

Copyright  
by  
Ahmed Mohamed Abdel Hadi  
2011

The Dissertation Committee for Ahmed Mohamed Abdel Hadi certifies that this is the approved version of the following dissertation:

**Multicast Networks: Capacity, Algorithms, and Implementation**

Committee:

---

Sriram Vishwanath, Supervisor

---

Andreas Gerstlauer, Supervisor

---

Maruthi Akella

---

Alan Bovik

---

Jonathan Valvano

---

Haris Vikalo

**Multicast Networks: Capacity, Algorithms, and  
Implementation**

by

**Ahmed Mohamed Abdel Hadi, B.S.E.E.; M.S.E.; M.S.E Phy & Math**

**DISSERTATION**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2011

Dedicated to my family.

## Acknowledgments

The most important acknowledge is to our God the Most Merciful, the Most Wise by whose mercy we were able to begin this work, His Mercy is such that we are given the ability to work in His cause through which we remember Him Swt and be grateful towards all He has given us. Allah states in the *Quran* "and say: My Lord! Increase me in knowledge" [Quran, 20:114]. May Allah accept our humble work as a effort to increase our knowledge. Ameen

May Allah's peace and blessing be upon our Beloved Prophet Muhammad who was a mercy unto us from Allah Swt. Prophet Muhammad Pbuh. mentioned in the *Hadith* that "Seeking Knowledge is mandatory for every believer" [Authenticated by: Ibn Majah]. This work is a trial to follow his footsteps.

I would like to thank my advisors Prof. Sriram Vishwanath and Prof. Andreas Gerstlauer for their invaluable guidance and encouragement. I am greatly indebted to them in every achievement at UT. Their guidance and support have made this dissertation possible. I also thank my dissertation committee members, Prof. Maruthi Akella, Prof. Alan Bovik, Prof. Jonathan Valvano, and Prof. Haris Vikalo for their valuable advice and comments on my research and dissertation.

I would like to express my appreciation to my colleagues in University of

Texas (UT), especially in WNCG. I thank Jonas, Liang, Chris, Kumar, Jubin, Vidur, Abhik and Tony for their helpful discussions. I also thank aerospace professor Prof. Chaput for his help and support and aerospace students Marlan, Mark, and Sean for their help in the execution of the experimental part of my graduate work. I also thank WNCG staff Julie, Jennifer and Melanie for their efficient and kind help in my graduate study.

# Multicast Networks: Capacity, Algorithms, and Implementation

Publication No. \_\_\_\_\_

Ahmed Mohamed Abdel Hadi, Ph.D.  
The University of Texas at Austin, 2011

Supervisors: Sriram Vishwanath  
Andreas Gerstlauer

In this dissertation, we investigate the capacity and performance of wireless networks with an emphasis on multicast traffic. The defining characteristic of a multicast network is a network where a number of different destinations all require the information generated by a single source. The models that we explore differ in the nature of the nodes from all-mobile case where all nodes are mobile to hybrid case where some nodes are mobile and some are static. We investigate different performance measure for these wireless multicast networks: upper bounds, capacity scaling laws, and achievable rates. The understanding of these measures for such networks helps in the development of efficient algorithms for operating these networks.

In addition, we study the practical realization of algorithms for real-time streaming of rich multimedia content in the context of mobile wireless networks for embedded and cyberphysical systems. Our initial work is in the

context of unicast and multiple unicast systems over an autonomous aerial vehicle (AAV) network. Bandwidth requirements and stringent delay constraints of real-time video streaming, paired with limitations on computational complexity and power consumptions imposed by the underlying implementation platform, make cross-layer and cross-domain co-design approaches a necessity. In this dissertation, we propose a novel, low-complexity rate-distortion optimized (RDO) protocol specifically targeted at video streaming over mobile embedded networks. First, we test the performance of our RDO algorithm on simulation models developed for aerial mobility of multiple wirelessly communicating AAVs. Second, we test the performance of our RDO algorithm and other proposed adaptive algorithms on a real network of AAVs and present a comparative study between these different algorithms. Note that generalizing these algorithms to multicast settings is relatively straightforward and thus is not highlighted to a great degree in this thesis.



# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Contributions . . . . .	3
1.2.1 Multicast Capacity of Wireless Networks . . . . .	3
1.2.2 Multicast Interference Alignment of Wireless Networks .	4
1.2.3 Real-Time Rate-Distortion Optimized (RDO) Streaming of Wireless Video . . . . .	5
1.3 Organization . . . . .	6
<b>Chapter 2. Multicast Capacity of Wireless Networks</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.1.1 Motivation and Related Work . . . . .	7
2.1.2 Our Contributions . . . . .	9
2.2 System Model . . . . .	10
2.2.1 Interference Models . . . . .	12
2.2.2 Mobility Model . . . . .	13
2.3 Multicast Upper Bounds . . . . .	14
2.4 Multicast Throughput: Protocol Model . . . . .	14
2.4.1 All Nodes are Mobile . . . . .	15
2.4.2 Hybrid (Static and Mobile) Nodes . . . . .	20
2.5 Multicast Throughput: Physical Model . . . . .	22

<b>Chapter 3. Multicast Interference Alignment of Wireless Networks</b>	<b>25</b>
3.1 Introduction . . . . .	25
3.1.1 Motivation and Related Work . . . . .	25
3.1.2 Our Contributions . . . . .	26
3.2 System Model . . . . .	26
3.2.1 Network Description . . . . .	26
3.2.2 Problem Statement . . . . .	28
3.3 Multicast Alignment Strategies . . . . .	29
3.3.1 Overprovisioned Network . . . . .	30
3.3.2 Underprovisioned Network . . . . .	33
<b>Chapter 4. Real-Time Rate-Distortion Optimized Streaming of Wireless Video</b>	<b>35</b>
4.1 Introduction . . . . .	35
4.1.1 Motivation . . . . .	36
4.1.2 Related Work . . . . .	38
4.1.3 Our Contributions . . . . .	40
4.2 Rate-Distortion Optimization Problem . . . . .	41
4.3 Testbed . . . . .	43
4.3.1 System Simulation . . . . .	44
4.3.2 System Implementation . . . . .	44
4.4 Low Complexity RDO with ACKs (LCRDO-Ack) . . . . .	50
4.4.1 LCRDO-Ack Algorithm . . . . .	51
4.4.2 Distortion Measure . . . . .	54
4.4.3 Experimental Setup . . . . .	55
4.4.4 Results . . . . .	56
4.5 Low Complexity RDO with Beacons and Adaptivity (LCRDO-Beacon and LCRDO-Adaptive) . . . . .	61
4.5.1 LCRDO-Beacon Algorithm . . . . .	61
4.5.2 LCRDO-Adaptive Algorithm for MPEG2 . . . . .	65
4.5.3 LCRDO-Adaptive Algorithm for MJPEG/SMOKE . . . . .	68
4.5.4 Distortion Measures . . . . .	71
4.5.5 Experimental Setup . . . . .	73
4.5.6 Results . . . . .	78

<b>Chapter 5. Conclusion</b>	<b>85</b>
<b>Appendices</b>	<b>87</b>
<b>Appendix A. Proofs of Chapter 2</b>	<b>88</b>
A.1 Proof of Lemma 2.4.1 . . . . .	88
A.2 Proof of Lemma 2.4.2 . . . . .	88
A.3 Proof of Lemma 2.4.3 . . . . .	89
A.4 Proof of Lemma 2.4.4 . . . . .	90
A.5 Proof of Lemma 2.4.5 . . . . .	91
A.6 Proof of Lemma 2.5.1 . . . . .	91
A.7 Proof of Lemma 2.5.2 . . . . .	93
<b>Appendix B. Proofs for Chapter 3</b>	<b>95</b>
B.1 Proof of Lemma 3.3.1 . . . . .	95
B.2 Proof of Corollary 3.3.2 . . . . .	96
B.3 Proof of Theorem 3.3.3 . . . . .	96
B.4 Proof of Theorem 3.3.4 . . . . .	96
<b>Bibliography</b>	<b>98</b>
<b>Vita</b>	<b>105</b>

## List of Tables

4.1	Payload parameters . . . . .	57
4.2	Received video duration for temporal distortion . . . . .	79
4.3	Average SSIM indices for spatial distortion . . . . .	80
4.4	Comparison between proposed RDO algorithms . . . . .	84

## List of Figures

2.1	System model. . . . .	10
2.2	The first phase. . . . .	15
2.3	The second phase. . . . .	16
3.1	System model. . . . .	27
3.2	Desirable alignment scheme. . . . .	29
3.3	Overprovisioned network. . . . .	30
3.4	Underprovisioned network. . . . .	33
4.1	Horus project diagram. . . . .	45
4.2	Autonomous aerial vehicle. . . . .	46
4.3	AAV block diagram. . . . .	47
4.4	A picture of the components mounted inside the AAV. . . . .	48
4.5	Conceptual block diagram of the RDO algorithm. . . . .	51
4.6	Simulated network topology. . . . .	54
4.7	Received packet counter for a complete GOF successfully received. $f_1$ frame counts to 50, $f_2$ frame counts to 20 and $f_3$ frame counts to 10. Any packet reception less than that count is not complete and the frame is considered dropped in the distortion measure. . . . .	56
4.8	Received packet counter for No-ACK transmission. . . . .	58
4.9	Received packet counter for ACK transmission. . . . .	58
4.10	Received packet counter for LCRDO-Ack transmission. . . . .	59
4.11	Distortion measure $D_M$ for different transmission protocols. . . . .	60
4.12	Block diagram of the LCRDO-Beacon algorithm at the transmitter. . . . .	62
4.13	Block diagram of the LCRDO-Beacon algorithm at the receiver. . . . .	62
4.14	Hysteresis: Controller switches to Encoder 2 when RSSI is less than $X_1$ and to Encoder 1 when RSSI is greater than $X_2$ . No change is done when RSSI is between $X_1$ and $X_2$ . . . . .	63

4.15	Block diagram of the LCRDO-Adaptive algorithm for MPEG2 at the transmitter. . . . .	66
4.16	Block diagram of the LCRDO-Adaptive algorithm for MPEG2 at the receiver. . . . .	66
4.17	Block diagram of the LCRDO-Adaptive algorithm for MJPEG/SMOKE at the transmitter. . . . .	69
4.18	Block diagram of the LCRDO-Adaptive algorithm for MJPEG/SMOKE at the receiver. . . . .	69
4.19	Unicast network topology. . . . .	74
4.20	Multiple unicast network topology. . . . .	75
4.21	Map of the actual site used for running Horus experiments. The internal circle shows the AAV flight path. The outer circle shows the viewing area of the camera attached to the AAV. . . . .	76
4.22	Flight phases used to calculate average SSIM index. . . . .	77
4.23	Received video duration for temporal distortion. . . . .	80
4.24	Average SSIM indices for spatial distortion. . . . .	82
4.25	Samples of spatial frame/image distortion (frame from source video is on the right and frame from destination video is on the left). . . . .	83

# Chapter 1

## Introduction

### 1.1 Motivation

The last decade has seen an explosion in the types and capabilities of wireless devices deployed all over the world. The demand for the quantity and quality of data transfer via these devices is ever-growing. It is therefore important to understand the fundamental limits on the performance of such wireless systems, and design efficient algorithms and techniques to deliver performance close to the limits. Multicasting is a more efficient mechanism for supporting group communication than unicasting or broadcasting. It allows transmission and routing of packets to multiple destinations using fewer network resources. The interest in supporting multicast communications over wireless networks is increasing due to the widespread deployment of wireless networks, the more complex capabilities of mobile devices, and the worldwide mobile work force. Therefore, service providers are increasingly interested in deploying multicast applications on a large scale.

Wireless multicast is also highly relevant for group oriented applications. Some of these applications are mobile commerce and targeted advertisement, military command and control of units, distance education, and

intelligent transportation systems. Mobile commerce applications will gain significant benefit if group communication among mobile users is supported by wireless networks. For example, mobile auctions require connectivity, secure and reliable wireless multicast. In military, essential commands and information can be multicast to different military units which demand minimum delay on a secure and reliable wireless multicast. Distance education and entertainment services can be offered to mobile and remote users. Again, as an example, applications such as video streaming requires high bandwidth and real-time wireless multicast for quality viewing. Intelligent transportation systems involve the dynamic routing of individual vehicles. Traffic information, for example direct and least time-consuming routes, can be multicast to drivers.

The main goal of multicasting over wireless networks is to reduce bandwidth and power consumption without affecting the reliability of wireless communication. It also helps to better utilization of limited resources by achieving higher per user bandwidth over unicasting. As the wireless network is broadcast in nature, the packet is received by all nodes in the transmission range of the transmitter. This suggests that each packet is sent only once and so received by all destinations. While such an approach seems efficient at first, it can still be wasteful if other simultaneous transmissions or the limited transmit power of mobile devices are not properly considered. Another consideration is when not all the destinations are ready to receive. Either, the transmitter has to wait, and the system becomes unstable, or the transmitter can



send regardless of the destinations status and serious packet drops occur and cause degradation in the throughput. Overall, multicasting needs to consider the trade-off between throughput, stability, and packet loss accompanied by reduction in power consumption.

Multicast communications has been supported for years in the Internet environment for fixed users using wired links. But it is not yet deployed in wireless networks. This research is motivated by the fact that multicast over wireless networks is an important and challenging goal, but several issues must be addressed before many group applications can be deployed on a large scale.

## **1.2 Contributions**

The completed contributions of my research during my Ph.D. program are as follows.

### **1.2.1 Multicast Capacity of Wireless Networks**

The dissertation investigates the impact of mobility on the capacity scaling laws for wireless multicast networks. The following summarizes our main contributions [40]:

1. Analogous to the beneficial impact that mobility has on the throughput of unicast networks, this work establishes that mobility can provide a similar gain in the order-wise growth-rate of the throughput for multicast networks. This work considers an all-mobile multicast network for both

protocol and physical interference models, and characterizes the multicast capacity scaling for these scenarios. These scaling results show that the growth-rate of the throughput in the all-mobile multicast network is order-wise higher compared to the all-static multicast network.

2. The dissertation considers a static-mobile hybrid multicast network for the protocol interference model, and establishes that, even when mobility is limited to some nodes in the network, mobility can impact the scaling law of the total throughput. In particular, the dissertation shows that, if there are sufficient number of mobile nodes (but order-wise smaller than the total number of nodes) in the network, then mobile nodes can enhance the order behavior of the total multicast throughput.

### **1.2.2 Multicast Interference Alignment of Wireless Networks**

This dissertation investigates alignment schemes for multicast traffic over an equal path length multihop time-varying circularly symmetric fading channels. The following summarizes our main contributions [21]:

1. Determine achievable rates and alignment mechanisms for multi-hop communication for multicast networks when the number of relays exceeds the number of sources/destinations.
2. Determine the alignment strategies when number of relays is smaller than the number of sources/destinations by combining elements of the multi-hop and ergodic alignment.

