smaller than those of other protocols as shown in Fig. 2.2(e) for a specified set of rates.

We now show the performance of the proposed contention based MAC protocols when a large number of hosts competing for a shared medium. Specifically, we
Figure 2.2: Performance comparisons for the proposed protocol versus IEEE 802.11 and IEEE 802.11e protocols; (a) Throughput for flow 1; (b) Throughput for flow 2; (c) Throughput for flow 3; (d) Overall collision; (e) Overall bandwidth usage.
simulate a scenario consisting of 30 hosts, and the bandwidth capacity is set to 54 Mbps. Each host injects one flow. The throughput of flow 1 ($R_1$) increases linearly from 120 kbps to 460 kbps with a step size of 20 kbps. The throughput of flow 2 ($R_2$) increases linearly from 80 kbps to 590 kbps with a step size of 30 kbps. For the remaining flows ($R_3$-$R_{30}$), their throughput requirements are set to 200 kbps. The parameters specified for Direct Sequence Spread Spectrum - Orthogonal Frequency Division Multiplexing (DSSS-OFDM) PHY layer with long preamble PPDU format are shown in Table 2.4. In this simulation, the PHY$_{hdr}$ and the control packets (RTS/CTS/ACK) are sent at only 1 Mbps while the data portion is sent at the full rate of 54 Mbps.

Table 2.4: DSSS-OFDM system parameters for IEEE 802.11e used to obtain numerical results

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet payload</td>
<td>1500 bytes</td>
</tr>
<tr>
<td>Coding rate</td>
<td>3/4</td>
</tr>
<tr>
<td>MAC header (MAC$_{hdr}$)</td>
<td>36 bytes</td>
</tr>
<tr>
<td>PHY header (PHY$_{hdr}$)</td>
<td>32 bytes</td>
</tr>
<tr>
<td>RTS</td>
<td>20 bytes+PHY$_{hdr}$</td>
</tr>
<tr>
<td>CTS</td>
<td>14 bytes+PHY$_{hdr}$</td>
</tr>
<tr>
<td>ACK</td>
<td>14 bytes+PHY$_{hdr}$</td>
</tr>
<tr>
<td>Channel capacity (BW)</td>
<td>54 Mbps</td>
</tr>
<tr>
<td>Slot time</td>
<td>20 µs</td>
</tr>
<tr>
<td>SIFS</td>
<td>10 µs</td>
</tr>
<tr>
<td>DIFS</td>
<td>50 µs</td>
</tr>
<tr>
<td>RTS timeout</td>
<td>352 µs</td>
</tr>
</tbody>
</table>

Fig. 2.3(a)-(d) show the observed throughputs for IEEE 802.11e resulted from using our proposed admission control as a function of the flow 1’s throughput. As
Figure 2.3: Proposed protocol validation with 30 flows over 54Mbps bandwidth. 
(a) Throughput for flows 1 and 2; (b) Throughput for flows 3-10; (c) Throughput 
for flows 11-20; (d) Throughput for flows 21-30; (e) Overall bandwidth usage; (f) 
Transmission probability.
seen, they all achieve the desired throughputs. Note however, because $\text{PHY}_{\text{hdr}}$ portions and control packets are sent at much slower rate (1 Mbps), this results in low rate of overall data portions as shown in Fig. 2.3(e). Also, the collision rate in our proposed protocol is about 1% of channel capacity when the total bandwidth usage is 90%. Fig. 2.3(f) represents the transmission probability for each host corresponding to their requested throughputs. As seen, our proposed MAC protocol performs reasonably well even in the network with a large number of competing hosts.

We now show the simulation results when applying cross-layer optimization for transmitting 3 video flows. First, to provide some intuitions about the interactions between the proposed MAC protocol and the layer allocation algorithms described in Section 2.5.2. Specifically, we present the simulation results for various quantities, e.g. throughputs, transmission probabilities, when using a simple greedy layer allocation. The simulation parameters are shown in Table 2.3. Since all video streams are in the same traffic class, they use the same $\text{TXOP}$ where $\text{TXOP} = \text{CTS} + \text{PHY}_{\text{hdr}} + \text{MAC}_{\text{hdr}} + \text{PAYLOAD} + \text{ACK} + 3\text{SIFS} + \text{DIFS}$. We use standard video profile set I in this simulation. Fig. 2.4(a) shows the average throughputs of different video streams increasing with the normalized bandwidth usage. From left to right, each point in the graph represents an additional layer being added to one of the videos according to the greedy algorithm. The rightmost point denotes the final number of layers for each video. Adding a layer to any video on each graph at this point would violate the bandwidth constraint. In other words, with the addition of a new layer, the Algorithm 2 in Section 2.3 will not able to
Figure 2.4: Performance results using the greedy algorithm. (a) Throughput; (b) Transmission probability; (c) Percentage of S slots; (d) Percentage of I, C, and S slots; (e) Distortion in MSE; (f) Rate-distortion characteristic.
find a set of transmission probabilities that satisfies the requested rates for all the videos. We note that, at this point, the total bandwidth usage is 95%, indicating a relatively high bandwidth utilization.

Fig. 2.4(b) shows the transmission probabilities for each host as a function of normalized bandwidth usage. As expected, as the number of layers increases for each video, their transmission probabilities also increase accordingly to ensure a higher chance for data transmissions. It is interesting to note that the transmission probabilities increase almost exponentially to compensate for roughly linear increase in the overall throughput. Fig. 2.4(c) shows the corresponding increases percentage of successful slots (over the number of non-data slots) for different video streams, as a direct result of increase in transmission probabilities.

However, as the transmission probabilities increase, the percentage of collision slots also increases substantially as shown in Fig. 2.4(d). Of course, the percentage of idle slots decreases accordingly. This agrees with our intuition about the proposed MAC protocol. We note that using this MAC protocol, one is able to control the rate of the flows precisely by tuning their transmission probabilities. These rates, in turn, control the visual quality of the video streams. Fig. 2.4(e) shows the visual quality of the three video streams in terms of MSE as a function of normalized bandwidth usage. In this case, the greedy algorithm which minimizes the total MSE for all the flows given the bandwidth constraint, yields an MSE of 38, 71, and 46 for Akiyo, Coastguard, and Foreman sequences, respectively. Fig. 2.4(f) shows the actual bandwidth percentage for various packet types. As seen, only minimal bandwidth overhead (2%) is incurred when using the RTS of 36 bytes
and packet payload of 1500 bytes.

Figure 2.5: Distortion performances of different algorithms for (a) Video profiles in Table 2.1; (b) video profiles in Table 2.2.
2.6.2 Layer Allocation Algorithm Performance

We now show the performance of different layer allocation algorithms. For simplicity, we assume there is no packet loss. Furthermore, by using standard video profiles H.264/SVC in Table 2.1, we require that the distortion levels (MSE) for Akiyo, Coastguard, and Foreman cannot be greater than 63, 103, and 56, respectively. For this simulation, these MSE values are chosen rather arbitrarily, but in practice a user can specify his or her visual quality requirement. Fig. 2.5(a) and Fig. 2.5(b) show the distortions resulted from using different algorithms for video profiles in Tables 2.1 and 2.2, respectively. For Table 2.2, the maximum distortion requirements for Foreman 1, Coastguard, and Foreman 2 are 21, 51, and 31, respectively. As expected, the optimal algorithm (exhaustive search) always produces the lowest distortion, albeit it has the highest computational cost.

Fig. 2.5(a) shows that the performances of the greedy and the double greedy are all identical for video profiles in Table 2.1. At 0.8 Mbps, the greedy and the double greedy algorithms fail to find the optimal solutions. However, the performances of the greedy, the double greedy, and the optimal algorithms are all identical at the capacity channel of 1 Mbps and 1.2 Mbps. This suggests that greedy and double greedy algorithms perhaps are sufficient in practice. As expected, the equal rate algorithm performs worst since it does not even try to minimize the overall distortion.

On the other hand, Fig. 2.5(b) shows that the greedy algorithm fails to find the optimal solutions in two instances ($BW = 1.6$ Mbps and 2.4 Mbps). In contrast, the
double greedy algorithm typically finds the optimal solutions. This is, however, not guaranteed as the double greedy algorithm does not converge to optimal solution for $BW=1.6$ Mbps. We note that as described in Section 2.5.2, the computational cost of the double greedy algorithm is twice that of the greedy algorithm since it must run the greedy and modified greedy algorithms, and picks the best one.

The modified greedy algorithm is the basic greedy algorithm, where at each step instead of adding the layer that most decreases the distortion, we add the layer that achieves the largest ratio of the reduction in distortion to the layer bit rate, provided that adding the layer does not violate the bandwidth constraint. The modified greedy algorithm might or might not produce a better solution than that of the greedy algorithm. Both the greedy and modified algorithms can produce an arbitrarily bad solution, while the double greedy algorithm which returns the best solution (lower distortion) of the greedy and modified greedy algorithm, guarantees that its solution is a constant approximation factor to the optimal solution.

Tables 2.5, 2.6, and 2.7 show the detail information associated with different algorithms for the video profiles I and II, respectively. As seen, the optimal algorithm always achieves in the lowest distortion. In all cases, the bandwidth overhead is relatively small, indicating the ability of the framework to utilize the bandwidth efficiently. Note that there are some other overhead bandwidths (i.e., PHY$_{hdr}$, MAC$_{hdr}$, SIFS, DIFS, ACK) that amount to 8 to 10% of the total bandwidth.
Table 2.5: Detail information associated with different algorithms for video profiles I (BW=1.2Mb/s)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Distortion (MSE)</th>
<th>Normalized Bandwidth Usage (%)</th>
<th>Total</th>
<th>Overhead S</th>
<th>Overhead C</th>
<th>Throughput</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal Solution</td>
<td>129.12</td>
<td>98.49</td>
<td>2.11</td>
<td>0.07</td>
<td>87.86</td>
<td>8.43</td>
<td></td>
</tr>
<tr>
<td>Greedy</td>
<td>129.12</td>
<td>98.49</td>
<td>2.11</td>
<td>0.07</td>
<td>87.86</td>
<td>8.43</td>
<td></td>
</tr>
<tr>
<td>Modified Greedy</td>
<td>129.12</td>
<td>98.49</td>
<td>2.11</td>
<td>0.07</td>
<td>87.86</td>
<td>8.43</td>
<td></td>
</tr>
<tr>
<td>Double Greedy</td>
<td>129.12</td>
<td>98.47</td>
<td>2.11</td>
<td>0.07</td>
<td>87.86</td>
<td>8.43</td>
<td></td>
</tr>
<tr>
<td>Equal Rate</td>
<td>156.42</td>
<td>95.86</td>
<td>2.50</td>
<td>0.02</td>
<td>83.92</td>
<td>8.02</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.6: Detail information associated with different algorithms for video profiles II (BW=2.4Mb/s)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Distortion (MSE)</th>
<th>Normalized Bandwidth Usage (%)</th>
<th>Total</th>
<th>Overhead S</th>
<th>Overhead C</th>
<th>Throughput</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal Solution</td>
<td>51.78</td>
<td>95.86</td>
<td>2.01</td>
<td>0.06</td>
<td>83.74</td>
<td>10.05</td>
<td></td>
</tr>
<tr>
<td>Greedy</td>
<td>52.02</td>
<td>97.14</td>
<td>2.04</td>
<td>0.06</td>
<td>84.84</td>
<td>10.18</td>
<td></td>
</tr>
<tr>
<td>Modified Greedy</td>
<td>51.78</td>
<td>95.86</td>
<td>2.01</td>
<td>0.06</td>
<td>83.74</td>
<td>10.05</td>
<td></td>
</tr>
<tr>
<td>Double Greedy</td>
<td>51.78</td>
<td>95.86</td>
<td>2.01</td>
<td>0.06</td>
<td>83.74</td>
<td>10.05</td>
<td></td>
</tr>
<tr>
<td>Equal Rate</td>
<td>52.60</td>
<td>96.47</td>
<td>2.92</td>
<td>0.07</td>
<td>84.26</td>
<td>10.12</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.7: Transmission probability for video profiles I (BW=1.2Mb/s) and II (BW=2.4Mb/s)

<table>
<thead>
<tr>
<th></th>
<th>Transmission probability - Profile 1</th>
<th>Transmission probability - Profile 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activity</td>
<td>Coastguard</td>
</tr>
<tr>
<td>Optimal Solution</td>
<td>0.0011</td>
<td>0.0209</td>
</tr>
<tr>
<td>Greedy</td>
<td>0.0015</td>
<td>0.0209</td>
</tr>
<tr>
<td>Modified Greedy</td>
<td>0.0015</td>
<td>0.0209</td>
</tr>
<tr>
<td>Double Greedy</td>
<td>0.0015</td>
<td>0.0209</td>
</tr>
<tr>
<td>Equal Rate</td>
<td>0.0046</td>
<td>0.0081</td>
</tr>
</tbody>
</table>
2.6.3 Throughput Jitter Evaluation

The proposed optimization framework guarantees that each flow will achieve its required throughput when it is averaged over a long period of time. However, the throughputs of the flows may fluctuate within a short period of time due to the probabilistic nature of the channel contention access. These throughput fluctuations may prevent smooth playback for many audio and video streaming applications. To alleviate this problem, many streaming applications employ the prebuffering technique in which, the receiver puts the received data into a buffer for a short period of time before starting to playback. A longer buffer results in a smoother playback session. On the other hand, using a larger buffer results in larger initial delay and required memory. Interactive applications such as video conferencing may not tolerate such large delay, and the low power wireless hosts may not have enough memory for buffering. Thus, it is important to characterize the throughput jitter resulting from using the protocol.

Fig. 2.6(a) shows the ratio of the actual throughputs to the requested throughputs of different flows in video profiles I, averaged over every 300 kbytes. The channel capacity is set to 1 Mbps. Fig. 2.6(b) shows throughput ratios for different flows in the video profiles II, averaged over every 600 kbytes, with the capacity channel set to 2 Mbps. As seen, the throughput ratios fluctuate around 1, indicating all the flows achieve their required throughputs. The magnitudes of these fluctuations are also small, e.g. 0-20%, suggesting that one can use a small streaming buffer for smooth playback.
Figure 2.6: Throughput jitter for each flow for (a) Video profiles in Table 2.1; (b) Video profiles in Table 2.2.
In this simulation, we repeatedly transmit three videos from video profiles I for 20 minutes. We want to quantify how long a streaming buffer should be in order to prevent lost packets due to late arrival. To prevent throughput fluctuation, we also request a slightly larger bandwidth than the recorded video bit rate. This bandwidth safety margin provides robustness against possible throughput depletion during a session. Fig. 2.7(a) to Fig. 2.7(d) show the number of late packets as a function of streaming buffer length for various bandwidth safety margins.

As seen, to have no late packet with no bandwidth safety margin, a user receiving Akiyo needs to wait on average, 48 seconds to smoothly playback a 20-minute video. Users receiving Coastguard and Foreman need to wait up to 88 and 40 seconds, respectively. However the required waiting time reduces with the increase in the bandwidth safety margin. For example, using the bandwidth safety margin of 3%, the waiting times for Akiyo, Coastguard, and Foreman users are reduced to 24, 32, and 12 seconds, respectively.

We also quantify the throughput jitter of a flow as the normalized standard deviation of its fractional throughput within a number of time slots. Specifically,

\[
Stdev_n(X_i(T)) = \frac{Stdev(X_i(T))}{R_i'},
\]

where \(X_i(T)\) is a fraction of the data slots of the flow \(i\) measured within \(T\) time slots, and \(R_i'\) denotes its average long term fractional throughput. Clearly, a larger \(T\) should result in a smaller normalized throughput standard deviation since we average the throughput over a longer period of time. It is straightforward to show
Figure 2.7: Percentage of late packets as a function of streaming buffer length (in seconds); (a) 0% bandwidth safety margin; (b) 1% bandwidth safety margin; (c) 2% bandwidth safety margin; d) 3% bandwidth safety margin. The bit rates of Akiyo, Coastguard, and Foreman are 192, 160, and 256kbps, respectively.
Figure 2.8: Normalized throughput standard deviation as a function of buffer size ($T$).

That

$$Stdev_n(X_i(T)) = TXOP_i \times \frac{\sqrt{(1 - \sum_{i=1}^{N} R'_i) \times S_i \times (1 - S_i)}}{R'_i \sqrt{T}},$$

(2.24)

where $N$ denotes the number of flows and $S_i$’s denotes the number of successful time slots for flow $i$. \(^2\)

To quantify the normalized throughput standard deviation, we simulate three flows with $R'_1$, $R'_2$, and $R'_3$ being set to 0.1, 0.27, and 0.4, respectively. Fig. 2.8 shows the normalized throughput standard deviations for three different flows as a function of buffer size ($T$). As expected, as $T$ increases, the normalized throughput standard deviation decreases. However, increasing $T$ implies an increase in playback delay.

\(^2\)This result can be obtained by noticing that the number of successful transmission slots within a period $T$ is binomially distributed with parameters $S_i$ and $T(1 - \sum_{i=1}^{N} R'_i)$. \(\)
2.6.4 Overall Performance

We now compare the performance of our proposed framework against the existing IEEE 802.11 without admission control and IEEE 802.11e with admission control proposed by Banchs et al. [12]. To simulate realistic settings, packet losses are introduced into the simulations. For simplicity, we assume that packet loss rates are identical for all the receivers. We also assume that a packet will be transmitted repeatedly until it is received successfully. As a result, the useful throughput reduces with an increase in packet loss rate. The admission control module is assumed to be able to measure the packet loss rate, and thus can determine the overall effective throughput. Using the overall effective throughput and the video profiles as inputs, it can use different optimization strategies to allocate bandwidths for receivers to minimize the average distortion.

Fig. 2.9(a) shows the average distortion for various strategies when streaming the videos in profile I. As expected, the optimal strategy, i.e., exhaustive search, always provides the smallest average distortion. The double greedy algorithm also performs equally well. The greedy algorithm performs slightly worse, followed by the equal rate algorithm, the IEEE 802.11 protocol without admission control, and IEEE 802.11e with admission control. The main reason for the worse performances when using the IEEE 802.11 and the IEEE 802.11e is the lack of layer allocation optimization. For the IEEE 802.11, due to the random contention-based access where each flow has an equal chance of accessing the shared channel, all three flows obtain approximately the same throughputs. However, this is not the optimal
allocation for minimizing the overall distortion. Intuitively, the IEEE 802.11 should perform as well as the equal rate allocation. On the other hand, the equal rate allocation scheme uses the proposed admission control which results in smaller collision bandwidth as compared to that of IEEE 802.11. As a result, the average distortion of the equal rate scheme is typically lower than that of the IEEE 802.11.

However, admission control mechanism alone does not improve the quality of the video. To see this, let us consider the performance of the IEEE 802.11e with admission control. Using this scheme, one is able to guarantee the minimum throughput rate for each flow. However, if after providing these minimum throughputs for the flows, there is still much bandwidth available, then as designed, the IEEE 802.11e protocol would allow each flow to increase its throughput proportionally until the wireless capacity is reached. The final throughput of each flow as obtained by the admission control algorithm then dictates the quality of a video. In other words, the layer allocation employed in this case is to allocate the rate proportionally according to the initial conditions. For example, if the minimum rates (or equivalently maximum distortions) for three flows are specified initially 200 kbps, 400 kbps, and 400 kbps, then with a channel capacity of 2 Mbps, the IEEE 802.11e protocol will roughly allocate 400 kbps, 800 kbps, and 800 kbps for these flows (for sake of illustration, we assume no collision bandwidth). Clearly, this allocation is not optimal as it does not take into the account the distortion profile for each video. That said, there is no reason why the IEEE 802.11e with admission control should perform better than the IEEE 802.11 or other schemes when there is enough bandwidth. In fact, Fig. 2.9(b) shows that the IEEE 802.11e results in
a larger average distortion consistently when using video profiles II. Our proposed framework which integrates the admission control with layer allocation optimization enables us to achieve the lowest distortion. Overall, our proposed framework improves the video quality up to 26% over that of a typical IEEE 802.11 based network.

2.7 Conclusions

We have proposed a framework to enhance the quality of video streaming applications in wireless home networks via a joint optimization of video layer-allocation technique, admission control algorithm, and MAC protocol. Using an Aloha-like MAC protocol, our admission control framework which can be viewed as an optimization problem that maximizes the average quality of admitted videos, given a specified minimum video quality for each flow. We provided some hardness results for the optimization problem under various conditions, and proposed two heuristic algorithms for obtaining a good solution. In particular, we showed that a simple greedy layer-allocation algorithm can perform reasonable well, although it is typically not optimal. Consequently, we presented a more expensive heuristic algorithm that guarantees to approximate the optimal solution within a constant factor. Based on our scenario, simulation results demonstrated that our proposed framework can improve the video quality up to 26% as compared to those of the existing approaches.
Figure 2.9: Distortions resulted from using various protocols for (a) Using video profiles I with channel capacity set to 1 Mbps; (b) Using video profiles II with channel capacity set to 2 Mbps.
Chapter 3 – On Throughput Guarantee of Aloha-Like Multi-Hop Wireless Networks

Providing Quality of Service (QoS) is one of significant issues for multimedia traffic. One approach to achieve the requested QoS is to characterize the traffic flows and guarantee their committed throughput. In a typical multi-hop wireless ad hoc network, determining the feasibility for a given set of flow characteristics is challenging due to the multi-user interference problem. To that end, this paper presents the following contributions. First, we describe a simple Aloha-like Medium Access Control (MAC) protocol that enables each flow to maintain its requested bandwidth, and thus is suitable for multimedia traffic. Second, we propose a bandwidth feasibility algorithm based on the Variable Elimination (VE) technique. The bandwidth feasibility algorithm determines whether or not a given network can support a set of flows of certain bit rates. Simulations indicate that our solution can precisely control the bit rates over all hosts while providing the throughput guarantees.

3.1 Introduction

Recent years have witnessed the rise of wireless ad hoc networking both in research and real-world deployment. A wireless ad hoc network is most useful in
places where installing a new communication infrastructure is either expensive or inconvenient to use. In such a network, each wireless host operates not only as a host but also as a router, forwarding packets on behalf of pairs of senders and receivers who are not within their radio ranges. Thus, packets are typically forwarded via multiple hops between the sending and receiving hosts. Consequently, this architecture increases the flexibility of wireless networking at the cost of increased multi-user interference. Therefore, to provide Quality of Service (QoS) for a flow, in such a network, it is imperative to determine the algorithm which takes into account the multi-user interference to accurately model the bandwidth requirement. In particular, providing the throughput guarantee is widely considered as one of the desired criteria for QoS traffic flows.

We note that guaranteeing the throughput is hard, even for a single-hop wireless network such as Wireless Local Area Network (WLAN). In a WLAN, the multi-user interference arises due to the channel contention access, in which the interference or collisions between the packets of the new flow and the existing flows reduce all the flows’ throughput. The number of these collisions increases nonlinearly with the number of competing flows, making it more difficult to decide whether or not to admit a new flow based simply on the available bandwidth [14]. On the other hand, the network will accept a new flow if it is able to guarantee that the achievable throughput of all the flows meets the requested requirements.

Furthermore, characterizing the collisions in a multi-hop wireless network is more difficult due to the hidden terminal problem. That said, designing an efficient throughput guarantee algorithm in a multi-hop wireless network is a challenging
problem. To mitigate multi-user interference, Medium Access Control (MAC) protocol is used to regulate competition for a shared communication medium among the flows. Thus, characterizing the multi-user interference is specific to a MAC protocol. In this paper, we describe an Aloha-like MAC protocol [17] which enables QoS support for media streams in terms of guaranteeing to achieve its average requested throughput. In particular, this is the first step towards providing throughput guarantee in a multi-hop network with a simple Aloha-like protocol. Using the proposed protocol, we present a bandwidth feasibility algorithm for determining whether or not a network can support a given number of flows, taking into account the multi-user interference. Therefore, the bandwidth feasibility algorithm is used in a novel framework which guarantees the throughput of all flows in a multi-hop wireless network. We note that our framework can be extended to provide the throughput guarantee and admission control algorithm in traditional IEEE 802.11 or other contention-based access networks.

For the ease of analysis, this paper mainly addresses the applications over linear wireless ad hoc networks (e.g., sensor network, railway wayside communication) where all communications require all hosts to be deployed along a line [48],[49]. In a wireless ad hoc sensor network, each sensor wirelessly transmits the detected signal to a centralized control device over a multi-hop route. In railway wayside communication, each mobile wireless device is able to access the internet through other existing devices/repeaters via a pre-defined ad hoc route. However, realistic implementation for a distance wireless connection of existing devices may require a point-to-point connection between them.
We note that our framework can be easily extended to non-linear wireless networks with the expense of higher computational complexities. Thus, our contributions are summarized as follows. First, we describe a simple Aloha-like MAC protocol that enables each flow to maintain its requested bandwidth. This is in contrast with the existing IEEE 802.11 protocols, in which the bandwidth of a flow can fluctuate widely, depending on the number of active flows. Second, we propose a bandwidth feasibility algorithm based on the Variable Elimination (VE) technique which has been used extensively in Artificial Intelligence (AI) and discrete optimization literatures. The bandwidth feasibility algorithm determines whether or not a given network can support a set of flows of certain bit rates. By using the bandwidth feasibility algorithm and the proposed MAC protocol, we guarantee that the solution satisfies all minimum requested bandwidth requirements of all the flows with minimal wasted bandwidth.

3.2 Preliminaries

3.2.1 Related Work

Providing QoS for the flows on wireless contention-based access networks is difficult because each host competes with other hosts for accessing the shared channel. The current design places no limit the number of flows entering the network, or attempt to regulate the bandwidth of individual flows unless admission control algorithm is used. For example, Aad et al. [28] proposed the method to enhancing IEEE 802.11
MAC protocol in congested networks by slowly changing the Contention Window (CW). Similarly, Banchs et al. [12] tuned up the parameters of the IEEE 802.11e in the contention-based mode. When these parameters are set appropriately, this can enable flows to achieve their requested throughputs or reduce the delay. Gao et al. [34] provided the framework by using the long-term average physical rates based scheme in IEEE 802.11e to reserve the channel for some amount of time, called the Transmission Opportunity (TXOP), for each associated host. Furthermore, Bai et al. [36] improved the bandwidth utilization by dynamically changing the transmission time based on the current traffic condition. Pong and Moors [13] proposed the strategy for QoS of flows in IEEE 802.11 by adjusting the CW and TXOP. In all aforementioned works, there is no explicit mechanism to precisely control the bit rates. However, our proposed framework sets right on how to control the bit rates by fine tuning the transmission probability for each host. The detailed discussion for our proposed framework will be discussed in section 3.3.

3.2.2 Characteristics of Wireless Networks

Transmissions in wireless networks typically take place in time slots. Often, all the hosts are equipped with a single antenna; as such sending and receiving must be performed in different time slots. Furthermore, all hosts are typically transmitting using the same carrier frequency. Therefore, a successful transmission from hosts $i$ to $j$ implies that no neighbors of host $j$ other than host $i$, are transmitting at the same time, otherwise interference would occur at host $j$. 

Centralized scheduling can be employed to coordinate the transmission schedules of all the hosts in the network in order to satisfy the above condition. However, this approach is complex and not scaled to a large network. Therefore, a scalable approach is to allow every host to send its packets opportunistically when a host determines that no other hosts are transmitting. This is the basic framework for the popular IEEE 802.11 protocols. Of course, this approach suffers from several problems. Multiple hosts may decide that no other hosts are transmitting, and thus they all decide to transmit at the same time, resulting in interference. In addition, there is also the hidden terminal problem. The hidden terminal problem arises in the scenario in which the host $i$ can listen to the transmissions of both hosts $j$ and $k$, but host $j$ cannot listen to the transmission of host $k$, and vice versa. The problem occurs when both hosts $j$ and $k$ try to transmit to host $i$ at the same time since host $j$ cannot listen to host $k$’s transmission and vice versa. Current WLAN employs IEEE 802.11 protocol to resolve these problems at the expense of some bandwidth overhead. We now first describe the basic of Aloha and IEEE 802.11 protocols. We then present our proposed MAC protocol as a conjunction between those protocols together with a feasibility algorithm to guarantee the requested bandwidth requirements.