### 1.2.3 Real-Time Rate-Distortion Optimized (RDO) Streaming of Wireless Video

The final aspect of this thesis is to understand optimized streaming in practice. The previous two sections describe theoretical limits that do not, in general, translate to any form of practical algorithms. Thus, in this section, we aim to bridge the gap by first developing algorithms for optimized streaming and then deploy them using a practical testbed.

The challenge for optimized streaming is addressed in this thesis by implementing unicast and multiple unicast systems, as no implementations currently exists for even such systems. Specifically, we make the following contributions:

1. A new low-complexity RDO algorithm [20] using default Media Access Control Layer (MAC-L) ACKs, which is validated through real-world simulations and shown to outperform ACKed and not ACKed transmission.
2. A RDO algorithm using MAC-L beacons in order to achieve optimized video streaming under even lower complexity, as validated through experiments on a real-world AAV testbed and shown to minimize drops.
3. A co-design of a RDO algorithm with adaptive video encoding to achieve optimized video streaming, developed for both MPEG2 and MJPEG streaming and validated through experiments on a real-world AAV testbed, where it is shown to improve received video quality.

Note that these algorithms and our results focus primarily on unicast and multiple unicast settings. Generalizing these algorithms for multicast and multiple multicast is relatively straightforward and therefore not the primary focus of this thesis.

### **1.3 Organization**

This dissertation is divided into three main topics namely (i) Capacity of multicast wireless networks, (ii) Interference Alignment of multicast wireless networks, and (iii) Implementation of rate-distortion optimized streaming of wireless video that are discussed respectively in Chapters 2, 3, and 4. This dissertation concludes in Chapter 5.

## Chapter 2

# Multicast Capacity of Wireless Networks

### 2.1 Introduction

#### 2.1.1 Motivation and Related Work

Capacity scaling characterizations for large-scale wireless networks under varying assumptions and constraints [34, 35, 46, 22] is now a fairly well-established field. As noted in the existing body of work, assumptions on the channel model (such as protocol, physical) and requirements (such as unicast and multicast) can greatly impact the overall order of growth of network throughput. Over the past decade, we have gained a fairly in-depth understanding of the capacity of unicast networks. In [35, 54], the authors present some of the early capacity results for unicast networks. For physical channel models, the authors of [22, 46] determine the impact of cooperation on network capacity. In particular, in [46], the authors show that a throughput of  $O(n)$  is achievable for unicast transmission over random networks using a hierarchical cooperation scheme. Such results have been generalized to arbitrary networks in [45]. More recently, models inspired by physics and electromagnetics have been introduced to determine capacity scaling results for unicast network in [32].

Although a majority of literature in the domain studies capacity scaling

laws for static networks, there is a growing body of work on mobile networks. Pioneering work on impact of unrestricted mobility on unicast capacity, where each node in the network is mobile is in [34]. In [33], the authors introduce new restricted mobility models to represent delay tolerant networks, and determine the impact of this restricted mobility on the capacity of unicast networks. Note that the existing body of work focuses primarily on unicast networks, but, as we show in this thesis, multicast network capacity is also impacted considerably by node mobility.

Multicast networks have been studied in literature in a variety of contexts [47, 42, 50, 53]. Capacity scaling laws for multicast networks (with static nodes) have been analyzed much more recently [49, 43, 27, 38]. In [49, 43], the authors show that the multicast requirement can significantly decrease the achievable throughput in a wireless network. In particular, for a multicast group per source of size  $\Theta(n^{1-\epsilon})$ , it is shown that a throughput of at most  $O(\sqrt{n^\epsilon})$  is sustainable. This throughput is obtained by constructing an (approximate) multicast tree from the source to the destinations in the network. This places a fairly stringent constraint on the number of possible multicast transmissions in a network. As there are multiple application settings in which multicasting plays an important role, it is useful to determine the impact that mobility can have on improving the performance of multicast communication in networks.

### 2.1.2 Our Contributions

We consider a dense<sup>1</sup> wireless network with  $n$  nodes placed randomly inside square area. The mobile nodes can move freely<sup>2</sup> on the entire area. There are  $n_s$  multicast sources and each source has a multicast destination group of size  $n_d$ .

- To ascertain the benefits of mobility to multicast networks, we first consider a network configuration where all  $n$  nodes are mobile. For this setting, we extend and generalize the results of [34] to multicast networks. We show that the total multicast throughput increases to  $\Theta(\min\{n_s, n/n_d\})$  from  $O(\min\{n_s, \sqrt{n/n_d}\})$  for the static case [49, 43]. Thus, we establish that the growth-rate of the throughput in the all-mobile multicast network is order-wise higher compared to the all-static multicast network.
- Next, we consider a hybrid setting where a subset  $n^\gamma$  ( $\gamma < 1$ ) nodes in the network are mobile while the remainder  $\Theta(n)$  nodes are static. In this setting, we show that the following total throughput is achievable:

$$\Omega \left( \min \left\{ n_s, \max \left\{ \frac{n^\gamma}{n_d \log n}, \sqrt{\frac{n}{n_d \log n}} \right\} \right\} \right). \quad (2.1)$$

---

<sup>1</sup>In this work, we do not consider extended networks where area scale with the number of nodes. We only consider dense networks where the area is fixed while the number of nodes scale.

<sup>2</sup>If the nodes are restricted to move in different but connected clusters, we expect degradation in the multicast throughput. But if the nodes are moving in a deterministic path, we expect improvement in the multicast throughput.

**Example 2.1.1.** Let  $n_d = \Theta(n^{1-\epsilon})$ . If  $\gamma \geq 1 - \frac{\epsilon}{2} + \delta$  for some  $\delta > 0$ , then we have  $n^{\gamma+\epsilon-1} \geq \sqrt{n^{\epsilon+2\delta}}$ . This ensures that the overall achievable throughput in (2.1) can increase with the number of mobile nodes in the network if the number of sources  $n_s$  were to grow as  $n^{\gamma+\epsilon-1}/\log n$ .

Thus, we establish that mobility enhances the order of growth of throughput of the hybrid network when the order of growth of mobile nodes in the network is greater than a threshold.

## 2.2 System Model

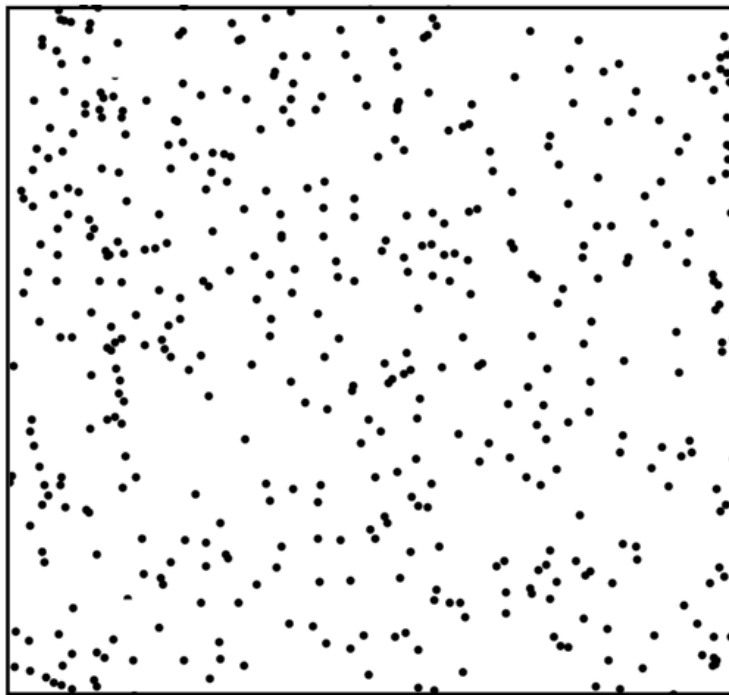


Figure 2.1: System model.



We consider a dense wireless network with  $n$  nodes inside a square  $A$  of unit area. This network contains two families of nodes - static and mobile. There are  $n^\gamma$  ( $0 \leq \gamma \leq 1$ ) mobile nodes, and the remaining nodes are static nodes. The mobile nodes can move freely on the entire area  $A$ . The locations of the mobile nodes  $X_i(t)$  are considered to be stationary and ergodic processes with uniform stationary distribution over the entire area  $A$ . The locations  $X_i(t)$  of the static nodes are fixed over time, and are uniformly and independently distributed over the entire area  $A$ . The trajectories of all nodes are independent.

The multicast scenario is modeled as follows: there are  $n_s$  nodes that are multicast sources, and each multicast source is accompanied by  $n_d$  destination nodes. Note that the number of destination nodes is the same for each source. We consider  $n_s = \Theta(n^\mu)$  and  $n_d = \Theta(n^\nu)$  with  $0 \leq \mu \leq 1$  and  $0 \leq \nu \leq 1$ . Since the case with constant  $n_s$  is straightforward, we consider  $n_s \rightarrow \infty$  as  $n \rightarrow \infty$ . The sources and destinations are selected uniformly over all traffic associations such the following conditions for uniform traffic among nodes are satisfied.

1. The  $n_s$  multicast sources are distinct.
2. For each multicast, the source node and destination nodes are distinct.
3. Each node in the wireless network is the destination to at most  $\lceil n_s n_d / n \rceil$  multicasts.

We use the same definitions of feasible throughput and throughput capacity (with destination replaced by the set of destination nodes) as defined in [35]. Next, we describe the two different interference models we use in this thesis the protocol model and the signal to interference and noise ratio (SINR) based physical model. These are based on models introduced in [35].

### 2.2.1 Interference Models

In both the interference models described below, if a transmission is successful,  $B$  bits of data per unit time can be transferred.  $X_i$  denotes the location of node  $i$ .

1. *Protocol Model:* All nodes choose a common radius of transmission  $r$ . At time  $t$ , a transmission from node  $i$  to node  $j$  is successful if the following two conditions are satisfied.

- (a) The receiver is within the transmission range of the transmitter, i.e.,

$$|X_i(t) - X_j(t)| \leq r. \quad (2.2)$$

- (b) Every other transmitter  $X_k(t)$  ( $k \neq i$ ) simultaneously transmitting in the same time slot  $t$  does not cause interference to node  $j$ , i.e.,

$$|X_k(t) - X_j(t)| \geq (1 + \Delta)r. \quad (2.3)$$

Here,  $\Delta > 0$  models a guard zone, and is a constant that does not depend on  $n$ .

2. *Physical Model:* At time  $t$ , the transmission from node  $i$  with power  $P_i(t)$  is successfully received at node  $j$  if the SINR satisfies

$$\frac{P_i(t)\eta_{ij}(t)}{N_0 + \sum_{k \neq i} P_k(t)\eta_{kj}(t)} > \zeta, \quad (2.4)$$

where  $N_0$  is the noise power (same at all nodes),  $\zeta$  is the minimum SINR (constant that does not depend on  $n$ ) for successful transmission, and  $\eta_{ij}(t)$  is the channel gain from node  $i$  to node  $j$ . The channel gain  $\eta_{ij}(t)$  depends on the distance between node  $i$  and node  $j$ , and is given by

$$\eta_{ij}(t) = |X_i(t) - X_j(t)|^{-\alpha_g}, \quad (2.5)$$

where  $\alpha_g \geq 2$  is the power decay parameter.

### 2.2.2 Mobility Model

In this model, the mobile nodes can move freely on the entire area  $A$ . Without loss of generality, we assume unit area. The locations of the mobile nodes  $X_i(t)$  are considered to be stationary and ergodic processes with uniform stationary distribution over the entire area  $A$ . The locations  $X_i(t)$  of the static nodes are fixed over time, and are uniformly and independently distributed over the entire area  $A$ . The trajectories of all nodes are independent.

We assume in the model that sources and destinations are randomly selected from the  $n$  nodes in the network. First,  $n_s$  sources are randomly selected from the  $n$  nodes. Then,  $n_d$  destinations for each source are randomly selected from the remaining nodes in the network, as a node can't be a source

and destination at the same time. However, any destination can act as a relay (intermediate node) for other sessions. We assume that each node has an infinite buffer and can carry temporarily as many packets of other sessions as required.

### 2.3 Multicast Upper Bounds

We provide two simple upper bounds that hold for the entire range of  $0 \leq \gamma \leq 1$ .

1. Since the sources can transmit at a maximum rate of  $W$  bps, the total throughput is upper bounded by  $n_s W$  bps.
2. Let  $\lambda(n)$  be any feasible throughput per multicast source. Then, there has to be at least  $n_s n_d \lambda(n)$  bps of data transfer in the network. Even when all nodes obtain full rate of  $W$  bps, the total data transfer is  $nW$  bps. Therefore,  $nW/n_d$  bps is an upper bound to the total throughput.

The above results are summarized in the following lemma.

**Lemma 2.3.1.** *Let  $0 \leq \gamma \leq 1$ . Then, an upper bound on the total multicast throughput is  $\min\{n_s, n/n_d\}W$  bps.*

### 2.4 Multicast Throughput: Protocol Model

The most natural setting to begin our analysis is one where all nodes in the network are mobile, i.e.,  $\gamma = 1$  (similar to that in [34]). Further, we con-

sider the protocol model for interference. Once we gain a good understanding of this, we extend our analysis to other settings in the next section.

### 2.4.1 All Nodes are Mobile

We consider the case where all nodes are mobile, i.e.,  $\gamma = 1$ .

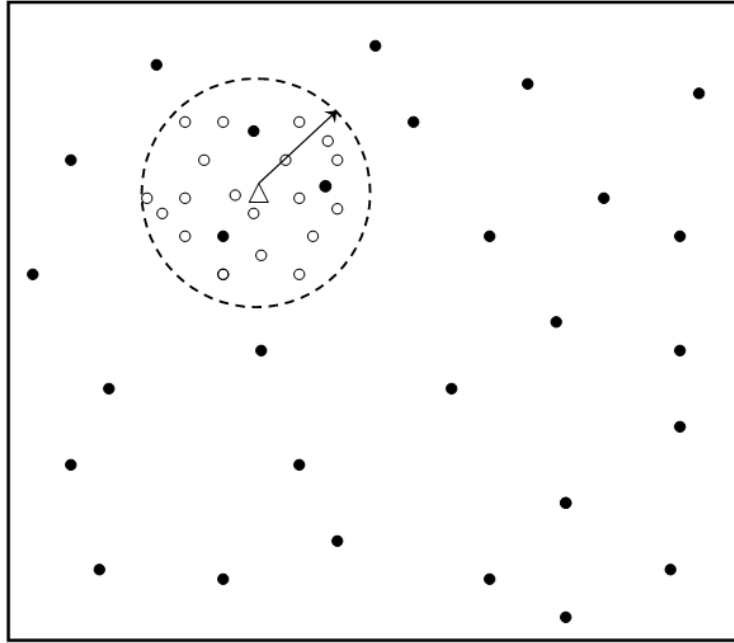


Figure 2.2: The first phase.