3.2.3 Aloha Protocol

Contestion based access enables multiple hosts to compete for a shared wireless channel. The ALOHA protocol is a MAC protocol for wireless networks with
broadcast topology. There are a few remarkable Aloha protocols such as pure Aloha and slotted Aloha protocols. The basic concept of a pure Aloha protocol is described as follows. When a host has data to send, it simply sends out the data to a shared medium. If the data collides with another transmission from other hosts, that host will wait and then re-transmit the data later. We note that pure Aloha has a maximum throughput of about 18.4% of the total available bandwidth. However, there is an improvement to pure Aloha protocol called a slotted Aloha protocol. In a slotted Aloha protocol, we discretize the channel into time slots. Therefore, a host can send only at the beginning of a time slot, and thus collision is reduced. That said, a slotted Aloha protocol provides significant improvement over a pure Aloha protocol. However, these protocols do not have capability to listen to the channel before making a decision to send out the data. Therefore, researchers propose an additional technique called Carrier Sense Multiple Access (CSMA) to avoid such collisions. Note that CSMA is now considered as the standard for typical networks.

3.2.4 IEEE 802.11 Protocol

While there are many parameters in contention-based IEEE 802.11 standards, for simplicity we focus our discussion on the CW and TXOP. To access the channel, a host first senses the channel. If the channel is idle for more than the Arbitration Interframe Space (AIFS) time, the host starts sending the data. Otherwise, it sets a backoff timer for a random number of time slots between \([0, CW_{\text{min}}]\) where
$CW_{min}$ is the minimum contention window size. The backoff timer is decremented by one for each idle time slot after the AIFS time, and halts decrementing when a transmission is detected. The decrementing resumes when the channel is sensed idle again for an AIFS time. A host can begin its transmission for TXOP time slots on the channel as soon as its backoff timer reaches zero. If a collision occurs, i.e., no acknowledgment packet is received after a short period of time, the backoff timer is chosen randomly between $[0, (CW_{min} + 1)2^i - 1]$ where $i$ is the number of retransmission attempts [1]. In effect, the contention window size is doubled for each retransmission in order to reduce the traffic in a heavily loaded network. By using this procedure, collisions still occur, although less frequently, since each host effectively reduces its transmission rate when there are many active senders.

3.3 Proposed MAC Protocol

One of the main reasons to propose a new MAC protocol is that the current IEEE 802.11 protocols do not support the precise rate control to allocate different bandwidth for different flows. Instead, every flow in the current IEEE 802.11b protocol has the same priority and consumes more or less the same amount of bandwidth under similar network conditions. There have been the work to provide the service differentiation of the flows in IEEE 802.11 networks by tuning the values of $CW_{min}$ and TXOP [26] for each flow. A detailed survey of existing MAC scheme for traffic differentiation can be found in [27]. We note that the recent IEEE 802.11e standard has capability to tune up those parameters [16]. However,
the service differentiation only guarantees that a flow belonging to one type can obtain larger bandwidth than those of other types, rather than providing a specified bandwidth for a flow.

The idea to control the average bit rate of individual flows is simple. Rather than doubling the contention window size of a flow after a collision is detected, every flow maintains a fixed window size unless it is told to change explicitly by the throughput guarantee algorithm. We note that doubling the contention window size makes sense when such algorithm is not employed in the network. Thus every host must reduce its rate corresponding to an increased traffic load in order to avoid collisions. We argue that when a proper throughput guarantee algorithm is employed, eliminating this doubling of the contention window size helps to increase the bandwidth efficiency by not reducing the sending rate of each flow unnecessarily.

For simplicity to analyze our proposed MAC protocol, similar to a slotted Aloha MAC protocol, we do not employ the RTS/CTS packets to reserve the channel before sending data. Each host arbitrarily sends packets with some transmission probability $p$. That said, $p$ is the probability for transmitting a packet given that the host has a packet to send out to its neighbor in the current time slot. Note that, for each packet, the host transmits it until the first success meets which is the characteristic of the geometric distribution. Therefore, the time taken for each host to access the shared channel is a geometrically distributed random variable with parameter $p$. Thus, the effective transmission rate of a host depends on the frequency of sending packets, i.e., $p$, and their successful percentage. Furthermore,
setting appropriate values of \( p_i \)'s for each host \( i \) is a way to precisely control the transmission rates of different flows. To translate the transmission probability \( p \) back to the contention window \( CW \) used in IEEE 802.11 protocols, Bianchi [50] extensively analyzed the performance of IEEE 802.11 and showed that \( CW \) can be approximately set to \( 2/p - 1 \approx 2/p \). This is only an approximation since \( CW \) in IEEE 802.11 protocols is not reset at every time slot. Note that our proposed protocol is intentionally designed for contention-based access networks. Therefore, in the case of multi-mode devices (e.g., CDMA/OFDMA), we may apply our protocol to the situation where the bandwidth is not sufficient for all requested throughputs and multiple hosts need to compete each other in order to access the shared channel. Thus, for computing probabilities \( p \)'s, we require one host to perform this computation.

To that end, a throughput guarantee algorithm based on the proposed MAC protocol needs to accurately answer the question: \textit{Given a number of flows with specified rates in a wireless network, is there a set of transmission probabilities \( p_i \) for each host \( i \) involved in the transmissions, such that all the specified rates are achievable?} In the next section, we describe such an algorithm.

### 3.4 Feasibility Algorithm

We first formally describe a wireless ad hoc network as a graph \( G(V, E) \) with a set of vertices \( V \) and a set of edges \( E \). Each vertex \( v_i \in V \) represents a host \( i \) in the network. An edge \( e_{ij} \) between vertices \( v_i \) and \( v_j \) exists if and only if hosts \( i \) and
$j$ can listen to each other’s transmissions. We assume that all hosts use the same underlying transmission technology with a fixed known transmission capacity, e.g., 54 Mbps for IEEE 802.11g networks. Using the proposed MAC protocol in Section 3.3, a host $i$ transmits a packet with probability $p_i$ when it has a packet to send out to its neighbor.

To accurately determine whether or not a given network can support a specified number of flows with the corresponding rates, we assume that the proposed feasibility algorithm has full knowledge of the network; including the network topology, the routes of all the flows, and their corresponding rates. We implicitly assume that the routes taken by the flows are established previously by some routing protocol. These assumptions are unrealistic for a large network with many active flows which requires a significant book-keeping effort. On the other hand, for a smaller network, this approach is feasible. We hypothesize that a reasonably accurate feasibility algorithm can be implemented without the full knowledge of such a network. However, the emphasis of this paper is on the theoretical aspects of the algorithm rather than its implementation. Thus, we will discuss the feasibility of the proposed algorithms with respect to the characteristics of network topologies.

To successfully maintain an average requested bit rate of a flow, it is necessary that each link (edge) in a path connecting the source and the destination is able to support the requested bit rate. To determine the bandwidth between two hosts, we must model the transmission behaviors of associated neighbors of those two hosts in order to take into account the multi-user interference. To that end, a successful transmission from a sender directly to a receiver implies that no neighbor of the
Figure 3.1: A simple subnetwork ad hoc model when host $i$ intends to send the packets to its neighbor $j$. We need to take into account hosts $i$, $j$, and $c$ for successfully transmitting packets from host $i$ to host $j$.

receiver or the receiver is transmitting a packet when the sender is transmitting a packet. Otherwise, the packet transmitted by the sender will be collided at the receiver.

We note that the scope of this paper is mainly for a linear wireless ad hoc network logically similar to a chain topology where all communications require all hosts to be deployed along a line. However, this framework can be applied to other network types with higher computational complexities. Therefore, as an example, a linear wireless ad hoc network is depicted in Fig. 3.1 where each large arrow represents a flow on a particular link and the circle represents the transmission ranges. In particular, the transmission from hosts $i$ to $j$ is successful if none of hosts $c$ and $j$ is transmitting packets in the same time slot. Note that transmission from either hosts $d$ or $e$ will not affect transmission from hosts $i$ to $j$ due to parallel transmission property [51]. In particular, the radio ranges from hosts $d$ or $e$ do not reach to a host $j$. As a result for the transmission from hosts $i$ to $j$, we need
to take into account all associated hosts $v \in \{j, N(j)\}$ where $N(j)$ represents the neighbors of host $j$.

Therefore we can formulate the equations representing wireless behaviors of ad hoc hosts over an entire network, specified by giving: $R_i$ is the total outgoing throughput from host $i$; $R_{ij}$ is the outgoing throughput from hosts $i$ to $j$; $p_i$ is the transmission probability that host $i$ sends a packet; and $S_{ij}$ is the percentage of successfully transmitting packets from hosts $i$ to $j$.

The relationship between $R_i$ and $R_{ij}$ is that $R_i = \sum_{j \in N_I(i)} R_{ij}$ where $N_I(i)$ is the neighbor of host $i$ that host $i$ intends to send the packets to. Therefore, the fractional throughput from hosts $i$ to $j$ over total outgoing throughput from host $i$ is $\theta_{ij} = R_{ij}/R_i$. That is,

$$\theta_{ij} p_i \prod_{v \in \{j, N(j) \setminus i\}} (1-p_v) \geq S_{ij} \quad (3.1)$$

where $v \in \{j, N(j) \setminus i\}$. The solution exists if (3.1) is achievable.

Note that the percentage of successfully transmitting packets $S_{ij}$ can be represented by (3.2) where the maximum payload size of each packet for every host is equal to the transmission opportunity (TXOP); the number of packets we need to transmit for the required throughput $R_{ij}$ is $\lceil R_{ij}/TXOP \rceil$; and the size of channel capacity is $BW$. Recall that we consider the channel into time slots. For simplicity of analysis, we assume one time slot is equal to $TXOP$. Each packet contains only
a payload without PHY or MAC overhead portions. That is,

\[
S_{ij} = \frac{\text{number of packets for } R_{ij}}{\text{channel capacity}}
= \left\lfloor \frac{R_{ij}}{TXOP} \right\rfloor \frac{BW}{T XOP} \tag{3.2}
\]

Finally, we can formulate the potential function \( f_{i\rightarrow j} \) to send packets from hosts \( i \) to \( j \) as shown in (3.3).

**Function** \( f_{i\rightarrow j} \)

\[
\begin{align*}
  f_{i\rightarrow j} &= 1 \quad \text{if } \theta_{ij} p_i \prod_{v \in \{j, N(j) \setminus i\}} (1 - p_v) \geq S_{ij} \\
  &= 0 \quad \text{otherwise}
\end{align*}
\]  

(3.3)

where \( v \in \{j, N(j) \setminus i\} \) and \( 0 < S_{ij} \leq 1 \). We note that the result computed for one subnetwork is not a solution for an entire network containing a number of subnetworks. Assume there are \( n \) active hosts. Therefore, the solution for an entire network exists if the result computed by (3.4) is greater than zero. That is,

\[
\max_{p_1, p_2, \ldots, p_n} \prod_{(i, j)} f_{i\rightarrow j} \prod_{i} (1 - p_i) \tag{3.4}
\]

where \( i = 1, 2, \ldots, n \) and \( j \in N_I(i) \). The product of \( f_{i\rightarrow j} \) returns a set of solutions to achieve all desired throughput requirements. In order to avoid unnecessarily transmitted packets due to assigning unnecessarily high transmission probabilities to everyone, we use the product of \( (1 - p_i) \) together with maximizing the final
Pseudocode for a generic variable elimination

1: \( x_i := \) an assigned value to a variable \( X_i \)
2: \( \pi := \) an elimination ordering ( assume \( \{ x_1, x_2, x_3, \ldots, x_i, \ldots, x_n \} \) )
3: \( \Phi := \) a product of all the functions
4: \( \Phi_x := \) a new function where variable \( x_i \) was eliminated
5: \( \Omega := \) a set of optimal solutions for all variables
6: for each variable \( x_i \) in \( \pi \) do
   7: remove all functions containing \( x_i \) from \( \Phi \) then multiply those removed functions to form a potential function \( \Phi_x \)
   8: create a table for \( \Phi_x \). Each entry (in the same row) contains a set of all possible combinations of all variables in \( \Phi_x \) except \( x_i \) together with the best \( x_i \) that maximizes \( \Phi_x \). Then return \( \text{argmax}_{x_i} \Phi_x \)
9: for each variable \( x_i \) in reverse order of \( \pi \) do
10:   if \( x_j \) is variable in \( \Phi_x \) then
11:      Search for optimal value \( x_j^* \) in \( \Omega \) then substitute the value of \( x_j \) in \( \Phi_x \) with \( x_j^* \)
12:      Find \( x_i^* \) that maximizes \( \Phi_x \) then record \( x_i^* \) in \( \Omega \)
13: return \( \Omega = \{ x_1^*, x_2^*, x_3^*, \ldots, x_i^*, \ldots, x_n^* \} \)

Figure 3.2: Pseudocode for a variable elimination technique.

computation. Thus, this technique would provide us a solution with minimal collisions.

Note that when a number of wireless hosts are close to each other, a linear wireless network turns into a non-linear wireless network. This is because the transmission range of one host covers a number of surrounding hosts considered as its neighbors. By using a feasibility algorithm, the computational complexity relies on all neighbors of considered wireless host where we intend to transmit the packet to. Thus, our framework may not be suitable for a large network with high density of wireless hosts. Therefore, based on our assumptions over the implementation aspects, our proposed protocol is able to operate in practice if we know all of the requested throughputs, network topologies, and channel conditions.
This implies that our protocol provides high efficiency if we operate under some certain conditions.

3.5 Variable elimination technique

To determine a solution for multiple variables, one option is to employ a useful technique called Variable Elimination (VE). This technique [52] multiplies all given potential functions (each composed of at least one variable) to achieve an overall function. Clearly, an overall function gathers the properties from all variables. VE technique eliminates one variable at a time by replacing a product of all functions containing that variable with a single function. In particular, if we are eliminating a variable $X_i$ with an assigned value of $x_i$, a product of all functions containing $x_i$ becomes a new function $\Phi_{x_i}$ which does not contain $x_i$. Note that a VE technique requires knowing the range of possible solutions for all variables. Once we know the range, we discretize it into multiple bins. When we create $\Phi_{x_i}$, we will generate a table containing the results for all possible combinations for all variables in $\Phi_{x_i}$ except $x_i$. In other words, each of possible combinations in $\Phi_{x_i}$ would keep the best $x_i$ maximizing $\Phi_{x_i}$. Once we eliminate all variables, we are now able to optimally determine the solution for each variable in reverse elimination ordering. Finally, the pseudocode for a VE technique is shown in Fig. 3.2.

For a concrete example as shown in Fig. 3.3, we have three variables (i.e., $p_1$, $p_2$, $p_3$) with the possible values of $\{0.3, 0.6, 0.9\}$. We have been asked to maximize $\Phi = p_1(1 - p_1p_2)(1 - p_2p_3)$ over $p_1$, $p_2$, and $p_3$. Assume an elimination ordering
Let $\pi = \{p_1, p_2, p_3\}$

$$\max_{p_1, p_2, p_3} p_1 \cdot (1 - p_1 p_2) \cdot (1 - p_2 p_3)$$

$$= \max_{p_1, p_2, p_3} \Phi_1(p_2) \cdot (1 - p_2 p_3)$$

$$= \max_{p_1, p_2} \Phi_2(p_1)$$

where $\Phi_1(p_2) = \max_{p_1} p_1 \cdot (1 - p_1 p_2)$

$$\Phi_2(p_1) = \max_{p_1} \Phi_1(p_2) \cdot (1 - p_2 p_3)$$

$$\phi_1(p_2) = \arg \max_{p_1} p_1 \cdot (1 - p_1 p_2)$$

$$\phi_2(p_1) = \arg \max_{p_1} \Phi_1(p_2) \cdot (1 - p_2 p_3)$$

Then $p_3^* = \arg \max_{p_1} \Phi_2(p_1) = 0.3$

$$p_2^* = \phi_2(p_3^*) = 0.3$$

$$p_1^* = \phi_1(p_2^*) = 0.9$$

Finally, $\Omega = \{p_1^*, p_2^*, p_3^*\} = \{0.9, 0.3, 0.3\}$

Figure 3.3: Example for a variable elimination technique.

$\pi = \{p_1, p_2, p_3\}$. First, we eliminate $p_1$ by creating a table for a product of all terms containing $p_1$, which is $p_1(1 - p_1 p_2)$. Second, we create a table for $\Phi_1(p_2)$ that contains the best $p_1$ for each possible value of $p_2$ such that $\Phi_1(p_2) = \max_{p_1} p_1(1 - p_1 p_2)$. On the other hand, $\phi_1(p_2) = \arg \max_{p_1} p_1(1 - p_1 p_2)$ returns the best $p_1$ for any given value of $p_2$. Next, recursively compute for $\Phi_2(p_3)$. Note that we first get the optimal value of $p_3$, denoted by $p_3^*$. Then, we compute $p_2^* = \phi_2(p_3^*)$ and $p_1^* = \phi_1(p_2^*)$, respectively. Finally, we have a set of optimal solution $\Omega = \{p_1^*, p_2^*, p_3^*\}$ to maximize $\Phi$ over $p_1$, $p_2$, and $p_3$.

Consider the computational complexity where we need to determine a solution for $n$ variables. Note that the theoretical computational cost of our proposed protocol depends on type of network topologies, feasibility algorithm, and VE.
technique. Because we may not be able to control network topologies in practice and the performance of feasibility algorithms relies on VE techniques, therefore, the computational cost of the system strictly depends on the VE performance.

Clearly, the problem becomes hard if the order of polynomial equations is high and there are so many variables in the system. Fortunately, each host in a multi-hop ad hoc network gets involved with only its nearby neighbors. Therefore, the number of variables in each potential function depends on the number of its nearby neighbors where the variable here is the transmission probability $p_i$ for each host $i$. Together with the VE technique, the complexity in each iteration where we intend to eliminate the variable $p_i$ is proportional to the number of all variables over all the functions containing $p_i$. In particular, if host $i$ gets involved with other $\alpha$ hosts and the possible values in each host are discretized into $\epsilon$ bins, then the computational complexity is $O(\epsilon^\alpha)$, compared to $O(\epsilon^n)$ where $\alpha \ll n$ for a large network. Thus, for example in Fig. 3.3, the computational costs are $O(\epsilon^2)$ and $O(\epsilon^3)$ for computations with and without the VE technique, respectively.

3.6 Simulations and experimental results

We have the experimental results based on a linear wireless ad hoc network topology as shown in Fig. 3.4. The dash line represents the wireless connection between hosts. On the other hand, it shows the coverage area from one host to its neighbors. The maximum bandwidth, TXOP, and time slot size are 2 Mbps, 500 bytes, and 500 bytes, respectively. To be fair among all flows (also hosts), they use the
Figure 3.4: Example for a linear wireless ad hoc network. (a) Network with one flow; (b) Network with two flows; (c) Network with ten flows.

same TXOP. We discretize the transmission probability from zero to one with step size of 0.004. For simplicity of analysis, we have the assumptions as follows: no PHY/MAC overhead in each packet; no acknowledgement packet (ACK) when the receiver successfully receives the sent packet; when a packet gets lost, the sender simply retransmits that packet; because this is an Aloha-like MAC protocol, we do not consider Short Interframe Space (SIFS) and DCF Interframe Space (DIFS) as in IEEE 802.11 [1].

At this point, we evaluate our proposed framework by using the network topol-
ogy as shown in Fig. 3.4(a). Assume flow 1 (from host $A$ to host $F$) linearly increases its outgoing rate from 100 kbps to 460 kbps with increased rate of 10 kbps per iteration. Therefore, all senders (hosts $A$-$E$) have the same desired outgoing rates. The simulation results are shown in Fig. 3.5 which we averaged out over 200 runs for each iteration. The transmission probabilities for all senders (hosts $A$-$E$) are shown in Fig. 3.5(a). As expected for a linear network topology with one traffic flow, the host closer to an origin of the flow has higher transmission probability than the host further away from an origin. On the other hand, the last-hop sender would have the lowest transmission probability because its transmission is always successfully received by the destination. Therefore, transmission probability for a host $E$ would be minimal compared to those for other hosts in this network. The outgoing (throughput) rates of all senders are shown in Fig. 3.5(b). We show that our novel framework precisely controls the outgoing rates of all senders.

To verify that our proposed protocol is not limited to a one-flow network, we setup the network topology with two flows as shown in Fig. 3.4(b). We linearly increase the outgoing rate of flow 1 (from hosts $A$ to $E$) from 100 kbps to 460 kbps with increased rate of 10 kbps per iteration. The outgoing rate for flow 2 (from host $C$ to host $F$) is constant at 100 kbps. In this example, both flows have the same direction. The corresponding results for transmission probabilities are shown in Fig. 3.6(a). Regardless of the number of flows in the network, our framework always precisely controls the rate with minimal deviation as shown in Fig. 3.6(b) if there exists the solution.
Table 3.1: Performance of our proposed framework based on a network topology as shown in Fig. 3.4(c)

<table>
<thead>
<tr>
<th></th>
<th>Expected outgoing rate (kbps)</th>
<th>Transmission probability</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>host A</td>
<td>210</td>
<td>0.700</td>
<td>99.8861</td>
</tr>
<tr>
<td>host B</td>
<td>310</td>
<td>0.600</td>
<td>99.8931</td>
</tr>
<tr>
<td>host C</td>
<td>540</td>
<td>0.565</td>
<td>99.8992</td>
</tr>
<tr>
<td>host D</td>
<td>560</td>
<td>0.370</td>
<td>99.8993</td>
</tr>
<tr>
<td>host E</td>
<td>470</td>
<td>0.240</td>
<td>99.8986</td>
</tr>
</tbody>
</table>

For concrete results, we show the degree of scalability of our proposed protocol with 10 flows over the network topology as shown in Fig. 3.4(c). The transmission rates for flows 1-10 are 100, 300, 30, 40, 50, 60, 70, 80, 90, and 100 kbps, respectively. The corresponding results are represented in Table 3.1. Note that our protocol guarantees the achievable throughput about 99.98% compared to expected throughput.

We note again that even though it is possible to have more than one solution, our proposed technique is able to provide a solution with minimal value of transmission probabilities based on (3.4). Furthermore, smaller step size of transmission probabilities would provide a better solution in terms of higher accuracy to control the outgoing rates of the hosts but computational cost is more expensive. Therefore, this framework is able to be applied for applications requiring less fluctuation in achievable throughput, especially for multimedia streaming.
3.7 Conclusions

In this paper, we propose an Aloha-like MAC protocol which enables QoS support for media streams in terms of guaranteeing to achieve its requested bandwidth. This is in contrast with the existing IEEE 802.11 protocols in which the bandwidth of a flow can fluctuate widely, depending on the number of active flows. Then, we propose a bandwidth feasibility algorithm based on Variable Elimination technique. By using the bandwidth feasibility algorithm together with a proposed MAC protocol, we are able to determine the solution (if one exists) to meet all requirements for a given network while providing minimal wasted bandwidth caused by multi-user interference. We note that this is the first step towards providing throughput guarantee in a multi-hop network. Furthermore, our framework is not limited to operate under a simple Aloha-like network but this also can be extended to provide the throughput guarantee and admission control algorithm in traditional IEEE 802.11 or other contention-based access networks with minimal modification. Our simulations indicate that the proposed framework is able to precisely control the achievable throughput for every host within 1% deviation from requested requirements while providing the throughput guarantee.
Figure 3.5: The outgoing rates from each sender based on a network topology as shown in Fig. 3.4(a). (a) Transmission probabilities; (b) Outgoing rates
Figure 3.6: The outgoing rates from each sender based on a network topology as shown in Fig. 3.4(b). (a) Transmission probabilities; (b) Outgoing rates
Chapter 4 – Efficient Wireless Broadcasting Through Joint Network Coding and Beamforming

In this paper, we develop a framework that exploits Network Coding (NC) and Multiple-Input/Multiple-Output (MIMO) techniques, jointly together, to improve throughput of downlink broadcast channels. Specifically, we consider a Base Station (BS) equipped with multiple transmit antennas that serves multiple Mobile Stations (MSs) simultaneously by generating multiple signal beams. Given the large number of MSs and the small number of transmit antennas, the BS must decide, at any transmission opportunity, which group of MSs it should transmit packets to, in order to maximize the overall throughput. We propose two algorithms for grouping MSs that take advantage of NC and the orthogonality of user channels to improve the overall throughput. Our results indicate that the proposed techniques increase the achievable throughput significantly, especially in highly lossy environments.

4.1 Introduction

In recent years, Multiple-Input/Multiple-Output (MIMO) has been recognized as a key enabling technology for improving the performance of wireless communication systems. Unlike traditional communications, MIMO techniques rely on multiple
antennas to transmit and/or receive signals. The number of antennas that a device can be equipped with can be limited due to spatial constraints. For example, in a cellular network, while there may be no limit on the number of antennas that the Base Station (BS) can be equipped with, there is a limit on that number when it comes to Mobile Station (MS) due to size and/or cost constraints. MIMO capabilities can be exploited to enable the Spatial Division Multiple Access (SDMA) technique, which basically allows multiple simultaneous transmissions from the BS to multiple MSs, thereby achieving higher overall data throughput [53, 6, 54]. Specifically, the BS exploits MIMO to generate radiation patterns that simultaneously target different groups of MSs [55],[8]; this is known as beamforming (BF). Mathematically, the signal beam from each antenna is coded and multiplied independently by a BF weight vector to control its shape and direction. A BF weight vector is determined by MSs’ locations as well as the characteristics of channels between the BS and MSs. Assuming that the BF weight vectors are available, an optimal SDMA scheme consists then of selecting the set of MSs whose Signal-to-Interference plus Noise Ratios (SINRs) are maximized.

A BS with $M$ transmit antennas can form and transmit a signal with at most $M$ beams [7, 11]. Therefore, assuming that one beam is allocated for each MS, when the number of MSs, $K$, exceeds the number of transmit antennas, $M$, the BS selects the best $M$ MSs; i.e., those whose SINRs are maximized [56]. An efficient scheduling algorithm is then required at the BS to map the MSs with these beams. For example, given 10 MSs ($z_1, z_2, ..., z_{10}$) and one BS equipped with 2 antennas, an efficient scheduler may select ($z_2, z_4$) in the first time slot for transmission, ($z_1,
$z_5$) in the second time slot, $(z_7, z_9)$ in the third time slot, and so on. After five time slots, the BS completes then transmitting all packets to their intended MSs. The key challenge in designing these scheduling algorithms lies in grouping the MSs in such a way that interference among MSs is minimized; i.e., a transmitted beam intended for one MS should create no or minimal interference to any other MSs in the group. One way to achieve this objective is to group together MSs whose channel vectors are somewhat orthogonal to one another.

Network Coding (NC) is another effective technique that is also well recognized for its great network throughput potentials, and has received a considerable attention due to its practical simplicity [57]. In single-hop wireless networks, such as cellular networks, NC can be used to significantly reduce the number of transmission/retransmission attempts required to successfully deliver data packets [58]. Consider the example of Fig. 4.1 for the purpose of illustration. Instead of immediately retransmitting a lost packet $a_1$ for an MS, e.g. $z_1$, the BS waits until there is another lost packet $a_2$ for another MS, e.g. $z_2$. Due to spatial proximity of $z_1$ and $z_2$, it is possible that $a_1$, although intended for $z_1$, may be received and cached successfully by $z_2$; likewise, $a_2$ may also be cached successfully by $z_1$. In that case, the BS only needs to retransmit a single packet $a_1 \oplus a_2$, which is a bit-wise eXclusive-OR (XOR)$^1$ of packets $a_1$ and $a_2$. If this packet is successfully received by both $z_1$ and $z_2$, then $z_1$ can recover $a_1$ as $a_2 \oplus (a_1 \oplus a_2) = a_1$ and $z_2$ can recover $a_2$ as $a_1 \oplus (a_1 \oplus a_2) = a_2$. Thus, one retransmission allows both MSs

$^1$Here, all packets are assumed to have the same bit length so as to be able to perform the XOR operation.
to recover their lost packets, resulting in a reduced number of retransmissions or, equivalently, in an increased throughput efficiency.