**Case  $n_s n_d \leq n$  :** The achievable scheme  $\Pi_1$  proposed next is a generalization of the *two phase scheme* introduced in [34]. In the *two phase scheme*, a data packet traverses at most two hops - one hop from its source node to a relay node or its destination node, and one hop from the relay node or its source node to its destination node. The main difference from the achievable scheme

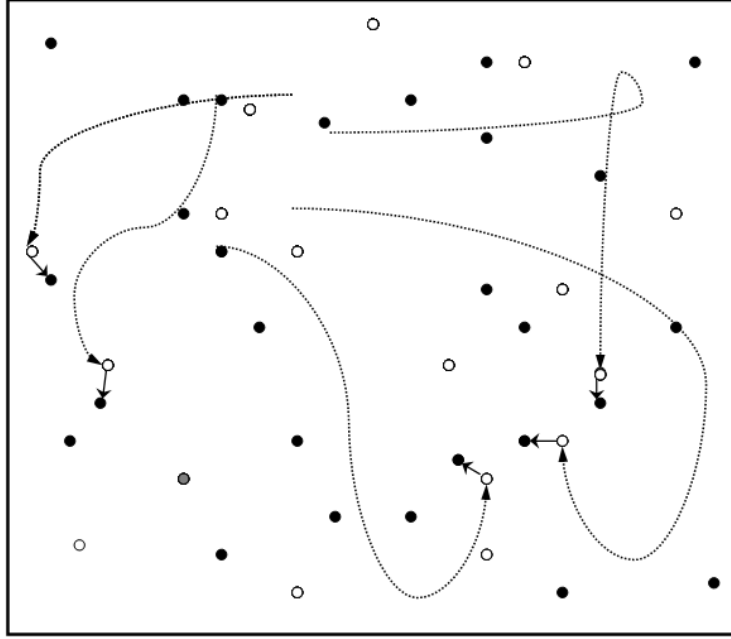


Figure 2.3: The second phase.

in [34] is that, in the first phase of  $\Pi_1$ , the sources simultaneously transmit the packets to multiple nodes per source. An outline of the achievable scheme  $\Pi_1$  is as follows.

1. In the first phase,  $\Theta(n_s)$  sources are selected such that, using a common radius of transmission of  $\Theta(1/\sqrt{n_s})$ , each of these sources can successfully transmit a packet to  $\Omega(n/n_s)$  nodes.
2. The nodes receiving packet can be relays or destinations for that packet. Since there are multiple destinations for each packet, both destination nodes and relay nodes store the packet in its buffer for the second phase.

3. In the second phase,  $\Theta(n_s n_d)$  nodes are selected such that, using a common radius of transmission of  $\Theta(1/\sqrt{n_s n_d})$ , each of these nodes can successfully transmit a packet to a destination node in the network. We assume that every transmitter has the knowledge of the packet IDs already received by the corresponding receiver. A packet is removed from the buffer of a transmitter when it learns that this packet is received by all its destinations.

In the first phase,  $K_1 n_s$  sources are selected such that the minimum distance between these selected sources is greater than  $\frac{K_2}{\sqrt{n_s}}$ . This is based on the following result.

**Lemma 2.4.1.** *Let  $m$  points be uniformly and identically distributed inside the unit disk. For large  $m$ , there exists  $K_1 > 0$  and  $K_2 > 0$  such that  $K_1 m$  points can be selected with high probability (w.h.p) such that the minimum distance between all these selected points is greater than  $\frac{K_2}{\sqrt{m}}$ .*

*Proof.* Please see Appendix A.1 □

These selected sources transmit with a common radius of transmission of  $\frac{K_2}{(2+\Delta)\sqrt{n_s}}$ . Let  $S_k$  denote the location of these source nodes and  $A_k$  be the circular area with  $S_k$  as the center and  $\frac{K_2}{(2+\Delta)\sqrt{n_s}}$  as the radius. From Lemma 2.4.1, it is clear that all the nodes inside  $A_k$  can decode the data packet transmitted by source  $S_k$  as (2.2) and (2.3) are satisfied.

The following result show that in all  $A_k$  w.h.p there are  $\Omega(n/n_s)$  nodes.

**Lemma 2.4.2.** *Let  $n/n_s \rightarrow \infty$ . Then, during the first phase of the achievable scheme  $\Pi_1$ ,  $\Theta(n_s)$  sources can simultaneously transmit to  $\Omega(n/n_s)$  nodes each w.h.p. if  $\epsilon < 1$ .*

*Proof.* Please see Appendix A.2 □

*Remark 2.4.1.* For  $\epsilon = 1$ , the first phase is scheduled in the same way as the second phase. For this reason, we do not address it separately.

In the second phase,  $K_1\bar{n}$  nodes are selected among randomly chosen  $\bar{n} = \frac{n_s n^{1-\epsilon}}{2}$  nodes such that the minimum distance between these selected nodes is greater than  $\frac{K_2}{\sqrt{\bar{n}}}$ . This follows from Lemma 2.4.1. These selected nodes transmit with a common radius of transmission of  $\frac{K_2}{(2+\Delta)\sqrt{\bar{n}}}$  to other (i.e. not among the  $\bar{n}$  nodes) destination nodes. Note that there are at least  $\frac{n_s n^{1-\epsilon}}{2}$  of these destination nodes. As mentioned before, the edge effects can be handled by looking at the constant  $K_3$  fraction of the area within the disk  $A$ . Let  $R_k$  denote the location of these selected nodes,  $B_k$  be the circular area with  $R_k$  as the center and  $\frac{K_2}{(2+\Delta)\sqrt{\bar{n}}}$  as the radius, and  $K_4 = \frac{\pi K_2^2 K_3}{(2+\Delta)^2}$ . Next, we prove that the second phase can be scheduled as claimed.

**Lemma 2.4.3.** *During the second phase of the achievable scheme  $\Pi_1$ ,  $\Theta(n_s n^{1-\epsilon})$  nodes can simultaneously transmit a packet to a destination node in the network w.h.p.*

*Proof.* Please see Appendix A.3 □



The two phases are alternatively scheduled, for instance, the first phase during odd time instants and the second phase during even time instants. We prove that this scheduling policy is achievable using union bound and Stirling's approximation. We have the following result.

**Lemma 2.4.4.** *Let  $n_s n_d \leq n$  and  $\gamma = 1$ . For the above scheme, the total multicast throughput achieved is  $\Omega(n_s)$ .*

*Proof.* Please see Appendix A.4 □

**Case  $n_s n_d > n$  :** The achievable scheme  $\Pi_2$  is based on  $\Pi_1$  performed over multiple time slots. Both first and second phases consist of  $\lceil n_s n_d / n \rceil$  time slots.

1. First phase: During  $\lceil n_s n_d / n \rceil$  time slots, the  $n_s$  multicast sources are scheduled in a round robin manner with  $n/n_d$  sources in each time slot. For each selected subset of sources, the first phase of  $\Pi_1$  is applied.
2. Second phase: During  $\lceil n_s n_d / n \rceil$  time slots, each node is considered as the destination of one of the multicast sources it is associated with in a round robin manner. For each time slot,  $\Theta(n)$  transmissions are scheduled as in the second phase of  $\Pi_1$ .

Next, we show that the above scheduling policy is orderwise throughput optimal.

**Lemma 2.4.5.** *Let  $n_s n_d > n$  and  $\gamma = 1$ . For the above scheme, the total multicast throughput achieved is  $\Omega(n/n_d)$ .*

*Proof.* Please see Appendix A.5 □

Summarizing the results obtained in this section (Lemma 2.3.1, Lemma 2.4.4 and Lemma 2.4.5), we obtain the following theorem on throughput capacity for the all-mobile multicast network.

**Theorem 2.4.6.** *The total throughput capacity of the all-mobile multicast network with protocol model for interference is  $\Theta(\min\{n_s, n/n_d\})$ .*

## 2.4.2 Hybrid (Static and Mobile) Nodes

We consider the hybrid network where  $n^\gamma$  ( $\gamma < 1$ ) are mobile nodes and the rest are static nodes. The main goal is to determine a regime where the presence of mobile nodes can provide an order-wise better throughput compared to the all-static multicast network. For this, we focus on a hybrid scheme consisting of two achievable schemes:

1. The achievable scheme based on forming multicast trees given in [49]
2. A simple extension of the achievable scheme proposed in Section 2.4.

In [49], the authors consider the all-static multicast network, and obtain the following main result.

*Result 2.4.1.* Let  $n_d = n^{1-\epsilon}$ ,  $0 < \epsilon < 1$  and  $\log n/n_s \rightarrow 0$  as  $n \rightarrow \infty$ . Then, the total throughput of the all-static multicast network with protocol model for interference is

$$\Theta \left( \min \left\{ n_s, \frac{\sqrt{n^\epsilon}}{\sqrt{\log n}} \right\} \right) \text{ w.h.p.} \quad (2.6)$$

The achievable scheme in this chapter is based on a comb architecture for multicast trees (see [49] for details).

For  $\gamma < 1$ , the above total throughput can be achieved as follows. The mobile sources transfer packets to nearby static nodes whereas the mobile destinations obtain packets from the multicast tree.

Next, we provide an achievable scheme  $\Pi_3$  for the hybrid network. Both first and second phases consist of  $\Theta(n_s n_d \log n/n^\gamma)$  time slots. The achievable scheme  $\Pi_3$  is as follows.

1. If  $n^\gamma \leq \sqrt{nn_d \log n}$ , then form multicast trees as described in [49].
2. Otherwise, perform two-phase scheme as described next.
  - (i) First phase: Let  $n_1 = \Theta(\min\{n_s, n^\gamma/n_d \log n\})$ . During each time slot,  $n_1$  sources are scheduled such that each of these sources can successfully transmit a packet to  $\Omega(n_d)$  mobile nodes. All sources are selected during  $\Theta(n_s n_d \log n/n^\gamma)$  time slots using time division.
  - (ii) Second phase: Let  $n_2 = \Theta(\min\{n_s n_d, n^\gamma/\log n\})$ . During each time slot,  $n_2$  mobile nodes are scheduled such that each of these

nodes can successfully transmit a packet to a destination node in the network. All destination nodes are selected during  $\Theta(n_s n_d \log n / n^\gamma)$  time slots using time division.

This scheme proof follows from [35] and proof results from section 2.4.1. Based on the analysis in Section 2.4, we obtain the following result on the total multicast throughput.

**Theorem 2.4.7.** *An achievable total throughput for the hybrid multicast network with protocol model for interference is*

$$\Omega \left( \min \left\{ n_s, \max \left\{ \frac{n^\gamma}{n_d \log n}, \sqrt{\frac{n}{n_d \log n}} \right\} \right\} \right).$$

*Remark 2.4.2.* From the above theorem, we conclude that presence of mobile nodes provide order-wise improvement in total multicast throughput if  $\gamma \geq 1 - \frac{\epsilon}{2}$  and  $n_s \geq \frac{\sqrt{n^\epsilon}}{\sqrt{\log n}}$ .

## 2.5 Multicast Throughput: Physical Model

We consider the case where all nodes are mobile and establish that, in this case, the total multicast throughput for physical model is same as that for protocol model. This suggests that the result is not interference model dependent. In this section, the key lemma (based on results in [34]) for the physical model is the following.

**Lemma 2.5.1.** *Let  $m$  sources  $S_1, S_2, \dots, S_m$  and a receiver  $V$  be uniformly and independently distributed inside the unit disk, and all the sources transmit*

with unit power. This node  $V$  can decode the packet transmitted from the nearest source with a non-vanishing probability.

*Proof.* Please see Appendix A.6 □

The scheduling policy  $\Pi_4$  for the case of physical model is similar to that of the case of protocol model. Note that the scheduling policy described next is for  $n_s n_d \leq n$ . It can be extended to  $n_s n_d > n$  similar to the case of protocol model. The two phases of  $\Pi_4$  are as follows.

1. All  $n_s$  sources transmit with unit power.  $\Theta(n_s)$  sources will have  $\Omega(n/n_s)$  nodes that decode successfully w.h.p.
2.  $\Theta(n_s n_d)$  nodes transmit with unit power.  $\Theta(n_s n_d)$  nodes will have a destination node that can decode successfully w.h.p.

This scheduling policy is based on the following lemma. In the lemma, the result for  $\beta > 1$  is required for the first phase, and  $\beta = 1$  is required for the second phase.

**Lemma 2.5.2.** *Consider  $m$  transmitter nodes that are uniformly and independently distributed inside the unit square. Also, consider another  $m^\beta$  ( $\beta \geq 1$ ) receiver nodes distributed uniformly and independently. With a non-vanishing probability, there will be  $\Theta(n)$  transmitter nodes that have  $\max\{1, \Theta(n^{\beta-1})\}$  receiver nodes that are nearer to it than other transmitter nodes.*

*Proof.* Please see Appendix A.7 □

The following result follows from Lemma 2.5.1, Lemma 2.5.2 and similar analysis as section 2.4.1.

**Theorem 2.5.3.** *The total throughput capacity of the all-mobile multicast network with physical model for interference is  $\Theta(\min\{n_s, n/n_d\})$ .*

## Chapter 3

# Multicast Interference Alignment of Wireless Networks

### 3.1 Introduction

#### 3.1.1 Motivation and Related Work

The interference channel has received significant attention in recent years. In particular, the concept of *alignment* has been developed and used effectively to determine rate regions for different classes of interference channels. Alignment desires to minimize the dimension of the space spanned by the interference for a given signal space dimension [26]. There have been various (related) notions of interference alignment developed in literature [51, 24, 19]. Ergodic alignment is a relatively recent concept where time variations in the  $K$ -user interference channel are effectively used to reduce the space spanned by interference [24]. In [24], the authors investigate the  $K$  user interference channel with circularly symmetric fading, and find that a rate of  $0.5 \log(1 + 2SNR)$  can be achieved for an additive Gaussian noise symmetric fading channel using ergodic alignment. Moreover, for a finite-field additive noise channel, the paper shows that ergodic alignment achieves capacity. Simultaneously, [39] analyzes unicast communication in a two-hop equal length network with time-varying channel states. Using store-and-forward relaying, the authors find that, for

finite field symmetric fading channels, *point-to-point* channel capacity can be achieved when the number of relays exceeds the number of source-destination pairs. The general message-set problem for a single-hop network is investigated in [25], where the authors find that ergodic alignment can be generalized to the multicast case using discrete Fourier transform (DFT) matrices. This proposed alignment scheme builds on the work in [39, 24, 25] to determine alignment strategies for a two-hop equal length multicast network.

### 3.1.2 Our Contributions

The following summarizes the two main contributions:

1. Determine alignment mechanisms for two-hop communication for multicast networks when the number of relays exceeds the number of sources/destinations.
2. Combine elements of the two-hop and ergodic alignment when the number of relays is smaller than the number of sources/destinations.