In this paper, we develop scheduling algorithms that exploit NC and BF techniques jointly together to maximize the overall throughput achievable on a wireless broadcast downlink channel. Specifically, we consider a BS with multiple transmit antennas that simultaneously serves multiple MSs by generating multiple signal beams with well-defined beamforming weight vectors, where each MS corresponds to a particular beam. Given a large number of MSs and a small number of transmit antennas, the BS must decide, at any transmission opportunity, which group of MSs it should transmit packets to, in order to maximize the overall throughput. In this work, we prove that the group formation problem is NP-hard, and propose two scheduling heuristics that take advantage of NC capabilities and the orthogonality of user channels to improve the overall throughput on broadcast transmissions. Our simulation results show that the proposed techniques achieve greater throughput than that achievable under existing techniques, especially in highly lossy environments.

The outline of our paper is as follows. First, we present a literature review in Section 4.2. Then, we overview NC techniques and discuss our joint BF-NC
framework in Sections 4.3 and 4.4, respectively. In Section 4.5, we describe our proposed joint BF-NC based scheduling algorithms. We show the simulation results for the proposed techniques in Section 4.6, and conclude the paper in Section 4.7.

4.2 Related Work

Simultaneous service support of multiple MSs through BF techniques can be very challenging due to interference incurred by different active signal beams. Therefore, several efficient scheduling algorithms and proper BF strategies have been proposed to reduce interference among multiple beams. To maximize the sum capacity, the scheduled MSs are typically selected through an exhaustive search, which can be very costly or even infeasible for a large MS pool. For example, Shad et al. [9] proposed an algorithm for selecting the MSs for each time slot over SDMA/TDMA networks by considering the largest SINR margin of scheduled MSs with respect to some SINR thresholds. Choi et al. [59] provided a scheduling algorithm, which relies on the feedback from MSs during each time slot to determine a subset of MSs in such a way to maximize the sum capacity. Ravindran et al. [10] studied the performance of zero-forcing BF techniques under limited channel feedback in Line Of Sight (LOS) channels. Their results show that a zero-forcing BF technique performs well especially when the system has reliable Channel State Information (CSI) from MSs. Furthermore, Yoo and Goldsmith [11] considered the orthogonality of CSIs among the MSs to schedule the MSs while minimizing the overall interference. In particular, their technique greedily selects a set of MSs
while assuming that the BS knows CSIs of all MSs in the system.

Another traditional technique, yet efficient, is to use a BF codebook containing a set of all possible coefficients (or codewords) to form a BF beam. In general, the codebook is generated offline and consists of a fixed number of codewords which do not depend on statistical characteristics of channel conditions [60]. Most codebooks are fixed and designed for a specific transmitter. For example, Love et al. [61] proposed a way to generate a codebook which cannot adaptively change as the channel changes. By using their technique, both BS and MSs have full knowledge of a codebook. Amiri et al. [62] provided an efficient algorithm to adaptively change a codebook based on new feedback information at the BS. However, their technique is complex because the BS needs to frequently evaluate new feedback information, and hence, the technique may not be suitable for systems with a large number of MSs. Huang et al. [63] showed the performance of orthogonal BF techniques by selecting a codeword for each MS in such a way to reduce the interference among all active beams. More specifically, their technique consists of finding a set of codewords from a fixed codebook for all active beams by relying on MSs’ CSIs.

Most of these reported works consider associating/allocating one BF beam for each intended MS. However, MSs within a close proximity often experience similar channel conditions and have similar SINR levels. Therefore, considering grouping and associating these MSs together under one BF beam not only allows to support more MSs, but also enables effective use of NC techniques, thus improving throughput even further. This work proposes a framework that takes into con-
sideration the proximity in SINRs and CSI orthogonality when grouping MSs and forming BF beams. We first prove that the group formation problem is NP-hard (the proof is given in Appendix), and then propose two heuristic techniques: a greedy-selection based technique [64] and a codebook based technique, which are described later in Section 4.5.

4.3 Network Coding and Broadcasting

For notational convenience, throughout this paper, uppercase boldface letters are used for matrices and lowercase boldface letters are used for vectors, and $X^*$, $X^T$, $\tilde{X}$, $|X|$, and $\|X\|$ denote conjugate transpose, transpose, normalized, size or magnitude, and L-2 norm of $X$, respectively.

In this section, we briefly describe the use of NC technique over wireless broadcast channels. We consider $K$ MSs (as receivers) and one BS (as the sender). Without loss of generality, we assume that each data packet can be transmitted in one time slot (all time slots are of a fixed duration). We also assume that MSs use either positive or negative acknowledgements (ACK/NAK) to let the BS know about the status of receiving each packet. Traditionally, an MS immediately sends a NAK if the packet is not successfully received. When using an XOR NC technique, instead of immediately retransmitting a lost packet to a certain MS right after receiving a NAK, the BS waits for a fixed interval of time (in time slots) before retransmitting starts. With NC enabled, the BS then retransmits a XOR-coded packet for each set of multiple lost packets. For this to happen, the BS maintains
Figure 4.2: An example of a pattern of lost packets at MSs $z_1$ and $z_2$.

Consider a broadcast transmission example of a cellular network, where all BS’s packet transmissions are assumed to reach all MSs. For a system with two MSs, i.e. $z_1$ and $z_2$, we consider the pattern of lost packets shown in Fig. 4.2 as an example, where a cross $X$ and an oval $O$ respectively represent a lost and a coded packet. Since $K = 2$, the coded packets generated by the BS and intended for the two MSs, $z_1$ and $z_2$, are $(a_1 \oplus a_2)$, $(a_3 \oplus a_4)$, and $a_5$. Packet $a_6$ is only intended for MS $z_2$ because MS $z_1$ has already received it successfully. Note that if some coded packets cannot be correctly received by any of the intended MSs, the BS will update the lost packet table and determine the best coded packet for the next retransmission opportunity.

Without loss of generality, for a system with $K$ MSs, we assume that $PER_i \leq PER_j$ if $i \leq j$ for any $i, j \in \{1, 2, ..., K\}$ where $PER_i$ is the packet error rate for
MS $z_i$. Recall that the BS will generate a coded packet combining as many lost packets as possible, and the MS with the highest PER dominates the performance of the system. Let $z_K$ be this MS. Let $X_i$ be a random variable representing the number of lost packets of MS $z_i$ after a certain number $N$ transmissions. When $N$ approaches infinity and each transmission follows a Bernoulli trial, $X_i$ will follow the Binomial random variable. Therefore, we have

$$E\left[ \max_{i \in \{1, 2, \ldots, K\}} \{X_i\} \right] \approx E[X_K] \quad (4.1)$$

With $N$ transmitted packets to $K$ MSs, the expected number of transmissions can be written as

$$N + E[X_K] = N + \frac{N \times \max_{i \in \{1, 2, \ldots, K\}} \{PER_i\}}{1 - \max_{i \in \{1, 2, \ldots, K\}} \{PER_i\}}\frac{1}{N} \quad (4.2)$$

Note that the NC technique described in this section considers that packets will be broadcast to all MSs. Thus, packets sent by the BS can either be heard by all MSs or by none.

### 4.4 Joint Beamforming and Network Coding

In this work, we consider broadcast transmissions in a downlink channel of a cellular network, where data packets sent by the BS are intended for and reach all MSs
in the network. When data packet transmission is not successful, the BS keeps retransmitting until the lost/unsuccesful packet is delivered to all MSs successfully. We assume that the BS needs to broadcast \( D \) packets to the MSs.

After the first transmission, one or more MSs may not receive the packets. When this happens, one simple way to ensure successful delivery of packets is for the BS to retransmit/broadcast each lost packet to all MSs again (e.g., Fig. 4.3(a)). Another retransmission technique is to use beamforming to retransmit the lost packets only to those MSs which did not receive the packets. In this paper, we investigate and develop techniques that rely on MIMO to form BF beams so as to direct transmission towards a set of MSs only; i.e., those MSs which did not receive the packets (e.g., Fig. 4.3(b)). In addition to BF techniques, our proposed techniques rely on and incorporate network coding (NC) to further improve throughput performances of broadcast channels.

Simply broadcasting lost packets to all MSs, instead of using BF technique, can be very inefficient for the following reasons. First, MSs are often located at
different positions with various distances from the BS, leading to high variations of MSs’ received signal strengths. With this, different MSs are likely to experience different packet success rates. Second, without directing the transmitted signals, power could be wasted when signals are sent in directions/regions that happen to have no MSs. Let us consider the example shown in Fig. 4.3, where there are 3 groups of MSs: \{z_1, z_2\}, \{z_3\}, and \{z_4\}. In this example, transmitting packets to a location that does not contain any of the groups will be a waste of energy resources. By using a BF technique as in Fig. 4.3(b), the BS can direct BF beams in such a way to increase the signal quality reaching the intended MSs. Note that the BS may use multiple time slots for transmitting a packet to everyone if all MSs are located apart from each other. The challenge as well as the novelty of the proposed BF technique lies in grouping all MSs into multiple clusters each corresponding to one beam in such a way to enhance broadcast throughput while minimizing power consumption.
We now describe how to combine NC and BF to further improve broadcast network throughput. Figs. 4.4(a) and 4.4(b) illustrate the use of NC to broadcast a stream of packets in the network respectively shown in Fig. 4.3(a) (without beamforming) and Fig. 4.3(b) (with beamforming). In this example, there are 4 MSs (i.e., $z_1, z_2, z_3, z_4$). The BS needs to deliver 7 packets (i.e., $a_1, a_2, ..., a_7$) to all MSs, and can generate at most 3 beams at once. Recall that the best strategy to generate a coded packet is to maximize the number of lost packets that can be coded at once. Therefore, we use 3 time slots to generate and send coded packets. In the first time slot, when not using a BF technique, the coded packet to be sent to all MSs is $(a_1 \oplus a_2 \oplus a_3 \oplus a_4)$. Whereas, when using a BF technique, the three coded/uncoded packets, $(a_1 \oplus a_2)$, $a_3$, and $a_4$, are to be sent on beams 1, 2, and 3, respectively. In the second time slot, the BS generates coded packets that combine the lost packets $a_5$ and $a_6$, and sends them to MSs $z_1$, $z_2$, and $z_4$. When not using a BF technique, the coded packet $(a_5 \oplus a_6)$ is then to be sent to all MSs. Whereas, when using a BF technique, the coded packet $(a_5 \oplus a_6)$ and uncoded packet $a_6$ are to be transmitted through beams 1 (for $z_1$ and $z_2$) and 3 (for $z_4$), respectively. In the third time slot, the BS needs to transmit only packet $a_7$ to MS $z_4$. In this case, it is obvious that there is no need to broadcast a packet $a_7$ to everyone. Hence, when using a BF technique, the signal quality sent to MS $z_4$ can be improved by allocating all transmit power to beam 3, thereby increasing the packet success rate and/or reducing energy consumption.

In the first time slot, note that the coded packet combining the first 4 lost packets (i.e., without BF) when sent to all MSs exactly suffices as it enables each
of the 4 MSs to recover its lost packet (i.e., $a_i$ for $z_i$). Thus, this may seem to be the best strategy. Likewise, the BF technique also suffices; i.e., sending $(a_1 \oplus a_2)$, $a_3$, and $a_4$ on three different beams 1, 2, and 3, respectively, also enables all MSs to recover their lost packets. The difference between the two techniques lies, however, on the fact that BF technique can make received signals stronger, thus decreasing the packet error rates by increasing the chance of correctly decoding received packets.

Note that there is a tradeoff in grouping MSs when using a BF technique. Grouping a large number of MSs under the same beam may reduce the number of coded packets but this would also lead to lesser chances of successfully receiving coded packets by an intended MS. Let us consider the example with 4 MSs as shown in Fig. 4.5. As can be seen in Fig. 4.5(a), with one beam, the BS can simply use one coded packet for all 4 intended MSs. On the other hand, forming multiple groups of MSs (e.g., 2 MSs per beam as in this example) requires generating more coded packets, but MSs are more likely to receive higher signal quality, resulting in higher chances to successfully receive coded packets. Referring to Fig. 4.5(b) again for illustration, the BS uses 2 coded packets: one for MSs $z_1$ and $z_2$, and the other for MSs $z_3$ and $z_4$. Note that the BS here can transmit 2 coded packets at the same time or at two successive time slots. One strategy to determine the sizes/groups of MSs for BF techniques is to minimize the variance of MSs’ lost packets in each group. This would allow the BS to efficiently generate coded packets while all MSs in each group can successfully receive all required packets at the same time. Therefore, efficient grouping algorithms that can determine the best sizes/groups
are needed. In this paper, we develop and propose two efficient algorithms for grouping MSs.

We assume that the BS is equipped with multiple antenna systems that enable it to simultaneously generate multiple transmitted beams. We also assume that the BS knows all channel vectors $h_i$ associated with each MS $z_i$ for all $i$ through feedback channels. Let $P_i$ be the power allocated to beam $i$ by the BS. For each beam $i$, we assign a group of MSs $V_i$ where $V_i \subseteq V$, $V_i \cap V_j = \emptyset$ for any $i, j \in \{1, 2, ..., M\}$, and $V$ is a set of all MSs. Groups are formed in such a way that all MSs belonging to the same group are likely to experience the same packet losses, reducing then the number of transmissions at the BS via bit-wise XOR operations.

The objective is to minimize the time $T$ for the BS to successfully deliver all $D$ packets to each MS. Note that minimizing the time $T$ is equivalent to maximizing the overall throughput under some fairness constraints. However, if a particular MS is far away from the BS such that it cannot receive packets, then clearly $T = \infty$. In this work, we do not consider this degenerated case, and such an MS is likely to be serviced by a neighboring BS. Furthermore, we assume that the BS uses a
round-robin transmission style and the channels at MSs are stationary. Thus, each MS will receive a fair share of transmission packets and the expected achievable throughput would reflect the performance of the considered system.