## 3.2 System Model

### 3.2.1 Network Description

We consider a finite-field network consisting of 3 layers (labeled source, relay and destination layers respectively). Each layer is assumed to have  $K_i$  active nodes, that can communicate with nodes in the neighboring layers (see Figure 3.1). For simplicity, we consider a system where the source and destination layers consist of  $K_1 = K_3 = K$ . The first hop is characterized by the



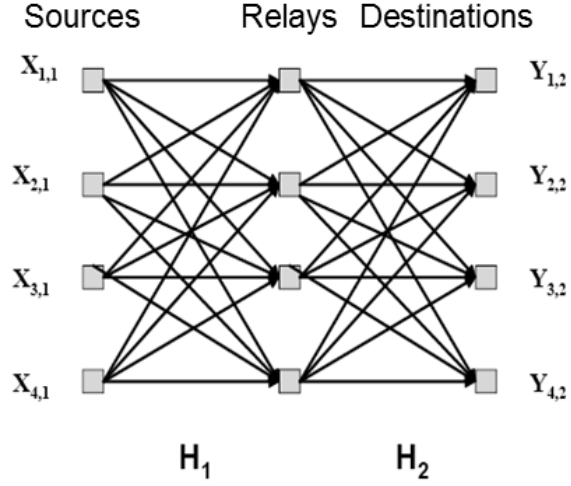


Figure 3.1: System model.

linear function

$$\mathbf{y}_1 = \mathbf{H}_1 \mathbf{x}_1$$

where all variables are vectors over  $\mathbb{F}_q$ . Here, the channel input from the source  $\mathbf{x}_1$  is a  $K \times 1$  vector over  $\mathbb{F}_q$  and  $\mathbf{H}_1$  is an  $K_2 \times K$  matrix.

The next hop is characterized by:

$$\mathbf{y}_2 = \mathbf{H}_2 \mathbf{x}_2$$

where  $\mathbf{x}_2$  is an  $K_2 \times 1$  channel input from the relay and  $\mathbf{H}_2$  is a  $K \times K_2$  matrix. In general,  $K_2 \neq K$ , which leads to two different cases that must be handled separately:

1. An “overprovisioned” network with more relays than sources  $K_2 > K$ .
2. An “underprovisioned” network where  $K_2 = \frac{K}{p}$ , where  $p \in \{2, 3, \dots\}$ .

The  $m$ -th hop transmission,  $m \in \{1, 2\}$ , is described as follows.

$$y_{j,m}[t] = \sum_{i=1}^K h_{j,i,m}[t]x_{i,m}[t]$$

Here the pair of indices  $(i, m)$  denote the  $i$ th transmitter, the pair  $(j, m)$  denote the  $j$  receiver, and  $h_{j,i,m}[t] \in \mathbb{F}_q$  is the channel connecting Transmitter  $i$  to Receiver  $j$  at time  $t$ . Here, we assume that  $h_{j,i,m}[t]$  are uniform and follow an i.i.d. fading model.

Note that our model does not incorporate an additive noise term at each hop of the network. This is for simplicity and to avoid distracting from the alignment scheme used in the chapter. Additive noise terms can be incorporated (with suitable modifications to the achieved set of rates) along lines similar to [24].

### 3.2.2 Problem Statement

We consider a set of length  $n$  block codes. Let  $W_k$  be the message of the  $k$ -th source uniformly distributed over  $\{1, 2, \dots, 2^{nR_k}\}$ , where  $R_k$  is the rate of the  $k$ -th source. A  $(2^{nR_1}, \dots, 2^{nR_K}; n)$  code consists of the following encoding, relaying, and decoding functions.

1. (Encoding) For  $k \in \{1, \dots, K\}$ , the set of encoding functions of the  $k$ -th source is given by  $\{f_{k,1,t}\}_{t=1}^n : \{1, 2, \dots, 2^{nR_k}\} \rightarrow \mathbb{F}_q^n$  such that  $x_{k,1}[t] = f_{k,1,t}(W_k)$  where  $t \in \{1, \dots, n\}$ .

2. (Relaying) For  $k \in \{1, \dots, K_2\}$ , the set of relaying functions of the  $(k, m)$ -th node is given by  $\{f_{k,2,t}\}_{t=1}^n : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$  such that  $x_{k,2}[t] = f_{k,2,t}(y_{k,1}[1], \dots, y_{k,1}[t-1])$  where  $t \in \{1, \dots, n\}$ .
3. (Decoding) For  $k \in \{1, \dots, K\}$ , the set of decoding functions of the  $k$ -th destination is given by  $g_k : \mathbb{F}_q^n \rightarrow \{1, 2, \dots, 2^{nR_k}\} \times \{1, 2, \dots, 2^{nR_{k^*}}\}$  such that  $(\hat{W}_k, \hat{W}_{k^*}) = g_k(y_{k,2}[1], \dots, y_{k,2}[n])$  where  $k^* = (k + 1) \bmod K$ .

The probability of error of the  $k$ -th destination is given by  $P_{e,k}^{(n)} = \Pr((\hat{W}_k, \hat{W}_{k^*}) \neq (W_k, W_{k^*}))$ . A set of rates  $(R_1, \dots, R_K)$  is said to be achievable if there exists a sequence of  $(2^{nR_1}, \dots, 2^{nR_K})$  codes with  $P_{e,k}^{(n)} \rightarrow 0$  as  $n \rightarrow \infty$  for all  $k \in \{1, \dots, K\}$ .

### 3.3 Multicast Alignment Strategies

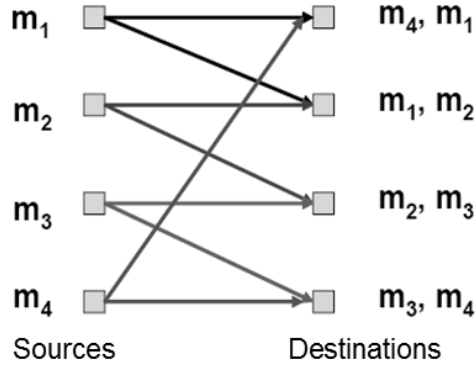


Figure 3.2: Desirable alignment scheme.

The main insight behind our strategies is similar to that of [24, 39] - different channel states across stages and/or times can be combined to create

an interference-free channel from source to destination. In the multicast network case, we desire to simultaneously sustain multiple access channel (MAC) between each destination and the corresponding sources from which it desires to receive messages, see Fig. 3.2.

### 3.3.1 Overprovisioned Network

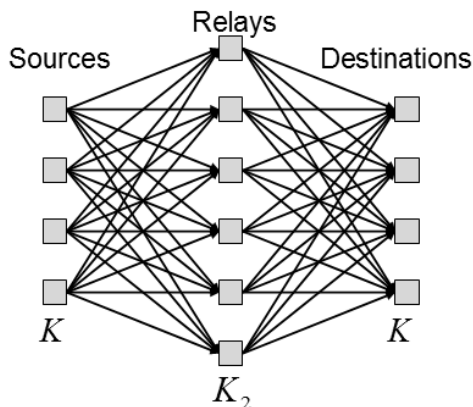


Figure 3.3: Overprovisioned network.

We start with the case when the number of relays exceeds the number of sources and/or destinations in the network, see Fig. 3.3. In this case, the alignment scheme resembles the unicast case analyzed in [39] in that it aligns channels between stages using a store-and-forward strategy.

**Definition 3.3.1.** The complementary time instant  $t_k^c$  is the time instant in the second hop that satisfy

$$\mathbf{H}_2[t_k^c]\mathbf{H}_1[t_k] = \mathbf{L}$$

for a pre-specified matrix  $\mathbf{L}$  and  $t_k$  is the  $k$ -th time instant in the first hop.

Here, the matrix  $\mathbf{L}$  represents the connections desired between a destination and its desired sources. In essence,  $\mathbf{L}$  has a non-zero entry in position  $(i, j)$  if the  $j$ th destination desires information from the  $i$ th source.

To simplify notation and present closed-form results, we consider the example symmetric multicast case first where

$$\mathbf{L}_K \triangleq \begin{bmatrix} 1 & l_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & l_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & l_{K-1} \\ l_K & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}. \quad (3.1)$$

**Lemma 3.3.1.** *For every full rank matrix  $\mathbf{H}_1[t]$  there exists a unique full rank matrix  $\mathbf{H}_2[t^c]$  such that  $\mathbf{H}_2[t^c]\mathbf{H}_1[t] = \mathbf{L}_K \forall t$ . Also, we have that  $(-1)^K \prod_{i=1}^K l_i \neq 1$ .*

*Proof.* Please see Appendix B.1 □

If the pair  $\mathbf{H}_1$  and  $\mathbf{H}_2$  satisfies the condition  $\mathbf{H}_2[t^c]\mathbf{H}_1[t] = \mathbf{L}_K \forall t$  then each destination  $k \in \{1, 2, \dots, K\}$  can receive messages from  $k$ -th and  $k^*$ -th source in the absence of interference.

Note that addressing low rank matrices cannot be done in a manner similar to the unicast case studied in [39]. In the multiple unicast case, every receiver is associated with a unique transmitter, and thus, in the case of low rank channel matrices, we can still find transmitter-receiver pairs that can still simultaneously communicate with each other. In the multicast case, however, the source-destination sets are characterized by the matrix  $\mathbf{L}_K$ , and thus, when

the matrix is low-rank, there is no guarantee that the rate between sources and destinations can be simultaneously sustained. Note that low rank matrices may not be as difficult an issue to tackle for such networks.

In a network where the number of relays exceeds the number of sources and/or destinations ( $K_2 > K$ ), our interest is in determining a  $K \times K$  sub-matrix of  $\mathbf{H}_1$  that is full rank. As long as such a matrix exists, the conditions imposed by Lemma 3.3.1 are met and thus simultaneous communication between source and destination(s) is possible. In fact, the following corollary is this realization stated formally:

**Corollary 3.3.2.** *For the case of an overprovisioned network (i.e.  $K_2 > K$ ). The sum of the achievable rates of the  $K$ -user finite-field relay channel is given by*

$$\sum_k R_k^s \geq \sum_k R_k^f \tag{3.2}$$

where  $R_k^f$  is the rate of  $k$ -th source in case when  $K_2 = K$  and  $R_k^s$  is the rate of  $k$ -th source for the case when  $K_2 > K$ , where  $k \in \{1, 2, \dots, K\}$ .

*Proof.* Please see Appendix B.2 □

Secondly, note that, if the alphabet size of the input is large enough, then the probability that a randomly generated  $K \times K_2$  matrix not being full rank is diminishingly small (this forms the basis for randomized network coding arguments, for example, see [36]). Thus, the case of low rank matrices can be tackled by concatenating the input to enlarge the channel alphabet

size. Finally, the low rank channel matrix case resembles the case of the underprovisioned network. This case is separately handled in the next section (Section 3.3.2). This leads to the main theorem for overprovisioned network:

**Theorem 3.3.3.** *In an overprovisioned network with  $K_2 \geq K$  and the multicast constraint matrix  $\mathbf{L}_K$  as specified by Equation 3.1, the following set of rates are simultaneously achievable:*

$$R_k + R_{k^*} \leq \log q \quad \forall k$$

*Proof.* Please see Appendix B.3 □

### 3.3.2 Underprovisioned Network

We consider underprovisioned networks that satisfy the condition  $K_2 = \frac{K}{p}$ , where  $p \in \{2, 3, 4, \dots\}$ , see Fig. 3.4. We have the following result.

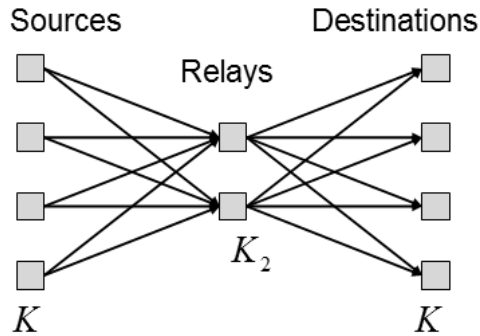


Figure 3.4: Underprovisioned network.

**Theorem 3.3.4.** *For the case of underprovisioned network, where  $K_2 = \frac{K}{p}$ . The sum of the achievable rates of the  $K$ -user finite-field relay network are*

given by

$$\sum_k R_k^u = \frac{1}{p} \sum_k R_k^f. \quad (3.3)$$

Also, individual rates may also be characterized as

$$R_k^u = \frac{1}{p} R_k^f \quad \forall k \quad (3.4)$$

where  $R_k^u$  is the rate of  $k$ -th source in case of underprovisioned network, and  $R_k^f$  represents any rates achieved on a network with  $K_2 = K$  where  $k \in \{1, 2, \dots, K\}$ ..

*Proof.* Please see Appendix B.4

□



## Chapter 4

# Real-Time Rate-Distortion Optimized Streaming of Wireless Video

### 4.1 Introduction

Previous chapters of this thesis have focused on fundamental, theoretical limits of multicast transmission. However, characterizing fundamental limits on a theoretical level does not directly translate into practically achievable performance, nor does it tell us what strategies or algorithms to implement in real-world deployments. In this final chapter of the thesis, we focus on implementation and algorithmic aspects of wireless multicast networks, in particular on optimized video streaming over an autonomous aerial vehicle (AAV) testbed. In contrast to previous chapters, in which we applied theoretical tools, this chapter deals with practical real-world aspects. In both cases, we aim to investigate constraints and capabilities for delivering information efficiently over wireless mobile networks, which are inherently broadcast and multicast in nature.

In this chapter, we describe algorithms primarily for unicast and multiple unicast video traffic delivery over mobile wireless networks. To the best of our knowledge, rate-distortion optimized real-time streaming of wireless

video has not yet been demonstrated in a real-world setting. We develop a first implementation of unicast video delivery on an AAV testbed. Given the broadcast nature of the medium, generalizing these algorithms to multicast traffic is straightforward and discussed at the end of this chapter.

#### 4.1.1 Motivation

An explosion of interest in multimedia systems in the last decade has resulted in the need for developing efficient protocols for the delivery of rich content (such as video) across a wireless mobile network. Such media is delay sensitive while being both computationally and bandwidth intensive, and it poses considerable challenges in guaranteeing its reliable delivery. The main features that make real-time packetized media delivery particularly challenging are [23]:

1. *High data rate:* media (especially video) requires high data rate even when the physical, wireless link between nodes is rapidly changing over time.
2. *Real-time constraints:* there is a time-to-live (TTL) associated with each packet, so a packet received after its TTL expires is lost. In addition, in live streaming, the video packets has to be delivered within a limited transmitter/camera to receiver/display latency. Otherwise, they will be dropped/lost.
3. *Dependencies between frames:* MPEG4-H.264, MPEG2 and other com-

pressed video formats are characterized by inter-frame dependency. This has two effects: first, if some frames are not successfully received this will lead to dropping of other successfully received frames because of their interdependency. Second, the number of interdependent frames determines the extent of compression in the video.

Rich multimedia delivery is increasingly an integral component specifically in many embedded and cyberphysical system applications, which creates additional implementation challenges. Such systems often have to operate in tightly constrained environments that severely limit the available computational performance or the amount of power that can be consumed. Given a vast array of possible embedded and cyberphysical system implementation options and parameters, analyzing and designing protocols and application algorithms<sup>1</sup> for them is a considerably daunting task that invariably requires integrated, cross-layer and cross-domain co-design approaches.

In this chapter, we target the analysis and co-design of multimedia delivery over a particular cyberphysical system consisting of a network of autonomous aerial vehicles (AAVs). Such systems are of considerable interest across multiple civilian and military applications, including search and rescue, perimeter monitoring and object tracking. A network of AAVs poses a vast

---

<sup>1</sup>The word "Algorithm" comes from the name Al-Khwrizm (c. 780-850), a Muslim mathematician, astronomer, geographer and a scholar in the House of Wisdom in Baghdad. He wrote an algorithm for distributing the inheritance of the deceased to his relatives according to the rules of *Quran*.

array of challenges - mobility, tracking and collision avoidance are essential for the physical operation of each AAV, while coordination and teaming critical for the network to carry out the task at hand. Central to all of these challenges is the ability to exchange high bandwidth delay sensitive data between the nodes in the network as reliably and efficiently as possible.

Our ultimate goal is to develop algorithms that exploit the structure of multimedia to deliver them efficiently and reliably over an AAV network, and test them in a real-world setting using a testbed. We have developed our own low-complexity rate-distortion optimized (RDO) streaming algorithms, and show that they outperform other mechanisms in the context of Horus, a custom built AAV testbed [8]. In a first step, we have developed software simulations for the mobility and channel models between AAVs, and we tested both existing and our proposed RDO video streaming techniques using these simulation models. Results show that optimized streaming can result in much more reliable and efficient video delivery than traditional protocols, in variants both with or without feedback. In the second step, we have implemented the system in realistic settings and tested our proposed RDO protocol and other protocols for different video compression. We used both temporal and spatial distortion measures to select the most reliable and efficient protocol.

#### **4.1.2 Related Work**

There is considerable existing literature on developing protocols for efficient data delivery over wireless networks. A majority of this literature tends

to focus on sensor networks designed for static settings, where nodes sense physical quantities that undergo a gradual change over time, e.g. temperature. For these applications, only a low data rate is required. Increasingly, sensor networks research incorporates dynamic network topologies as well. In [41, 44, 55], the authors conduct an experimental analysis on a dynamic sensor network where nodes move in a large area gathering data and then sending the collected data when near an access point. In our network, the nodes move in a prespecified pattern and gather media, e.g. video signals, and communicate them through the wireless network to an access point in another network. Note that our network is dynamically changing rapidly and intended to support a much higher data rate than conventional sensor networks.

Simultaneously, there is a growing body of work on media compression and streaming, both over wired and wireless networks. One of these research efforts is presented in [28], where the authors address the problem of streaming packetized media over a lossy network in a rate-distortion optimized way. In [28], simulation results show that systems based on rate-distortion optimization (RDO) algorithms have steady-state gains of more than 2-6 dB compared to systems that are not rate-distortion optimized. In this work, a simplified simulation model approximating real-world conditions is assumed, which is only a first step in measuring the performance and expected improvement an algorithm has over existing implementations. For wide-spread system deployment and evaluation of achievable gains of any algorithm, experiments and validations must be carried out in a realistic setting. Towards this goal,

we model and deploy RDO implementations in the context of an actual AAV testbed, where realistic simulations are a first step followed by running physical experiments in the field. Furthermore, the original RDO algorithm presented in [28] is based on ideal assumptions, e.g. in terms of its implementability. We instead propose modified RDO versions that can be efficiently realized with little to no overhead as part of standard network stacks on restricted embedded platforms.

### 4.1.3 Our Contributions

Our contributions in this chapter are summarized as:

- A new low-complexity RDO algorithm using default Media Access Control Layer (MAC-L) ACKs, which is validated through real-world simulations and shown to outperform ACKed and not ACKed transmission.
- A RDO algorithm using MAC-L beacons in order to achieve optimized video streaming under even lower complexity, as validated through experiments on a real-world AAV testbed and shown to minimize drops.
- A co-design of a RDO algorithm with adaptive video encoding to achieve optimized video streaming, developed for both MPEG2 and MJPEG streaming and validated through experiments on a real-world AAV testbed, where it is shown to improve received video quality.

## 4.2 Rate-Distortion Optimization Problem

The RDO problem aims to optimize the amount of distortion in a network against the rate. Distortion is defined as the degradation in media quality as packets are dropped. The rate represents the amount of data, i.e. the number of packets transmitted per unit time. The data packets that comprise a stream vary in their importance in contributing to output quality and, conversely, distortion. As such, RDO is concerned with deciding which packets to drop based on media quality metrics, measuring both the deviation from the source material and the bit cost for each possible decision outcome. In other words, the problem aims to solve the question of, *which* packets to select for transmission, *when* to transmit them, and *how* to transmit them (e.g., how many times), such that the expected distortion is minimized, subject to constraints on the expected rate.

In [28], the authors presented an algorithm that minimize the distortion  $D$  for a given rate  $R$ . This is done by minimizing the Lagrangian  $D + \lambda R$  for some Lagrange multiplier  $\lambda$ . This algorithm is based on off-line transmission policy computation and on-line transmission policy truncation. This problem is defined and solved for every data unit (i.e. video packet)  $l$ . As such, there exist a Lagrange multiplier  $\lambda_l$  for every data unit  $l$ . Depending on the expected channel rate, the value  $\lambda_l$  is a packet threshold used to decide if data unit  $l$  is the optimal video packet for transmission at this time instant or not. Solving the rate distortion optimization problem is not efficient for embedded applications in a real-time setting as the transmission policy computation for every

packet threshold  $\lambda_l$  is time consuming and needs to be performed off-line. Instead, we propose novel RDO algorithms with low computational complexity in which the transmission policy is computed online in real time. In addition, solving the rate-distortion optimization problem presented in [28] requires a mathematical channel model with parameters that are updated regularly. If used in a real-time system, this will add significant computational complexity. Instead of a mathematical channel model, we measure the channel state by using measurable physical quantities that are already available in default system operation and therefore will not require extra computation. We use two types of channel state feedback for our two proposed algorithms:

1. MAC-L ACKs<sup>2</sup>: In standard 802.11 wireless networks, the destination sends ACKs to the source when packets are successfully received. These ACKs are used in our algorithm at the source to measure the channel state. We call the low complexity RDO algorithm that uses MAC-L ACKs **LCRDO-Ack**.
2. MAC-L beacons<sup>3</sup>: In standard 802.11 wireless networks, stations send beacons. To further reduce complexity, these beacons can be used instead of more frequent ACKs to measure the channel state at the source.

---

<sup>2</sup>In 802.11 networks, a transmitting station can not listen for collisions while sending data, mainly because a station can not have its receiver on while transmitting a frame. As a result, the receiving station needs to send an acknowledgement (ACK) if it detects no errors in the received frame.

<sup>3</sup>In 802.11 networks, access points periodically broadcast a beacon. The radio network interface card (NIC) receives these beacons while scanning and takes note of the corresponding signal strengths. The beacons contain information about the access point, including service set identifier (SSID), supported data rates, etc.



We call the low complexity RDO algorithm that uses MAC-L beacons **LCRDO-Beacon**.

In addition to basic LCRDO variants, we co-design LCRDO-Beacon with adaptive video encoding algorithms using both MPEG2 and MJPEG compressions. We call this third low complexity RDO algorithm with adaptive co-design **LCRDO-Adaptive**.

The proposed LCRDO-Ack, LCRDO-Beacon, LCRDO-Adaptive algorithms have two important characteristics:

1. Real-time compatibility, where the transmission policy/channel state is computed on-line.
2. Lower complexity in terms of computational processing requirements.

### **4.3 Testbed**

We demonstrate real-world performance of our optimized RDO algorithms in the context of Horus, a testbed composed of a network of AAVs communicating wirelessly in an ad-hoc fashion. In our experimental analysis, the AAV nodes are equipped with video cameras and are capable of streaming packetized media between them. Our network consists of a fixed number of nodes, that are placed in a prespecified topology. In this network, sources are streaming video data in real time to destinations. The routing path is given a priori and the topology of the network is fixed throughout the experiment,

but nodes are mobile and moving in fixed circular paths. This continuous movement results in a time-varying nature of the wireless channels and reveals the effectiveness of the implemented algorithms. We implement rate-distortion optimized algorithms and measure the system performance for these networks.

#### **4.3.1 System Simulation**

We have setup a simulation of Horus network in the OMNeT++ network simulator framework [13] using the MiXiM package [12]. To accurately mimic and evaluate RDO behavior in the Horus setup, we model both the time-varying nature of the network topology as well as the modified protocol stack including our RDO layer on top of the OMNET++ component library. Each AAV node is described using a standard OMNET++/MiXiM network stack consisting of a physical layer, a MAC-L layer and an application layer running the RDO optimized video streaming. Furthermore, we simulated mobile, time-varying network topologies using the circular motion module of OMNET++.

#### **4.3.2 System Implementation**

In the following, we describe the actual implementation of our initial realization of the Horus network. We are considering AAVs as aerial nodes that form the network under test, see Figure 4.1 and Figure 4.2. The AAVs are controlled manually during take-off and landing using a remote control (R/C) module running at 72MHz. In the air, the AAVs are operating under automatic control by the on-board inertial measurement unit (IMU). The IMU

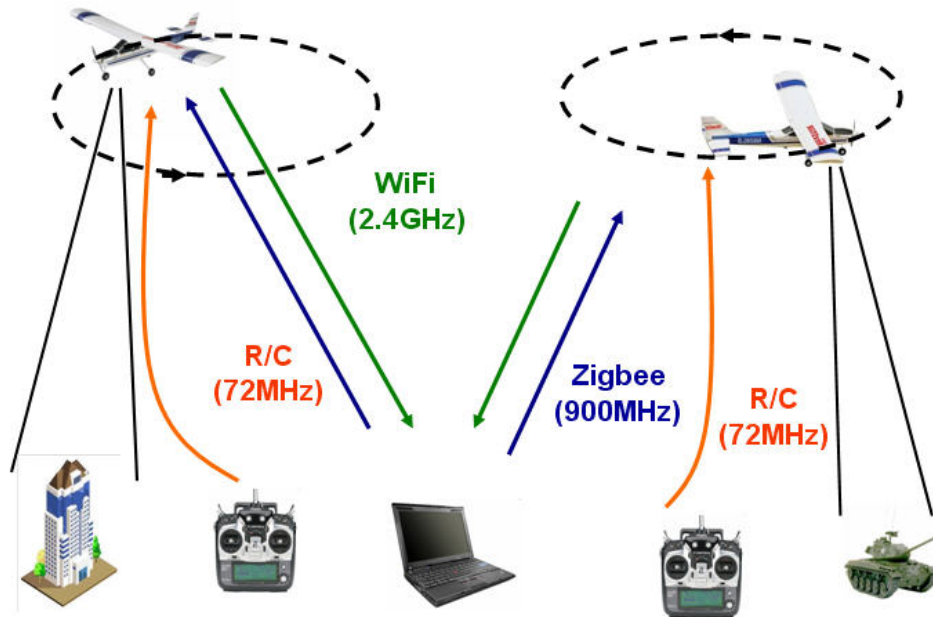


Figure 4.1: Horus project diagram.

uses stored way-points and GPS<sup>4</sup> to follow a preprogrammed path. The path can be changed while the AAVs are in the air by uploading new waypoints from the ground laptop to the on-board IMU via a Zigbee radio link at 900MHz.

Due to the nature of AAV nodes, there are constraints on the weight and dimensions of the components, as well as the power consumption, which directly affects the possible transmission range. We divide the AAV node architecture into three subsystems for propulsion, control, and communication, see Figure 4.3<sup>5</sup>. A picture of the components mounted inside the AAV is shown in Figure 4.4. The following lists represent the components that are

<sup>4</sup>GPS stands for Global Positioning System.

<sup>5</sup>UART, PWM, and USB stands for Universal Asynchronous Receiver/Transmitter, Pulse-Width Modulation, and Universal Serial Bus, respectively.



Figure 4.2: Autonomous aerial vehicle.

used to construct our system nodes and their corresponding functions. The communication subsystem consists of the following components:

1. **Via EPIA Nano-ITX [17]:** is a x86 computer, which is the central unit for managing and operating the communication between nodes.
2. **Atheros WiFi radio [4]:** is a wireless card for the IEEE 802.11 2.4 GHz frequency band, which is used for wireless video transmission.
3. **Video Camera [10]:** captures videos for our measurements and could be used for location identification and object recognition in future extensions of Horus.
4. **Communication Battery:** provides an independent power source for the communication subsystem.

The control section consists of the following components:

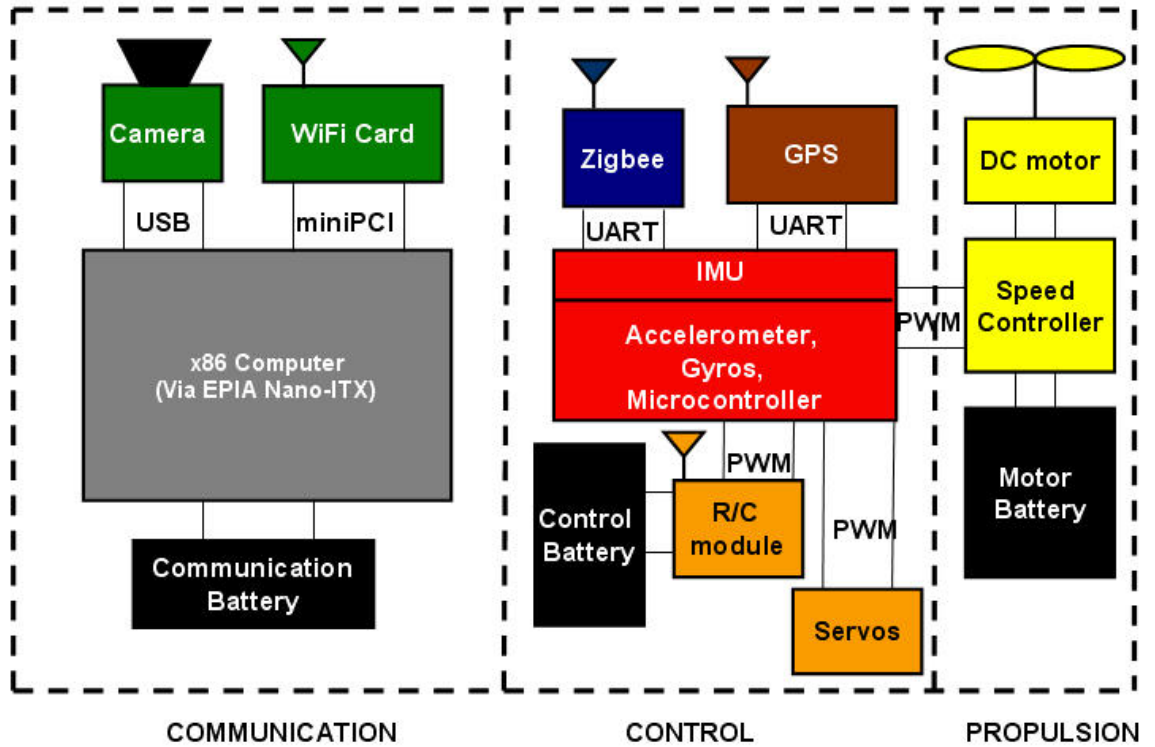


Figure 4.3: AAV block diagram.