Recall that the signal quality at MSs can be assessed/quantified via SINR. In particular, $SINR_{ij}$ for MS $z_j$ in beam $i$ can be computed as

$$\text{SINR}_{ij} = \frac{P_i|h_jw_i|^2}{\sum_{k=1(k\neq i)}^{M} P_k|h_jw_k|^2 + \sigma^2}$$

(4.3)

where $P_i$, $w_i$, $h_j$, $\sigma^2$, and $M$ are transmit power, weight vector for beam $i$, channel vector for MS $z_j$, noise power, and number of beams, respectively. If all transmitted beams are pairwise orthogonal such that $|w_iw_k|^2 = 0$ and $|\tilde{h}_jw_i|^2 = 1$ (assume that $h_j$ is normalized as $\tilde{h}_j$ so as to check its orthogonality) where $z_j \in V_i \setminus V_k$, then $|h_jw_k|^2 = 0$. Thus, Eq. (4.3) would reduce to $\text{SINR}_{ij} = P_i|h_jw_i|^2/\sigma^2$ (Signal-to-Noise Ratio (SNR)). If all scheduled MSs have high SINR (or SNR), the equal power allocation is close to an optimal water-filling method [63]. Thus, the allocated transmit powers for all beams are all equal. Given a fixed total transmit power $P$ at the BS, this implies that the transmit power allocated for each beam $j$ is $P_j = P/M$ where $M$ is the number of active beams during that time.

Let $n_j$ be a set of lost packets for MS $z_j$. The number of transmissions for broadcast schemes over a group $V_i$ depends on the combination of all lost packets from all MSs in $V_i$. Therefore, a set of coded and uncoded lost packets in $V_i$ can be considered as $\mathcal{N}_{V_i} = \cup z_j n_j$ where $n_j$ belongs to $z_j$ and $z_j \in V_i$. Each group $V_i$
is independent of any other group $V_j$ for $i \neq j$. This implies that the combination process of group $V_i$ is totally independent of that of group $V_j$. Within each beam, the MS with the highest packet error rate would dominate the performance of that beam. For each beam $i$, the expected number of transmissions can be formulated as

$$\frac{N_{V_i}}{1 - \max_j \{x(i, j) \cdot \text{PER}_{ij}\}}$$

(4.4)

where $N_{V_i}$ is a set of coded/uncoded lost packets based on the NC technique applied to beam $i$, $\text{PER}_{ij}$ is packet error rate for MS $z_j$ in beam $i$, and $x(i, j)$ is an assignment binary variable for MS $z_j$ in beam $i$ defined as

$$x(i, j) = \begin{cases} 1 & \text{if MS } z_j \text{ is in beam } i \\ 0 & \text{Otherwise} \end{cases}$$

Furthermore, each MS can only belong to one beam; i.e., $\sum_i x(i, j) = 1$. For simplicity of analysis, we do not use any error correcting code. Therefore, $\text{PER}_{ij}$ can be computed in terms of bit error rate ($\text{BER}_{ij}$) such that $\text{PER}_{ij} = 1 - (1 - \text{BER}_{ij})^\alpha$ where $\alpha$ is a packet size in bits. Assume BPSK is used. Thus,

$$\text{BER}_{ij} = 0.5[1 - \text{erf}(\sqrt{\text{SINR}_{ij}})]$$

where $\text{erf}(\gamma) \approx \frac{2}{\sqrt{\pi}} \left[ \gamma - \frac{\gamma^3}{3} + \frac{\gamma^5}{10} \right]$ and $\gamma = \sqrt{\text{SINR}_{ij}}$ [65]. The expected number of transmissions for this system is the time that the BS spends until all MSs
successfully receive all of their required packets.

Without loss of generality, Eq. (4.4) can be written as

$$\left\{ \frac{N_{V_i}}{1 - \max_j \{x(i,j)PER_{ij}\}} \right\}_{l_q} \tag{4.5}$$

where $l_q$ is the index of sets of BF beams. For each set $l_q$, the BS requires to successfully transmit all lost packets over a set of all BF beams before moving forward to the next set of BF beams, i.e. $l_{q+1}$. Assume there are $Q$ sets of BF beams to be used at the BS. By taking into account all $Q$ sets of BF beams, our goal is then to minimize the total number of transmissions; i.e.,

$$\text{minimize} \quad \sum_q \left\{ \max_i \left\{ \frac{N_{V_i}}{1 - \max_j \{x(i,j)PER_{ij}\}} \right\}_{l_q} \right\}$$

s.t. \quad \sum_i x(i,j) = 1 \tag{4.6}

$$x(i,j) \in \{0, 1\}$$

where $q = \{1, 2, ..., Q\}$. The solution to this optimization problem will be $x(i,j)$ assignments that minimize the total number of transmissions. However, we prove that this assignment problem is NP-hard (we provide the proof of NP-hardness in the Appendix), and instead, we propose two heuristics/techniques that solve this group formation problem by taking advantage of NC capabilities and the orthogonality of user channels. One proposed technique uses greedy selection based on SINR levels, whereas the other technique uses a predefined codebook to generate
a set of orthogonal BF beams. Let \( L = \{l_1, l_2, ..., l_Q\} \) denote the \( Q \) sets of \( M \) orthogonal beams. For example, \( l_i = 1 \) means the BS uses the \( i^{th} \) set of \( M \) orthogonal beams in a codebook, otherwise \( l_i = 0 \). Next, we describe our two proposed scheduling techniques.

### 4.5 Proposed Scheduling Algorithms

Because our assignment problem is NP-hard, we propose to use two techniques to design heuristic algorithms that efficiently solve it. The first technique schedules and groups MSs based on their SINRs, and the second one uses a predefined codebook that first generates multiple sets of orthogonal BF beams, and then assigns each MS to the beam whose SINR is maximized. The details of proposed techniques will be described next.

#### 4.5.1 Technique 1: Greedy Selection-Based Scheduler

We first describe and formulate the group formation problem which basically tries to find the optimal set of MSs that maximizes the overall throughput, and then propose a scheduling algorithm that solves it. The scheduling algorithm improves overall throughput through joint exploitation of both BF and NC techniques. Given an \( M \)-transmit antenna BS and \( K \) one-receive antenna MSs located within the communication range of the BS, the BS can choose to transmit packets to a selected set of MSs that are located far apart from each other, and whose quality
levels are above a certain threshold. Note that the use of linear, equally spaced antenna arrays limits the number of concurrent signal beams to a certain number. We denote this number by $M^{opt}$ where $M^{opt} \leq M$.

When the number of MSs, $K$, exceeds $M^{opt}$, the BS chooses the best $M^{opt}$ (or fewer) MSs with the highest SINRs. We consider spatial separation among MSs by using the orthogonality in channel vectors, described in [11]. Below, we describe our greedy selection-based scheduling algorithm.

**Algorithm 1** Greedy Selection-Based Algorithm

**step 1:** Initialization

$i = 1$

$T_i = \{1, 2, ..., K\}$

$S = \emptyset$ (empty set)

**step 2:** For each MS $k \in T_i$, compute $g_k$ where $g_k$ is determined by $h_k$ semiorthogonal to the subspace $\{g_1, ..., g_{i-1}\}$

$$g_k = h_k \left( I - \sum_{j=1}^{i-1} g_j^* g_j \right)$$

(Note that when $i = 1$, assume $g_k = h_k$)

**step 3:** Select the $i^{th}$ MS such that

$$\pi(i) = \arg \max_{k \in T_i} \|g_k\|$$

$S := S \cup \{\pi(i)\}$

$h_i = h_{\pi(i)}$

$g_i = g_{\pi(i)}$

**step 4:** Find a cluster of MSs whose SINRs are greater than some threshold $\Omega$. We denote that cluster by $G(\pi(i))$ where $\pi(i) \in G(\pi(i))$ and $G(\pi(i)) \subset T_i$.

**step 5:** If $|S| < M^{opt}$, compute $T_{i+1}$ as follows:

$T_{i+1} := \{k\}$

where $k \in T_i, k \notin G(\pi(i))$, and $\frac{|h_k h_i^*|}{\|h_k\| \|h_i\|} \leq \alpha$

$i = i + 1$

(Note that if $T_{i+1} \neq \emptyset$ and $|S| < M^{opt}$ then go to step 2. Otherwise, the algorithm terminates and returns a set of scheduled MSs.)
The details of our proposed algorithm are described as follows: In step 1, \( T_i \) is a set of all available MSs and \( S \) is a set of scheduled MSs where \( i \) represents the current iteration number. Thus, \( S \) is initially empty and \( T_i \) is equal to \( \{1, 2, ..., K\} \). In step 2, we compute \( g_k \), the component of channel vector \( h_k \) for each \( k \) in \( T_i \), orthogonal to previously determined subspace spanned by \( \{g_1, ..., g_{i-1}\} \). Note that when \( i = 1 \), we assume \( g_k = h_k \) for each MS \( k \). In step 3, we select the MS, denoted by \( \pi(i) \), whose norm \( \|g_k\| \) is maximized (i.e., best MS) over each \( k \in T_i \). After that we update the values of \( S \), \( h_i \), and \( g_i \), consecutively. In step 4, after finding the best MS \( \pi(i) \), we further determine a cluster of MSs whose SINRs are greater than some threshold \( \Omega \). We denote this cluster by \( G(\pi(i)) \). Note that the value of \( \Omega \) depends on MSs’ channel gain as well as their spatial distribution within a BS’s coverage area. Finally in step 5, we determine \( T_{i+1} \) for next iteration by eliminating all MSs in \( G(\pi(i)) \) and all MSs whose channel vectors are not semiorthogonal to \( \pi(i) \) within a certain threshold \( \alpha \). If \( T_{i+1} \) is not an empty set and \( |S| < M^{\text{opt}} \), then go to step 2. Otherwise, the algorithm terminates, and returns a set of \( \pi(i) \) with their corresponding cluster \( G(\pi(i)) \).

Note that Algorithm 1 requires two thresholds \( \Omega \) and \( \alpha \). In practice, one can try multiple thresholds in order to select/determine the one that minimizes the number of total transmissions. We now propose our second technique.
4.5.2 Technique 2: Codebook-Based Scheduler

The second technique, as described in the pseudocode shown below, uses two algorithms: Algorithm 2, which determines the best beam for each MS whose SINR is maximized; and Algorithm 3, which further minimizes the number of transmissions. The outcome will be a set of MSs in $V_1, V_2, ..., V_{Q \times M}$, an optimized codebook $L$, and the expected number of transmissions $T_Q$.

**Pseudocode for technique 2**

\[
V = \{z_1, z_2, ..., z_K\}
\]

\[
V_1 = V_2 = ... = V_{Q \times M} = \emptyset \text{ (empty set)}
\]

begin

Run Algorithm 2

\{determine the best beam for each MS\}

Run Algorithm 3

\{further minimize the number of transmissions\}

end

**return** a set of MSs in $V_1, V_2, ..., V_{Q \times M}$, an optimized codebook $L$, and the expected number of transmissions $T_Q$

Algorithm 2 is described as follows: In step 1, $V$ represents the set of all $K$ MSs, and $Q \times M$ represents the number of all possible beams in a codebook. $V_i$ is the set
Algorithm 2 Codebook-based Allocation Algorithm

**step 1:** Initialization

\[ V = \{z_1, z_2, ..., z_K\} \]
\[ V_1 = V_2 = ... = V_{Q \times M} = \emptyset \text{ (empty set)} \]

**step 2:** For each MS \( z_j \) in \( V \), compute \( SINR_{ij} \) over all possible \( Q \times M \) beams. Then, assign MS \( z_j \) to beam \( i^* \) that maximizes \( SINR_{ij} \) such that

\[ i^* = \text{arg max}_i \{ SINR_{ij} \} \]

(Note that if MS \( z_j \) is assigned to beam \( i^* \) in set \( l_q \) of a codebook, the interference will come from all other beams in set \( l_q \), except beam \( i^* \).)

**step 3:** Update

\[ V = V - \{z_j\} \]
\[ V_{i^*} = V_{i^*} \cup \{z_j\} \]

**step 4:** If \( V \neq \emptyset \) then go to step 2. Otherwise, the algorithm terminates and returns a set of MSs in \( V_1, V_2, ..., V_{Q \times M} \).

of all assigned MSs in beam \( i \), and is initially empty where \( i \in \{1, 2, ..., Q \times M\} \).

In step 2, we select beam \( i^* \) for MS \( z_j \) whose SINR is maximized (the best assigned beam) over all possible beams. In step 3, we update the value of \( V \) and \( V_{i^*} \). Finally in step 4, if \( V \) is not an empty set, then go to step 2. Otherwise, the algorithm terminates, and returns a set of MSs in \( V_1, V_2, ..., V_{Q \times M} \). If some set of MSs (e.g., \( V_i \) for beam \( i \)) is an empty set, this implies that the BS will not transmit any packets over that beam whose power allocation is zero. Note that Algorithm 2 minimizes the variance of SINRs of all MSs in each beam.

Using all \( Q \) sets, a codebook may not be the best solution. In other words, using a large number of sets of a codebook implies that the BS spends an amount of time to successfully retransmit all lost packets while all MSs most likely have high SINR levels. However, when we lower SINR values for a certain set of MSs by eliminating some sets of a codebook and assigning these MSs to the remaining
sets, the overall number of transmissions can be reduced. Therefore, we propose an algorithm to further minimize the number of transmissions as shown next in Algorithm 3.

**Algorithm 3** Optimized Codebook Algorithm

**step 1:** Initialization

\[ L = \{l_1, l_2, \ldots, l_Q\} \]

Given \( V_1, V_2, \ldots, V_{Q \times M} \)

compute \( T_Q \)

**step 2:** For each set \( l_q \) of a codebook where \( q \in \{1, 2, \ldots, Q\} \), compute the maximum value of PERs over all MSs assigned to a set \( l_q \) such that

\[ X_{l_q} = \max_i \left\{ \max_j \{ \text{PER}_{ij} \} \right\}_{l_q} \]

**step 3:** Eliminate a set \( l'_q \) over all existing sets such that

\[ i_q = \arg \min_{l_q} \left\{ X_{l_q} \right\} \]

**step 4:** For each MS \( z_j \) in a set \( l'_q \), compute \( \text{SINR}_{ij} \) over all existing beams \( i \) except all beams in a set \( l'_q \). Then, assign MS \( z_j \) to beam \( i^* \) that maximizes \( \text{SINR}_{ij} \) such that

\[ i^* = \arg \max_i \left\{ \text{SINR}_{ij} \right\} \]

**step 5:** Compute \( T_{Q-1} \)

**step 6:** If \( T_{Q-1} \leq T_Q \),

update \( V_1, V_2, \ldots, V_{(Q-1) \times M} \)

\[ L = L - \{l'_q\} \]

\[ Q = Q - 1 \]

then go to step 2

**step 7:** The algorithm terminates, then returns a set of MSs in \( V_1, V_2, \ldots, V_{Q \times M} \), an optimized codebook \( L \), and the expected number of transmissions \( T_Q \).