1. **Zigbee [18]**: a radio for the IEEE 802.15 900 MHz frequency band used for receiving way-points for autonomous AAV navigation.
2. **IMU unit [2]**: controls the AAV movement during flight and sustains the required flight paths and mobile network topology.
3. **GPS unit [11]**: determines the location of the AAV for use by the autopilot in the IMU.
4. **R/C module [7]**: uses the 72 MHz frequency band and is the main manual ground control of the AAV. In Horus, it is used to control take-

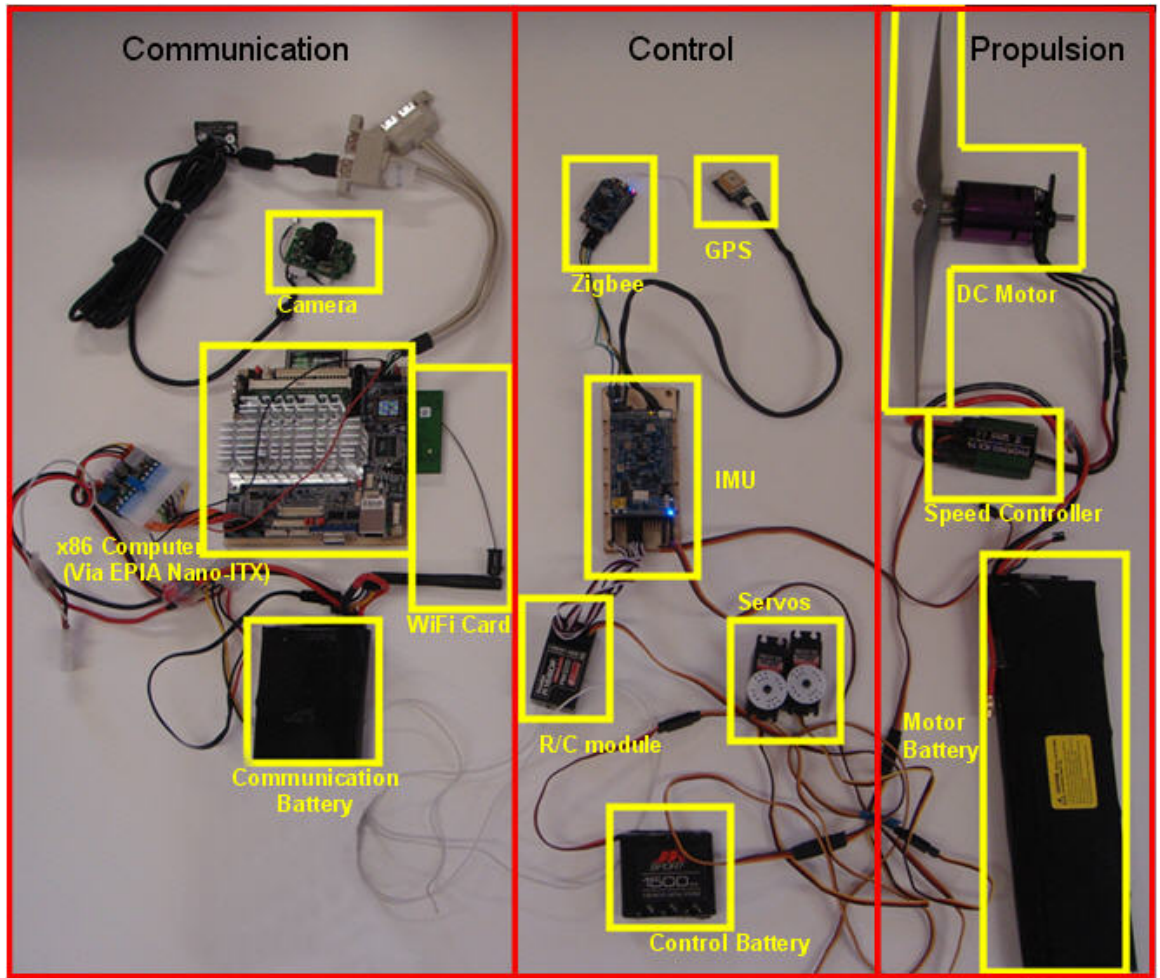


Figure 4.4: A picture of the components mounted inside the AAV.

off and landing of the AAV.

5. **Servo motors [9]:** act on flaps and rudder for controlling the direction and orientation of the AAV.
6. **Control Battery:** independent power source for the control subsystem.

Finally, the propulsion section consists of the following components:

1. **DC motor [6]:** is the main moving force of AAV and is connected to the propeller.
2. **Speed Controller [1]:** controls the speed of the DC motor by regulating the input current.
3. **Motor Battery [16]:** main power source for AAV propulsion.

Our choice of the components and division into subsystems is driven by the purpose of achieving:

1. **Reliability:** to ensure reliability, we choose communication links with different frequency bands to avoid any interference between the different transmissions. The following frequencies are used:
  - (a) R/C module: has a carrier frequency of 72 MHz where each node is on a different channel to avoid interference
  - (b) Zigbee network: operates on the 900 MHz frequency band to achieve a longer transmission range than at 2.4 GHz, and is used for sending way-points for AAV navigation
  - (c) WiFi network: uses 2.4 GHz for sending and receiving compressed video data
2. **Safety:** is achieved by separating the power supply of each section of the AAVs (propulsion, control, communication). Hence, for example, if

the propulsion battery is drained, we can still maintain control over the AAV for safe landing and avoiding a possible crash.

3. **Modularity:** the components are chosen such that they are able to act independently or interact with each other in various combinations and configurations. This simplifies testing and debugging process.
4. **Design Flexibility:** we are including more than one wireless transceiver to give flexibility in experimenting with various network designs.
5. **Availability and Cost:** is achieved by relying on off the shelf components.
6. **Weight and Dimensions:** the AAV node size and weight constraints are maintained.

#### 4.4 Low Complexity RDO with ACKs (LCRDO-Ack)

We realize the LCRDO-Ack algorithm as part of the application layer of our network. A conceptual block diagram for the LCRDO-Ack algorithm mapped to OSI network layers is shown in Figure 4.5. The video streams received from the camera are compressed into frames with different priorities. The *channel estimator* estimates the channel condition based on the received ACKs from previously sent packets. The *channel estimator* block is by default receiving ACKs from the destination in IEEE 802.11 networks. The *packet selector* block is responsible for selecting the suitable packet for transmission



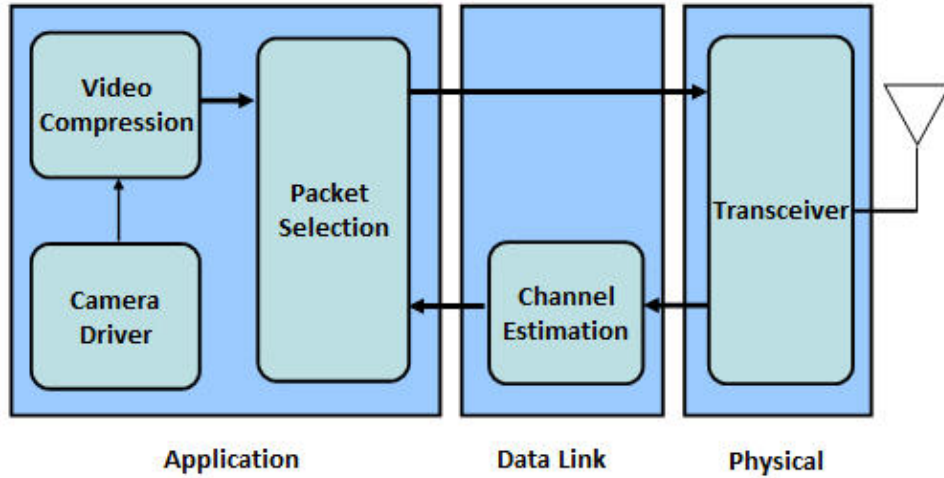


Figure 4.5: Conceptual block diagram of the RDO algorithm.

based on (1) the information received from the channel estimator and (2) the packet timestamp associated with each packet. In our case, the decision for retransmitting a packet or sending a new packet is made by the *packet selector* block according to the algorithm described in Section 4.4.1.

#### 4.4.1 LCRDO-Ack Algorithm

In MPEG4/MPEG2 video encoding, frames are compressed with different ratios and dependencies, giving each frame a different priority. Frames are divided into packets, where the amount of data and hence the number of packets typically increases with the frame priority. On the receiver side, these packets are recombined to form a frame that is then decoded. Therefore, losing a frame with high priority will lead to more deterioration in the quality of decoded video.

In line with existing video standards, we assume that the compressed video is composed of three types of frames: frames with first priority  $f_1$  (also known as *i frames*), frames with second priority  $f_2$  (*p frames*), and frames with third priority  $f_3$  (*b frames*). For compression purposes, the video frames are divided into Groups of Frames (GOF). Each GOF contains one  $f_1$  frame and a fixed number of  $f_2$  and  $f_3$  frames. The LCRDO-Ack algorithm that we use to optimize the video transmission is implemented in the application layer and is shown in Algorithm 1.

---

**Algorithm 1** LCRDO-Ack Algorithm

---

```

loop
  if  $t < t_{GOF\_timestamp}$  then
    if  $ID_{frame} = 1$  then
      if  $ID_{pkt} > ID_{finalf_1pkt}$  then
         $ID_{frame} ++$  {switch to next frame}
      else
        transmit  $f_1$  packet
        if success then
           $ID_{pkt} ++$  {switch to next  $f_1$  packet}
        end if
      end if
    else if  $ID_{pkt} \leq ID_{finalGOFpkt}$  then
      transmit  $f_2$  and  $f_3$  packets without retransmission
       $ID_{pkt} ++$  {switch to next packet}
    end if
  else
     $t_{GOF\_timestamp} += \Delta t_{GOF}$  {start new GOF interval}
     $ID_{frame} \leftarrow 1$  { $ID_{frame} = 1$  is dedicated for i frame in GOF}
     $ID_{pkt} \leftarrow ID_{finalGOFpkt} + 1$  {packet ID set to first packet in GOF}
  end if
end loop

```

---

The algorithm consists of two main sections, *timestamp check* and

*packet selection.* In the timestamp check, the algorithm starts by comparing the current time with the timestamp of the current GOF,  $t_{GOF\_timestamp}$ . If the GOF timestamp is not yet reached, packet selection is performed. Otherwise, the algorithm aborts the current GOF and advances to the next one. In packet selection (i.e. within the same GOF), the transmitter first sends  $f_1$  frame packets. Each  $f_1$  packet is (re-)transmitted until it is successfully sent and the algorithm can switch to the next one. After finishing the transmission of all  $f_1$  packets,  $f_2$  and  $f_3$  packets are sent. These lower priority packets are transmitted without waiting for feedback. This avoid wasting time in packet retransmissions, acknowledgements and reception times for these less important packets. At the end of packet selection, the packet ID is incremented to send the next packet until all packets in the current GOF are transmitted or a timeout is reached.

The LCRDO-Ack algorithm is implemented on top of standard protocol stacks. As an additional optimization we can, however, modify the 802.11 MAC-L to further improve overall real-time performance. Specifically, when the MAC-L receives a packet with a  $f_1$  flag, it uses its default behavior to retransmit the packet up to 3 times until an ACK is received. If no ACK is received after 3 tries, the MAC-L reports a drop to the application layer. By contrast, in case of  $f_2$  or  $f_3$  packets, the RDO algorithm does not require feedback about transmission success and a modified MAC-L can transmit the packets only once without waiting for any ACK. This further reduces overall overhead and latencies.

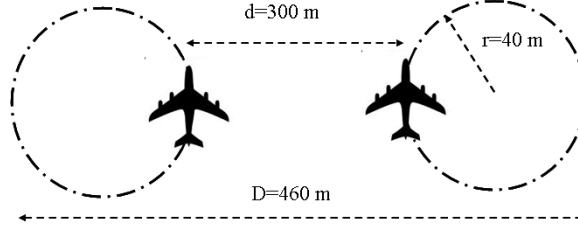


Figure 4.6: Simulated network topology.

#### 4.4.2 Distortion Measure

We investigate the LCRDO-Ack algorithm using a multiplicative distortion measure. The multiplicative distortion  $D_m$  is initially set to the maximum distortion level  $D_0$ . When frames are successfully received, the distortion decreases by the number of frames of type  $i$ ,  $N_{f_i}$ , multiplied by all the frames of higher priority successfully received within a GOF. This gives zero weight if frames of higher priority have not been successfully received.  $D_m$  is evaluated every GOF as

$$D_m = D_0 - N_{f_1} \left( 1 + N_{f_2} (1 + N_{f_3}) \right); \quad 0 \leq D_m \leq D_0.$$

We also define the sum of multiplicative distortion  $D_M$  as the sum of  $D_m$  for all transmitted GOFs:

$$D_M = \sum_{GOF} D_m.$$

$D_M$  is the distortion measure we use to compare different transmission protocols.

### 4.4.3 Experimental Setup

For our experiments, we assume a topology in which two or three AAVs (hosts) move in fixed circular patterns with a radius of 40 meters and a distance of 380 meters between the circles centers, as shown in Figure 4.6. Nodes send data packets in a one-hop fashion over a 802.11 wireless connection, where the source node transmits packets directly to the destination node. Next to the transmission under test, we include a third node that simultaneously transmits other packets not related to the main video stream. This setting allows us to analyze RDO transmission in the presences of high interference and consequently when experiencing a large packet loss.

We simulated this setup in OMNET++ using a simple path loss channel model as the one most closely resembling AAV-to-AAV conditions with little to no fading and no shadowing effects. The continuous motion of the nodes in circular paths leads to time-varying channel effects and a network packet drop rate that depends on the relative position of the AAVs. We use different path loss exponents  $\alpha$  to model and experiment with normal and worst case channel conditions. For worst-case analysis, we assume an exponent  $\alpha$  of 3.7. Together with interference from a third node as described above, we observe an overall packet drop rate of 40%, which allows for comparison of various transmitters under realistic conditions.