Algorithm 3 works as follows: In step 1, we have the original sets of a codebook \( L = \{l_1, l_2, \ldots, l_Q\} \) and the groups of MSs in each \( V_1, V_2, \ldots, V_{Q \times M} \). Then, we compute the expected number of transmissions \( T_Q \) where \( Q \) is the number of sets of a codebook. In step 2, we determine the maximum PER over all MSs assigned
Figure 4.6: Broadcast transmission/retransmission schemes.

to each set $l_q$ of a codebook denoted by $X_{l_q}$. The best set of a codebook ($l^*_q$); i.e., the set that has the lowest value of the maximum PERs among all sets of a codebook is determined in step 3. Because the set with the highest value of maximum PERs dominates the transmission performance in the system, when we eliminate the set $l^*_q$, the number of transmissions is likely to decrease. Then, we assign all MSs in the set $l^*_q$ to all other existing sets as done in step 4. In step 5, we compute the expected number of transmissions $T_{Q-1}$ after eliminating the set $l^*_q$. In step 6, if $T_{Q-1} \leq T_Q$, then we go to step 2 for further codebook optimization. Otherwise, the algorithm returns the optimized codebook together with a set of MSs in $V_1, V_2, ..., V_{Q \times M}$, and the expected number of transmissions $T_Q$ as described in step 7.

4.6 Performance Evaluation

We consider a downlink channel consisting of one BS and multiple MSs. The BS and each MS are respectively equipped with 4-transmit antenna elements and a 1-receive antenna element. The system parameters are set as follows: maximum
In broadcast scenarios, the BS transmits a packet to all MSs at once. One example of broadcast transmissions is digital TV broadcasting, where all MSs watch the same TV channel at the same time. Therefore, the BS needs only to transmit data packets omni-directionally, which makes them reach all MSs. We assume that the BS uses one antenna to do so, and there is no need for applying BF techniques during data transmissions. For the purpose of evaluating the performance of our proposed scheduling algorithms, we consider evaluating four retransmission schemes: 1-Omni ARQ (i.e., traditional retransmission technique); 1-Omni NC (i.e., traditional with NC only); BF-NC with SINR (proposed technique 1: greedy selection-based scheduler), and BF-NC with codebook (proposed technique 2: codebook-based scheduler). Both 1-Omni ARQ and 1-Omni NC based retrans-
mission schemes use one transmit antenna element, and retransmissions of lost packets are respectively done with ARQ and NC techniques only. The proposed BF-NC based retransmission schemes use either technique 1 (Algorithm 1) or technique 2 (Algorithms 2 and 3). A graphical illustration of these schemes is given in Fig. 4.6.

4.6.1 Uniform MS distribution

First, we study the broadcast scenario shown in Fig. 4.7(a), where there are 120 uniformly distributed MSs, all having the same channel gain. Recall that the BS uses one transmit antenna during the data transmission phase, making signals/packets (transmitted omni-directionally) reach all MSs. In Fig. 4.8(a), we show the average loss rate of packet transmissions during these omni-directional transmissions as a function of the transmit power. Figs. 4.8(b) and (c) show the total number of transmitted packets and throughput gain, respectively, as a function of power for all simulated retransmission schemes. Fig. 4.8(b) shows that 1-Omni NC based retransmission scheme performs the best in this network scenario when compared with all the other retransmission schemes, followed by the BF-NC with codebook, the BF-NC with SINR, and then the 1-Omni ARQ. The achieved performances of the 1-Omni NC scheme are expected in this network topology, since all MSs experience the same channel gain and signal’s strength. We will see later that this is no longer the case when considering more realistic scenarios where MSs may experience different gains. For the BF-NC with SINR
Figure 4.8: Simulation results for network topology scenario of Fig. 4.7(a) with 120 uniformly distributed MSs.
(i.e., with greedy selection) scheme, SINR thresholds are set to 3 levels (6.75, 7.25, and 8.00) for grouping MSs in each beam. For completeness and to assess the impact of incorporating the NC technique, we also measure and plot in Fig. 4.8(c) the throughput gains of the 1-Omni NC, the BF-NC with SINR, and the BF-NC with codebook over the 1-Omni ARQ based retransmission scheme. In all figures, a throughput gain of ratio 1 means that both retransmission schemes use the same number of total transmissions. The figure shows that the 1-Omni NC and the BF-NC based retransmission schemes reach throughput gains of ratio 2.2 (i.e., 120% gain) and 1.5 (i.e., 50% gain), respectively. Even though both of the proposed BF-NC based retransmission schemes have similar results, we lean towards the codebook based one, since it does not require multiple levels of SINR thresholds while still achieving similar performances.

4.6.2 Clustered MS distribution

In this section, we consider studying a network topology with a clustered MS distribution. Specifically, we consider the network topology scenario of Fig. 4.7(b), where 60 MSs are grouped into 3-clusters each having 20 MSs. All MSs in each cluster are close to one another, and the channel gains for all MSs belonging to the same cluster are the same (or close). Fig. 4.9(a) shows the average loss rate of omni-directional packet transmissions (without including retransmissions) as function of the transmit power. In Fig. 4.9(b), we show the number of total transmissions (including retransmissions) under each of the studied retransmission
Figure 4.9: Simulation results for network topology scenario of Fig. 4.7(b) with 60 MSs formed into 3-clusters.
schemes as function of the transmit power. First, observe that unlike the case when MSs are uniformly distributed (as illustrated in Subsection 4.6.1), when MSs are not uniformly distributed, the proposed retransmission schemes (both BF-NC with SINR as well as BF-NC with codebook) outperform not only the traditional retransmission scheme (i.e., 1-Omni ARQ), but also the traditional with NC (i.e., 1-Omni NC scheme). The total number of transmissions needed under the proposed schemes is substantially lesser than those needed under the 1-Omni ARQ and the 1-Omni NC, especially for low transmit power levels. This is because the BS can now efficiently allocate the power to a set of active BF beams instead of spreading the power in all directions, strengthening then the signals received at intended MSs, and as a consequence, increasing MSs’ SINRs. The second observation that we can draw from the figure is that both proposed techniques achieve very similar performances in terms of the total number of transmissions. This is because when all MSs in the same cluster have similar channel gain, each coded packet most likely comes from all MSs whose packet loss rates are similar. In other words, this maximizes the number of lost packets to be coded at once.

We now plot the throughput gains of the proposed schemes and the 1-Omni NC scheme when compared with the traditional scheme (i.e., 1-Omni ARQ). As shown in Fig. 4.9(c), the throughput gains for the 1-Omni NC and the BF-NC based retransmission schemes can reach up to 120% (ratio of 2.2) and 175% (ratio of 2.75), respectively. Note that the incorporation of NC into the 1-Omni scheme results in almost halving the number of total transmissions. Whereas, the incorporation of these same NC techniques into BF based schemes (whether SINR or codebook
based) reduces the number of total transmissions by almost a third.

4.6.3 Impact of number of MSs

In this section, we want to study the impact of the number of MSs on the performance behaviors. In this simulation, we use the network topology of Fig. 4.7(a), where the number of MSs is varied from 120 to 300. In all scenarios, all MSs have the same channel gains. Fig. 4.10(a) shows how the number of transmissions changes as the number of MSs is varied for each of the proposed retransmission schemes. As expected, when increasing the number of MSs, the BS tends to spend more time to successfully retransmit all lost packets to all corresponding MSs. The number of transmissions when using the BF-NC with codebook retransmission scheme is less than that of the BF-NC with greedy selection (with SINR) retransmission scheme. Note that the number of MSs affects the throughput gain of both BF-NC based retransmission schemes; this is shown in Fig. 4.10(b). The higher the number of MSs, the higher the chances of combining more lost packets before retransmitting, thus leading to higher throughput gains (lesser transmissions).

4.7 Conclusion

This paper proposes two joint BF-NC based scheduling algorithms for wireless broadcast downlink channels. The first algorithm relies on greedy selection of MSs
Figure 4.10: Simulation results for network topology scenario of Fig. 4.7(a) when varying the number of MSs.
based on their SINRs, whereas the second algorithm uses codebook to group and schedule MSs. The greedy selection approach consists of using MSs’ orthogonality and their physical locations to maximize their signal quality over a wireless network. The codebook-based approach, on the other hand, uses a predefined codebook to generate multiple sets of orthogonal beams. Our results show that both of the proposed techniques/algorithms increase the overall throughput by minimizing the number of transmission attempts. Our simulations indicate that our techniques reduce the number of retransmission attempts significantly, resulting in achieving high throughput gain of up to 175% of broadcast transmissions when compared to that of traditional ARQ retransmission approaches.
Chapter 5 – Conclusions and Future Work

5.1 Contributions

In this thesis, we mainly focus on protocol design and optimization for wireless networks to maximize either throughput or QoS quality (e.g. video quality). We categorize our researches into 3 major contributions.

First of all, we propose a protocol for single-hop wireless networks where everyone can listen to and/or cache the packet belonging to each other by using a contention-based access MAC protocol as shown in Chapter 2. In particular, we provide an admission control framework in wireless home networks to enhance the quality of video streaming applications via a joint optimization of video layer-allocation technique, admission control algorithm, and MAC protocol. More specifically, instead of maximizing the throughput, we view our framework as a cross-layer optimization problem to maximize the averaged quality of admitted videos, given a specified minimum video quality for each flow. We then propose some heuristic algorithms to solve this problem with a good solution. We have extensive results and we show that a simple greedy layer-allocation algorithm performs well although it is typically not optimal. We also provide a more expensive heuristic algorithm achieving the result close to the optimal solution within a constant factor. To be complete, we theoretically analyze the computational complexity of our
problem under various conditions. We prove that when one of considered assumptions does not hold, the class of problems is NP-complete. Based on our scenario, the simulation results demonstrate that our proposed framework can improve the video quality up to 26% compared to those of existing approaches. This work has been published in [66] and [67].

Second contribution of our work is to provide a QoS guarantee in multi-hop wireless ad hoc networks as described in Chapter 3. Note that determining the feasibility of a given set of flow characteristics is hard due to the effect of multi-user interference. In particular, we first describe a simple Aloha-like MAC protocol to maintain the requested bandwidth for each flow. Thus, this protocol is obviously suitable for multimedia applications. We then propose a bandwidth feasibility algorithm based on Variable Elimination technique to determine whether or not a given network can support a set of flows with certain bit rates. Unlike other existing protocols, our proposed framework not only guarantees the required bandwidth but also precisely controls the achievable throughput for every considered host within 1% deviation from the requested requirements. This work has been published in [68].

Last contribution is to provide a framework for a large-scale wireless network in downlink channels where the transmission is made from the Base Station (BS) to Mobile Stations (MSs) as shown in Chapter 4. In particular, we consider in broadcast transmissions where one transmission is intended for everyone in the system. Specially, we consider a BS equipped with multiple antennas serving multiple MSs simultaneously by generating multiple signal beams. Formally, we use a beamform-
ing (BF) technique to provide multiple transmitted beams into certain directions to achieve high signal quality at intended MSs. When the number of MSs is large, the BS must decide which groups of MSs it should transmit the packets to at any transmission opportunity in such a way that interference among MSs is minimized resulting in maximizing the overall throughput. To further improve the achievable throughput, we also apply a network coding (NC) technique to reduce the number of retransmissions by encoding multiple lost packets into one combined packet then transmitting it to a set of intended MSs. Therefore, in this proposed framework, we develop two heuristic scheduling algorithms that exploit NC and BF techniques jointly together to achieve high overall throughput. The first algorithm relies on greedy selection of MSs based on their SINRs, orthogonality, and their physical locations to maximize their signal quality. The second algorithm is a codebook-based scheduling algorithm which uses a predefined codebook to generate multiple sets of orthogonal beams. The simulations indicate that our techniques reduce the number of retransmission attempts significantly. Especially, we can achieve high throughput gain up to 175% of broadcast transmissions when compared to that of traditional ARQ retransmission approaches. Finally, we also prove that the group formation problem is NP-hard. Some of this work has been partially published in [64].
5.2 Future Work

Due to high demands of multimedia applications and a large number of MSs under limited bandwidth constraints, it is challenge to investigate the technique to guarantee QoS quality of multimedia applications while providing services to multiple MSs simultaneously.

First of all, we would like to present research interests in a cross-layer optimization problem in wireless downlink channels for both broadcast and unicast transmissions. For instance, instead of guaranteeing and achieving high throughput over a joint NC and BF techniques, we provide a framework to maximize the overall QoS quality for all MSs. For more realistic scenario, multimedia server is not typically located at the BS close to intended MSs. Therefore, we need to take into account the delay and the fluctuation in network traffic between the source (i.e. multimedia server) and the destination (i.e. intended MS). In this case, we would like to propose a novel routing protocol together with joint NC and BF techniques that fully utilize network resources to maximize the overall QoS quality while avoiding the congestion in the system.
Bibliography


[16] *IEEE 802.11e: Wireless LAN medium access control (MAC) and physical later (PHY) specifications*, IEEE Std., 2005.