For testing the LCRDO-Ack algorithm, we compare it against conventional transmission algorithms. Overall, we define three types of transmitters:

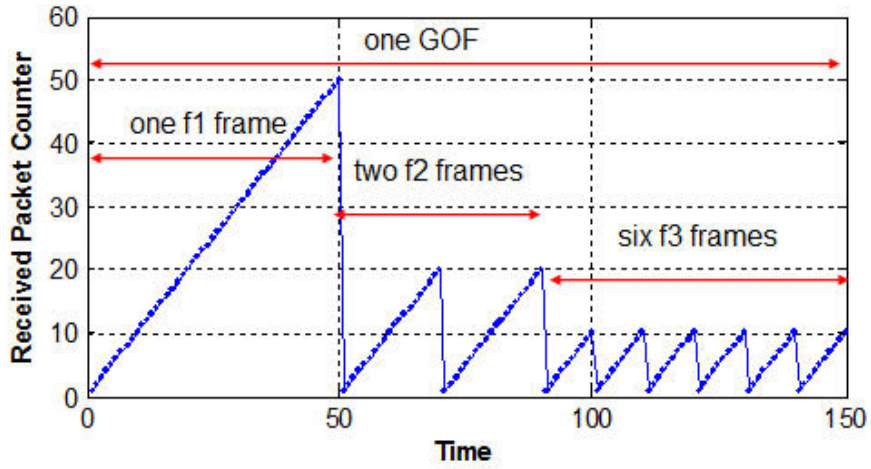


Figure 4.7: Received packet counter for a complete GOF successfully received.  $f_1$  frame counts to 50,  $f_2$  frame counts to 20 and  $f_3$  frame counts to 10. Any packet reception less than that count is not complete and the frame is considered dropped in the distortion measure.

1. *Transmitter without ACK*: transmits every packet without waiting for an ACK from the receiver.
2. *Transmitter with ACK*: retransmits every packet until it receives an ACK for each packet.
3. *Transmitter running the LCRDO-Ack algorithm*: implements the LCRDO-Ack algorithm described previously.

#### 4.4.4 Results

To compare different transmitters, we run the network simulator for each transmitter under the exact network conditions mentioned in the previous subsection. We specify 3500 packets to be transmitted from the source to the

Table 4.1: Payload parameters

Total number of packets	3500
Number of priority $f_1$ frames per GOF	1
Number of priority $f_2$ frames per GOF	2
Number of priority $f_3$ frames per GOF	6
Number of packets per frame $f_1$	50
Number of packets per frame $f_2$	20
Number of packets per frame $f_3$	10

destination node. For simplicity, we fix the number of packets per frame for a given priority frame. The payload parameters are shown in Table 4.1 (i.e.  $D_0=15$  and the maximum distortion measure for the whole transmission, where no frame is successfully received, is 350).

Our performance investigation for this problem includes both a network measure (e.g. number of packets drops) and an optimization measure (e.g. quality of the received media). We illustrate both using an easy-to-visualize proxy variable, which is a counter at the receiver. This counter counts the number of successfully received packets and is incremented until the end of the frame is reached, upon which the counter is reset to zero. Counting then starts in the same manner for the next frame, and so on, see Figure 4.7. The counter can determine if the received packet is in the current frame or not by checking the packet ID number associated with it. This way, plotting the counter values over time visualizes the performance of the receiver with respect to the video frames. These values are later used to measure the distortion.

The No-ACK Transmitter sends packets continuously without receiving any ACKs from the receiver. This leads to loss of packets with equal prob-

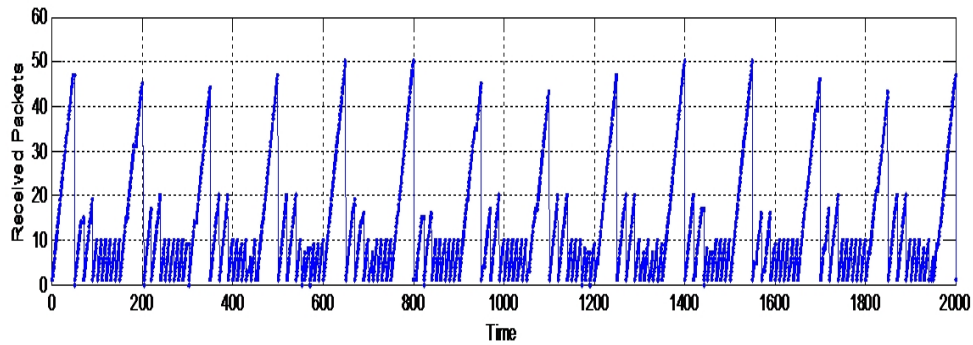


Figure 4.8: Received packet counter for No-ACK transmission.

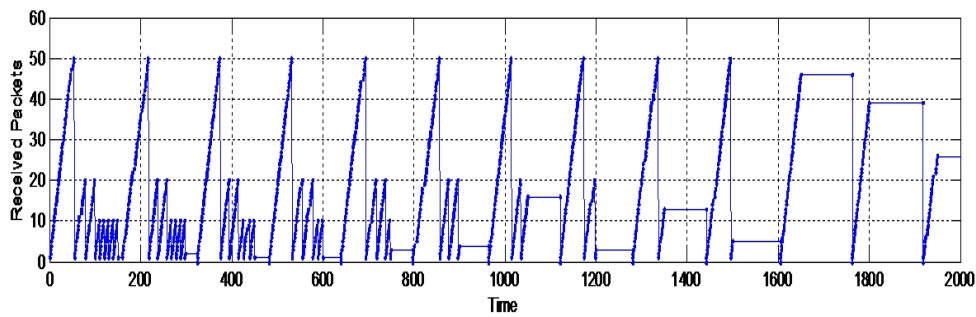


Figure 4.9: Received packet counter for ACK transmission.

ability for different priority frames, which results in a 40% loss of the first priority packets essential to decode the other packets sent within a GOF. Due to this behavior (Figure 4.8), about 40% of frames are not completely received leading to high distortion, as shown in Figure 4.11.

The ACK Transmitter sends new packets only after receiving an ACK for the previous packet, and it otherwise continues to retransmit the same packet. Therefore, all the frame packets within the current GOF timestamp



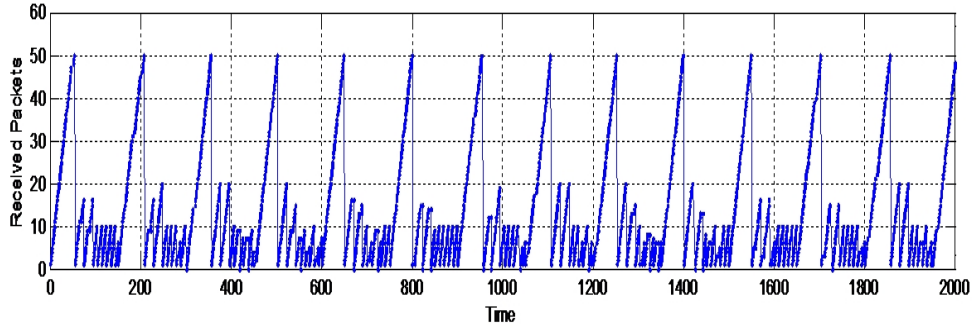


Figure 4.10: Received packet counter for LCRDO-Ack transmission.

are received successfully regardless of their priority. The frames received after their GOF timestamp are dropped, but at the same time cause more delay to build up with time. This accumulative delay is caused by retransmission and ACKing of packets of lower priority. Due to this delay, the number of frames that are lost increases as the transmission continues, causing a large degradation in the video quality over time. This leads to a significant increase in the distortion at the end of simulation. As shown in Figure 4.9, the number of frames lost per GOF increase as the transmission continues, leading to high distortion  $D_M$  (Figure 4.11). This transmitter experiences the highest distortion when the simulation is allowed to run for a sufficiently long time.

The LCRDO-Ack transmitter is designed to minimize the distortion measure and give better performance than conventional transmitters. It retransmits until an ACK is received only for the first priority packets. Second and third priority packets are sent without waiting for an ACK from the destination. This guarantees that frames of priority  $f_1$  will be received at the

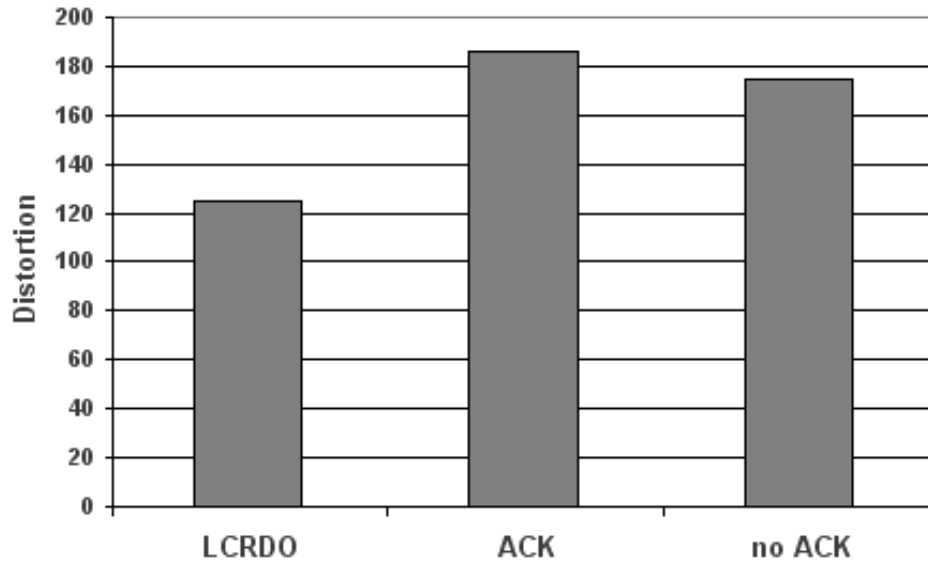


Figure 4.11: Distortion measure  $D_M$  for different transmission protocols.

receiver even under bad channel conditions, which is the case in our simulation. The second and third priority frames,  $f_2$  and  $f_3$  respectively, are dropped with about 40% probability, as shown in Figure 4.11. Overall, the LCRDO-Ack transmitter guarantees a minimum video quality at the receiver side and gives a better distortion than other transmitters.

In all simulated cases, the average packet drop rate is 40%. In the LCRDO-Ack case, it is guaranteed that most of these drops are lower priority packets, which affects transmission quality less and makes the LCRDO-Ack algorithm more robust. In the No-ACK and ACK transmitters, these dropped packets can be of any type of packet priority. Therefore, in these two cases, the 40% drop rate significantly affects transmission quality.

## 4.5 Low Complexity RDO with Beaconing and Adaptivity (LCRDO-Beacon and LCRDO-Adaptive)

We implemented LCRDO-Beacon and LCRDO-Adaptive algorithms in our physical testbed setup as described in Section 4.3.2. On the Via EPIA computer mounted inside the AAV, we run a Ubuntu 10.10 Linux operating systems with kernel version 2.6.35. We use the GStreamer open source multimedia framework [5] for video compression and the Click Modular Router open source network stack [15] for implementing our transmission algorithm. The operating system used for the ground station (i.e. the ground laptop) is Ubuntu 10.10 with Linux kernel 2.6.35. We implement the LCRDO-Beacon algorithm in connection with MPEG2 video compression and the LCRDO-Adaptive algorithm for both MPEG2 and MJPEG compressions.

### 4.5.1 LCRDO-Beacon Algorithm

The LCRDO-Beacon algorithm, used in the system implementation, is computationally less complex than the LCRDO-Ack algorithm (Algorithm 1) used in the system simulation. The reason for further reducing the complexity of the algorithm are the high computational demands of MPEG2 video encoding, which requires most of the computation power of Via EPIA computer. The modifications of switching from ACKs to beacons for channel state measurement ensure a more reliable real-time performance and concurrently show significant improvement in the general performance with respect to un-optimized methods, as will be shown in section 4.5.6.

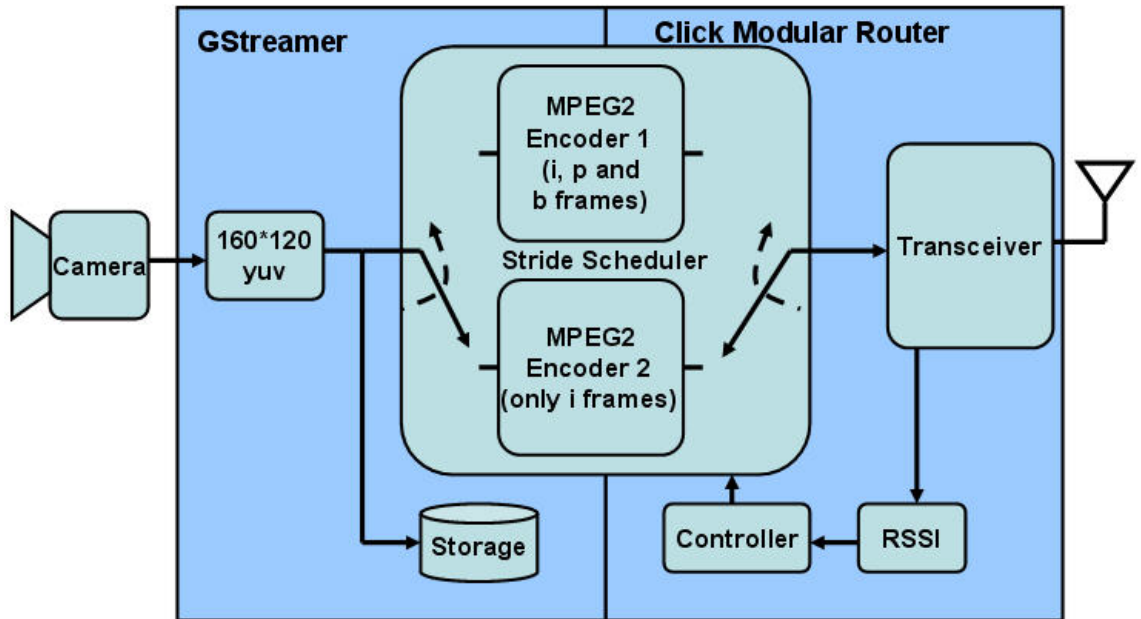


Figure 4.12: Block diagram of the LCRDO-Beacon algorithm at the transmitter.

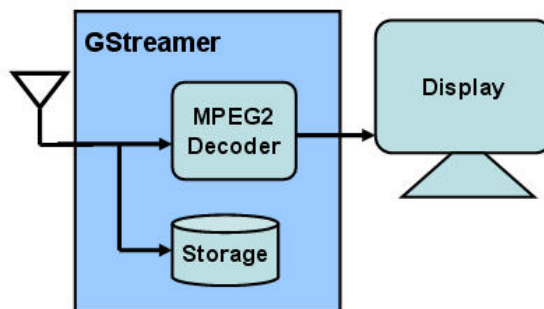


Figure 4.13: Block diagram of the LCRDO-Beacon algorithm at the receiver.