APPENDICES
A.1 Approximation Bound of Double-Greedy Algorithms

In this section, we show that by using double-greedy algorithms, one can guarantee an approximation factor of $0.5(1 - \frac{1}{e})$ from the optimal solution. Our proof requires minor modification to the proof provided in [43]. Throughout the rest of this section, we use the following notations:

- $S = \{x_1, ..., x_n\}$ is a set of $n$ elements;
- each element $x_i$ has an associated cost $c_i$;
- $G_i = \{x_1, ..., x_i\}$ is the solution from a modified greedy algorithm such that $G_i \subseteq S$;
- $W(G_i)$ is the weight of a set $G_i$;
- $C(G_i)$ is the cost for a set $G_i$;
- $L$ is the cost budget;
- $W'_i$ is the weight covered by a set $G_i$, not by $G_{i-1}$ such that $W'_i = W(G_i) - W(G_{i-1})$; and $OPT$ is the optimal set. For generality of analysis, the increase in cost when adding an element $x_i$ into a set $G_{i-1}$ can be computed such that $c_i = C(G_i) - C(G_{i-1})$ where $C(G_i) = C(G_{i-1} \cup \{x_i\})$.

For element $x_i$ in $OPT \setminus G_{i-1}$, the ratio of the weight to the cost is at most $\frac{W'_i}{c_i}$. The total cost of $OPT \setminus G_{i-1}$ is bounded by $L$. That is:

$$W(OPT) - W(G_{i-1}) \leq L \cdot \frac{W'_i}{c_i} \quad (A.1)$$

From $W'_i = W(G_i) - W(G_{i-1})$, we can rewrite (A.1) as follows:

$$W(G_i) - W(G_{i-1}) \geq \frac{c_i}{L} \cdot [W(OPT) - W(G_{i-1})] \quad (A.2)$$

We know that:
\[ W(G_i) = W(G_{i-1}) + [W(G_i) - W(G_{i-1})] \]
\[ \geq W(G_{i-1}) + \frac{c_i}{L} \cdot [W(OPT) - W(G_{i-1})] \]
\[ = \left( 1 - \frac{c_i}{L} \right) \cdot W(G_{i-1}) + \frac{c_i}{L} \cdot W(OPT) \quad (A.3) \]

By induction hypothesis, assume \( W(G_{i-1}) \) can be derived such that:
\[ W(G_{i-1}) \geq \left[ 1 - \prod_{k=1}^{i-1} \left( 1 - \frac{c_k}{L} \right) \right] \cdot W(OPT) \quad (A.4) \]

Substitute \( W(G_{i-1}) \) in (A.3) with (A.4), then we have:
\[ W(G_i) \geq \left( 1 - \frac{c_i}{L} \right) \cdot \left[ 1 - \prod_{k=1}^{i-1} \left( 1 - \frac{c_k}{L} \right) \right] \cdot W(OPT) + \frac{c_i}{L} \cdot W(OPT) \]
\[ = \left[ 1 - \prod_{k=1}^{i} \left( 1 - \frac{c_k}{L} \right) \right] \cdot W(OPT) \quad (A.5) \]

Note that the above derivation will be used to determine an approximation bound of double-greedy algorithms. At this point, we introduce three lemmas:

**Lemma 1** \( W(G_i) \geq \left[ 1 - \prod_{k=1}^{i} \left( 1 - \frac{c_k}{L} \right) \right] \cdot W(OPT) \)

**Lemma 2** For \( a_1, ..., a_n \in \mathbb{R}^+ \) and \( \sum_{i=1}^{n} a_i = A \), \( \left[ 1 - \prod_{i=1}^{n} \left( 1 - \frac{a_i}{A} \right) \right] \) is minimized if \( a_1 = ... = a_n = \frac{A}{n} \).
Lemma 3 From \( \lim_{n \to \infty} \left(\frac{1}{n}\right)^n = \frac{1}{e} \), then \( (1 - \frac{1}{n})^n \geq 1 - \frac{1}{e} \) where \( n < \infty \) and \( n \in \mathbb{R}^+ \).

Assume the number of iterations until getting the solution from a modified greedy algorithm is \( l \). That means the solution from a modified greedy algorithm will be a set \( G_l \). This implies that the final cost for a set \( G_l \) is \( C(G_l) \). Adding an element \( x_{l+1} \) into \( G_l \) will violate the cost budget \( L \). That is:

\[
C(G_{l+1}) = C(G_l) + c_{l+1}
\]

\[
> L \quad \text{(A.6)}
\]

Refer to Lemma 1, we can determine \( W(G_{l+1}) \) as follows:

\[
W(G_{l+1}) \geq \left[ 1 - \prod_{k=1}^{l+1} \left(1 - \frac{c_k}{L}\right) \right] \cdot W(OPT)
\]

\[
\geq \left[ 1 - \prod_{k=1}^{l+1} \left(1 - \frac{c_k}{C(G_{l+1})}\right) \right] \cdot W(OPT) \quad \text{(A.7)}
\]

Refer to Lemmas 2 and 3, assume we consider unity cost where \( c_1 = ... = c_{l+1} = 1 \). Then we have:
\[ W(G_{l+1}) \geq \left[ 1 - \left( 1 - \frac{1}{l+1} \right)^{l+1} \right] \cdot W(OPT) \]
\[ \geq \left( 1 - \frac{1}{e} \right) \cdot W(OPT) \quad (A.8) \]

However, the weight \( W(G_{l+1}) \) can be written as \( W(G_{l+1}) = W(G_l) + W(\{x_{l+1}\}) \). Assume original greedy algorithm returns the result with the weight \( W_{Original} \). If we substitute \( W(\{x_{l+1}\}) \) with \( W_{Original} \), then we can rewrite (A.8) as follows:

\[ W(G_l) + W_{Original} \geq \left( 1 - \frac{1}{e} \right) \cdot W(OPT) \quad (A.9) \]

This implies that at least one of \( W(G_l) \) and \( W_{Original} \) is greater than or equal to \( 0.5(1 - \frac{1}{e}) \cdot W(OPT) \). \( \square \)
A.2 Approximation Bound of Triple-Greedy Algorithms

In this section, we show that by using triple-greedy algorithms, one can guarantee an approximation factor of \((1 - \frac{1}{e})\) from the optimal solution. Recall that triple-greedy algorithm is a modified greedy algorithm with enumerating all triples from all unselected elements. For generality of analysis, we initially use a modified greedy algorithm to select a set of \(k\) elements in each iteration. Throughout the rest of this section, we use the following notations: \(S = \{x_1, ..., x_n\}\) is a set of \(n\) elements; \(G_i = \{x_1, ..., x_i\}\) is the solution from a modified greedy algorithm such that \(G_i \subseteq S\); \(W(G_i)\) is the weight of a set \(G_i\); \(L\) is the cost budget; and \(OPT\) is the optimal set.

Let \(Y_1\) be the first set of \(k\) selected elements. This implies that a set \(Y_1\) has the highest weight of \(W(Y_1)\). Let \(l\) be the number of iterations until getting the solution by using a modified greedy algorithm with the selection of \(k\) elements for each iteration. Let \(G_l = Y_1 \cup Y'\) where \(Y_1, Y' \subseteq G_l\) and \(Y_1 \cap Y' = \emptyset\). That is \(W(G_l) = W(Y_1) + W(Y')\). Furthermore, adding an element \(x_{l+1}\) into a set \(G_l\) will violate the cost budget \(L\). That is:

\[
W(G_{l+1}) = W(G_l) + W(\{x_{l+1}\})
= W(Y_1) + W(Y') + W(\{x_{l+1}\})
\geq (1 - \frac{1}{e}) \cdot W(OPT) \quad (A.10)
\]
Note that inequality in (A.10) comes from the analysis of double-greedy algorithms in (A.8). Without inclusion of a set $Y_1$, we may rewrite (A.10) as follows:

$$W(Y') \geq \left(1 - \frac{1}{e}\right) \cdot W(OPT \setminus Y_1) - W(\{x_{l+1}\}) \quad (A.11)$$

That is:

$$W(G_l) = W(Y_1) + W(Y')$$

$$\geq W(Y_1) + \left(1 - \frac{1}{e}\right) \cdot W(OPT \setminus Y_1) - W(\{x_{l+1}\})$$

$$\geq W(Y_1) + \left(1 - \frac{1}{e}\right) \cdot W(OPT \setminus Y_1) - \frac{W(Y_1)}{k}$$

$$\geq \left(1 - \frac{1}{k}\right) \cdot W(Y_1) + \left(1 - \frac{1}{e}\right) \cdot W(OPT \setminus Y_1)$$

$$\geq \left(1 - \frac{1}{e}\right) \cdot W(OPT) \quad (A.12)$$

We note again that the second inequality in (A.12) comes from the fact that $\frac{W(Y_1)}{k} \geq W(\{x_{l+1}\})$. This is because all $k$ elements in a set $Y_1$ have the most highest weight based on greedily selecting fashions. We know that $W(Y_1) + W(OPT \setminus Y_1) = W(OPT)$. When $k \geq 3$, the third inequality in (A.12) can be reduced to $W(G_l) \geq (1 - \frac{1}{e}) \cdot W(OPT)$. That means by using triple-greedy algorithms ($k = 3$), we can guarantee the result with an approximation factor of $(1 - \frac{1}{e})$ from the optimal solution. □
A.3 Hardness Results of a BF-NC Scheduling Problem

In this section, we prove that our BF-NC scheduling problem is NP-hard. First of all, we know that the well-known problem “Scheduling Unrelated Parallel Machines (SUPM)” is NP-hard [69]. To prove that our BF-NC based scheduling problem is NP-hard, we need to show the reduction from the SUPM problem to our problem. In the SUPM problem, there is a set of $K'$ jobs denoted by $J = \{p_1, p_2, ..., p_{K'}\}$ and a set of $M$ parallel machines with their related speed factors denoted by $S = \{s_1, s_2, ..., s_M\}$. For generalization of identical machines, they can run at different speeds but do so uniformly. For each machine $i$ with a speed factor $s_i$, the processing time for job $j$ ($p_j$) on machine $i$ denoted by $p_{ij}$ can be computed as $p_{ij} = p_j / s_i$. Note that each machine $i$ can process only one job at a time. Therefore, the total time on machine $i$ is $\sum_j p_{ij}$. The goal is to find a scheduler that minimizes the makespan such that:

Figure A.1: Illustration of the reduction from an SUPM problem to a BF-NC based scheduling problem.
\[
\text{minimize} \quad \max_i \left\{ \sum_j x(i, j)p_{ij} \right\}
\]
\[
\text{s.t.} \quad \sum_i x(i, j) = 1
\]
\[
x(i, j) \in \{0, 1\}
\]
(A.13)

where \(i \in \{1, 2, ..., M\}\) and \(j \in \{1, 2, ..., K'\}\). Formally, an instance of an SUPM problem is a 3-tuple as follows:

\[
\left\{ \{p_1, p_2, ..., p_{K'}\}, \{s_1, s_2, ..., s_M\}, \{x(1, 1), ..., x(M, K')\} \right\}
\]

Given an instance of an SUPM problem, we can construct the following version of our BF-NC based scheduling problem by reduction as follows. All MSs are in a set \(V\). A set of lost packets belongs to MS \(z_k\) is \(p_k\). In particular, the job \(p_k\) in original problem is reducible to our problem as a set of lost packets \(n_k\) for MS \(z_k\) (i.e. \(n_k \leftarrow p_k\)). For each beam \(i\), we randomly assign MS \(z_k\) with a set of lost packets \(p_k\) to a group of MSs \(V_i\) over all \(M\) groups as shown in Fig. A.1 where \(V = \cup_{\forall i} V_i\) and \(i \in \{1, 2, ..., M\}\). In this case, we set \(x(i, k) = 1\) for assigning a set of lost packets \(p_k\) to a group \(V_i\) (for beam \(i\)). Note that this reduction is the special case of our BF-NC based scheduling problem when all lost packets in each group cannot be combined. Therefore, the reduction for variables \(p_k\) to \(n_k\) can be done in \(O(1)\).

Suppose we have the optimal solution to solve a BF-NC based scheduling problem. To be complete, we can show that the optimal solution to our BF-NC based scheduling problem is the optimal solution to an SUPM problem as follows. First,
we use only one set of orthogonal BF beams from a given codebook. For example, the BS uses only the $q^{th}$ set, then we set $l_q = 1$ and $l_p = 0$ in Eq. (4.6) where $p \neq q$. Therefore, we have:

$$\text{minimize } \max_i \left\{ \frac{N_{Vi}}{1 - \max_j \{x(i,j)PER_{ij}\}} \right\}$$

s.t. $$\sum_i x(i,j) = 1$$

$$x(i,j) \in \{0,1\}$$

where $N_{Vi}$ and $PER_{ij}$ are a set of coded/uncoded lost packets based on an NC technique in beam $i$ and packet error rate for MS $z_j$ in beam $i$, respectively. Recall that all lost packets in each beam $i$ cannot be combined in this special case. Thus, a set of lost packets $n_j$ for MS $z_j$ strictly depends on $PER_{ij}$ where $n_j \subseteq N_{Vi}$. In such case, we can rewrite Eq. (A.14) as follows:

$$\text{minimize } \max_i \left\{ \sum_j x(i,j)\frac{n_j}{1 - PER_{ij}} \right\}$$

s.t. $$\sum_i x(i,j) = 1$$

$$x(i,j) \in \{0,1\}$$

In Eq. (A.15), the time to successfully retransmit all lost packets in $N_{Vi}$ is the summation of the time to retransmit all $n_j$ where $n_j \subseteq N_{Vi}$.

Comparing Eq. (A.13) to Eq. (A.15), and assuming there exists a function for reduction from $p_{ij} = p_j/s_i$ to $\frac{n_j}{1 - PER_{ij}}$, all instances of an SUPM problem are correctly reducible to our BF-NC based scheduling problem. That is:
\[
\text{minimize } \max_i \left\{ \sum_j x(i,j) \frac{p_j}{s_i} \right\}
\]
\[
\Leftrightarrow \text{minimize } \max_i \left\{ \sum_j x(i,j) \frac{n_j}{1 - \text{PER}_{ij}} \right\} \quad (A.16)
\]

The solution to Eq. (A.15) is a set of \(x(i,j)\) for all \(i\) and \(j\). From Eq. (A.16), both objective functions can be reducible to each other and vice versa. That means the optimal solution to our BF-NC based scheduling problem in Eq. (A.15) is the optimal solution to an SUPM problem in Eq. (A.13). Furthermore, the SUPM problem can be reducible to our problem in polynomial time. This concludes that our problem is NP-hard. \(\square\)