The block diagram of the transmitter is show in Figure 4.12. The camera driver outputs raw video, which is resized to 160x120 (without loss of generality, this resolution is chosen to minimize the computation on the Via

---

**Algorithm 2** LCRDO-Beacon Algorithm

---

```
loop
  transmit packets received from the selected/active encoder
  if transceiver receives a beacon then
    if  $RSSI < X_1$  then
      Clear Queue1 and Queue2 {Ensures real-time delivery}
      Switch to Encoder 2 (i frames only)
    else
      if  $RSSI > X_2$  then
        Clear Queue1 and Queue2 {Ensures real-time delivery}
        Switch to Encoder 1 (i, p and b frames)
      end if
    end if
  end if
end loop
```

---

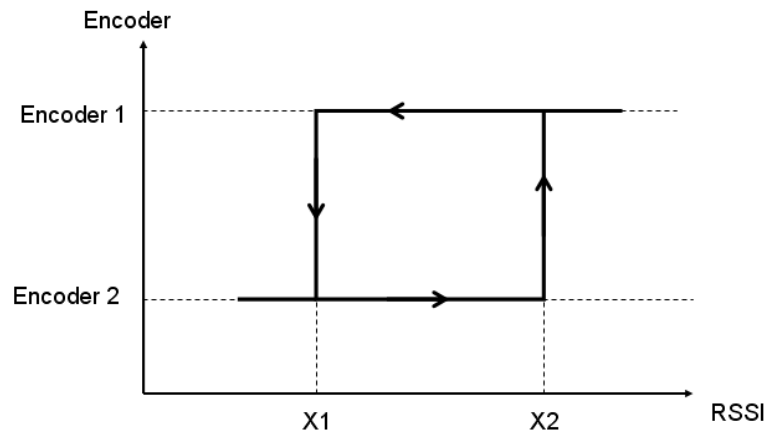


Figure 4.14: Hysteresis: Controller switches to Encoder 2 when RSSI is less than  $X_1$  and to Encoder 1 when RSSI is greater than  $X_2$ . No change is done when RSSI is between  $X_1$  and  $X_2$ .

EPIA computer; the algorithm applies equally for higher resolutions). A stride scheduler chooses between different parameters for encoding the raw video into an MPEG2 stream. We realize two different encoders:

1. MPEG2 encoder with GOF<sup>6</sup>=5 and an overall frame rate of 25 frames per second (f/s). This corresponds to sending i, p and b frames with an i frame rate of 5 f/s.
2. MPEG2 encoder with GOF=1 and an overall frame rate of 5 f/s. This corresponds to sending i frames only at a rate of 5 f/s.

Note that overall, this setup is equivalent to a general RDO architecture as described in Section 4.4 (Figure 4.5), where packets of p and b frames of a single, fixed MPEG2 encoder (running at 25 f/s) are selectively dropped depending on the chosen transmission policy. An equivalent implementation that alternates between two separate encoders as controlled by a stride scheduler was chosen due to limitations of the GStreamer-internal architecture.

The two outputs of the two MPEG2 compression and packet selection blocks are fed into the transmitter via two queues, *Queue1* and *Queue2*. These queues are not drawn to simplify the block diagram. For later comparison purposes, a high quality reference copy of the original video is stored in compressed MPEG2 form (with GOF=5 and 25 f/s) using GStreamer. Inside the Click Modular Router, a transceiver realizes wireless transmission of encoded videos and wireless reception of signal strength beacons. The beacons received in the

---

<sup>6</sup>Group Of Frames (GOF) is the group of video frames that starts by an i frame then proceeded by p and b frames only. The MPEG video file is a sequence of GOFs; e.g. GOF=5 has one i frame and four p and b frames, while GOF=1 has only i frames and no p and b frames.

transceiver are passed through a Received Signal Strength Indicator (RSSI<sup>7</sup>) block that decodes the RSSI value and passes it on to a controller block. Finally, the controller block determines packet selection and controls the stride scheduler by executing Algorithm 2.

In the LCRDO-Beacon algorithm (Algorithm 2), the controller switches to the Encoder 2 (i frames only) in the video compression block whenever the RSSI drops below a value X1. Likewise, if the RSSI rises above a value X2, the controller switches to select the Encoder 1 (i, p and b frames) in the video compression block. This hysteresis is shown in Figure 4.14. In the switching instant, all the contents of *Queue1* and *Queue2* are cleared. This is done to avoid sending any residual packets in the queue when switching back and forth between the different encoding modes, ensuring reliable real-time performance.

The block diagram of the receiver is shown in Figure 4.13. The received signal is decoded using a standard MPEG2 decoder and displayed on the monitor of the ground station (laptop). At the same time, the received video is stored in compressed MPEG2 format for later analysis.

#### 4.5.2 LCRDO-Adaptive Algorithm for MPEG2

In addition to selectively dropping packets, a generalized method for performing RDO and improving the received video quality is to co-design

---

<sup>7</sup>The Atheros based card returns an RSSI value of 0 to 127 (0x7f) with 128 (0x80) indicating an invalid value. There is no specified relationship of any particular physical parameter to the RSSI reading.

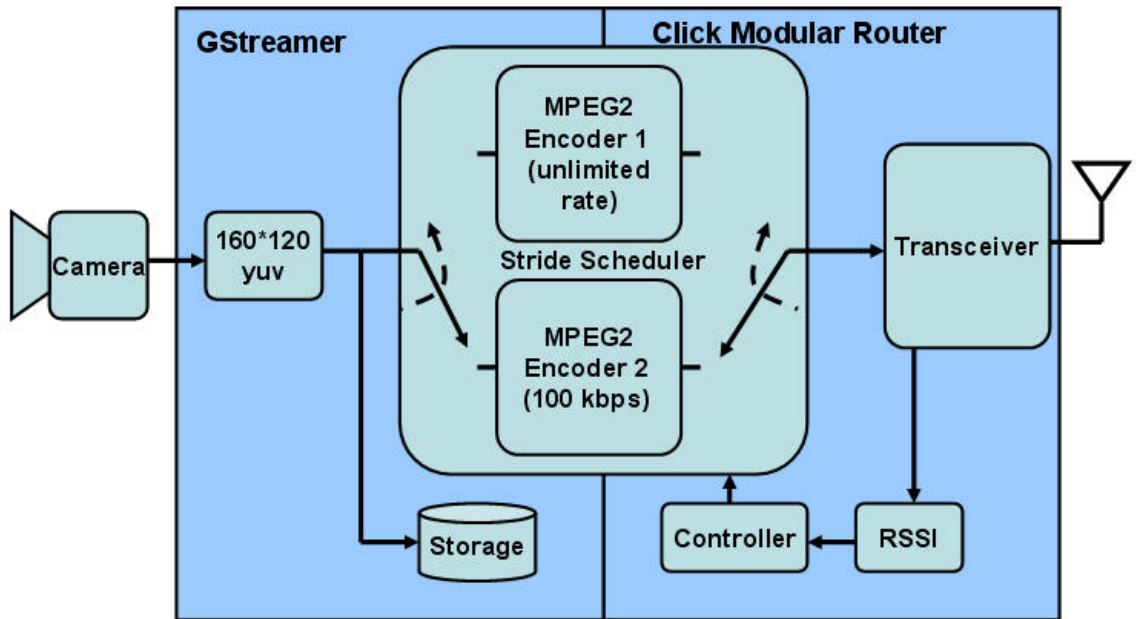


Figure 4.15: Block diagram of the LCRDO-Adaptive algorithm for MPEG2 at the transmitter.

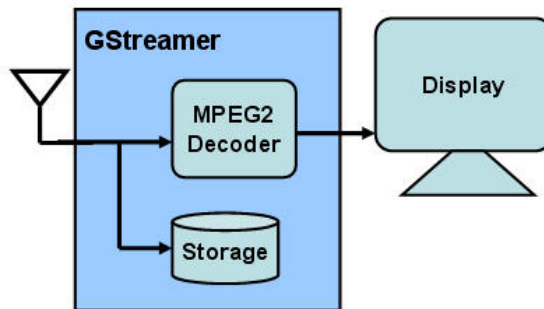


Figure 4.16: Block diagram of the LCRDO-Adaptive algorithm for MPEG2 at the receiver.

RDO-type packet selection with adaptive video encoding. In such an approach, the video encoding rate is adjusted to the transmission rate in a distortion-optimized way. In addition to improving video quality, adapting



encoding parameters to rate variations can significantly reduce average computational requirements in the real-time video encoder. Similar to the LCRDO-Beacon algorithm, such an adaptive approach has an operating mechanism in which the video encoder switches between different modes. Both algorithms require predetermined thresholds used in a hysteresis (see Figure 4.14). In the LCRDO-Beacon algorithm, the channel state measurement determines when to transmit both independent and dependent or when to drop dependent and only transmit independent frames. By contrast, for the adaptive algorithm, the channel state measurements determine when to transmit video at high quality, i.e. with high bit rate, and when to transmit video at low quality, i.e. with lower bit rate. Both algorithms require channel state measurements and seek to minimize distortion and maximize video quality.

---

**Algorithm 3** LCRDO-Adaptive Algorithm for MPEG2

---

```

loop
  transmit packets received from the selected/active encoder
  if transceiver receives a beacon then
    if  $RSSI < X_1$  then
      Clear Queue1 and Queue2 {Ensures real-time delivery}
      Switch to Encoder 2 (100 kbps)
    else
      if  $RSSI > X_2$  then
        Clear Queue1 and Queue2 {Ensures real-time delivery}
        Switch to Encoder 1 (unlimited rate)
      end if
    end if
  end if
end loop

```

---

The block diagram of the low complexity RDO with adaptive co-design

(LCRDO-Adaptive) algorithm for MPEG2 transmissions is shown in Figure 4.15. The block diagram is similar to the block diagram of LCRDO-Beacon with the exception that the stride scheduler allows choosing between two different MPEG2 encoders for video compression:

1. MPEG2 encoder with 5 frames per GOF, a frame rate of 25 f/s, and unlimited bit rate.
2. MPEG2 encoder with 5 frames per GOF, 25 f/s frame rate, and 100 kbps transmission rate.

The controller block determines packet selection and controls the stride scheduler by executing Algorithm 3.

The block diagram of the receiver is shown in Figure 4.16, which is similar to the LCRDO-Beacon receiver.

### 4.5.3 LCRDO-Adaptive Algorithm for MJPEG/SMOKE

The attractive property of motion JPEG (MJPEG) video compression is the very low computational complexity compared to MPEG2 video compression. This comes at the expense of lower compression leading to higher bandwidth requirements. Furthermore, MJPEG compression is characteristic by all frames being independent, whereas MPEG2 compression has both independent and dependent frames. A variant of MJPEG compression is the SMOKE codec [14], which includes both JPEG frames and delta frames. JPEG frames are key-frames that are each followed by  $N - 1$  delta frames. JPEG frames are

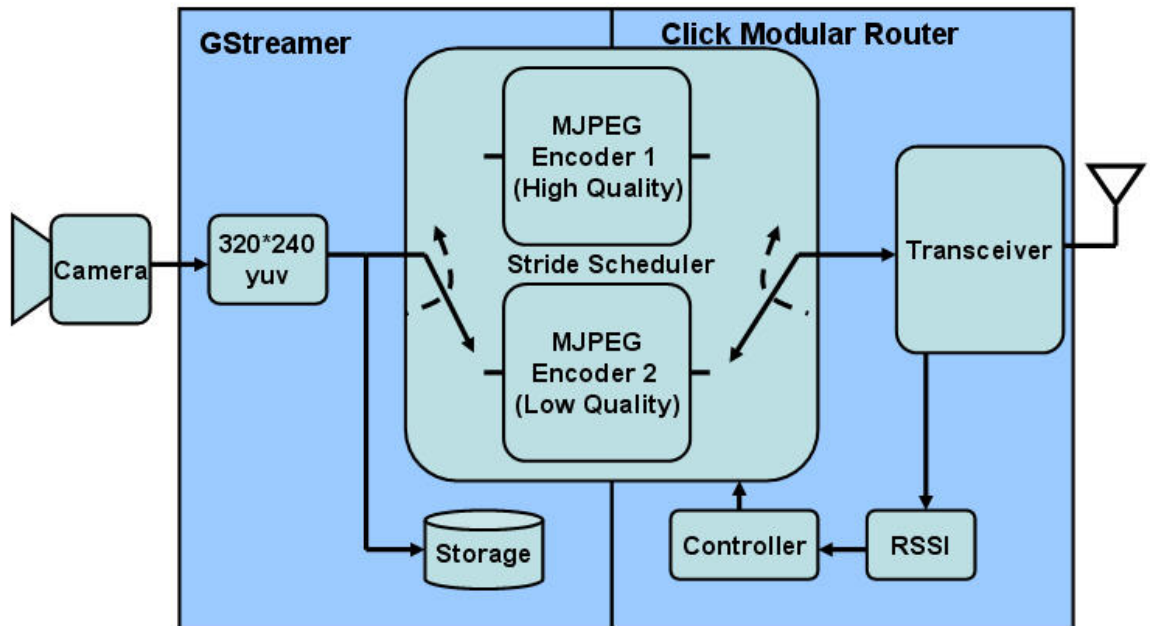


Figure 4.17: Block diagram of the LCRDO-Adaptive algorithm for MJPEG/SMOKE at the transmitter.

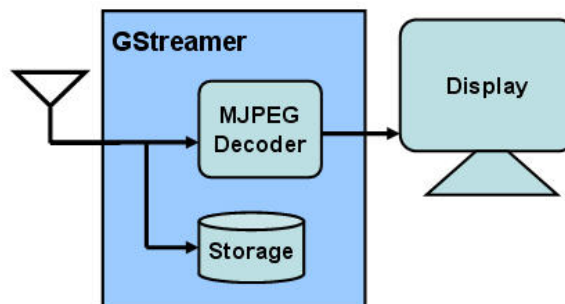


Figure 4.18: Block diagram of the LCRDO-Adaptive algorithm for MJPEG/SMOKE at the receiver.

independent while delta frames are constructed according to a motion estimation threshold using the key-frame. This threshold specifies how much each 16x16 block of pixels may differ before a new block is generated. A large value

---

**Algorithm 4** LCRDO-Adaptive Algorithm for MJPEG/SMOKE

---

```
loop
  transmit packets received from the selected/active encoder
  if transceiver receives a beacon then
    if  $RSSI < X_1$  then
      Clear Queue1 and Queue2 {Ensures real-time delivery}
      Switch to Encoder 2 (low quality)
    else
      if  $RSSI > X_2$  then
        Clear Queue1 and Queue2 {Ensures real-time delivery}
        Switch to Encoder 1 (high quality)
      end if
    end if
  end if
end loop
```

---

of the threshold causes more blocks to stay the same for more frames, decreasing bandwidth usage, but producing less accurate output. Likewise, a small number of delta frames between key-frames increase received video quality at the cost of a higher bit rate, and vice versa.

The block diagram of the LCRDO-Adaptive for MJPEG/SMOKE transmitter is show in Figure 4.17. The camera driver outputs raw video which is resized to 320x240 with frame rate of 10 f/s. A stride scheduler chooses between different parameters for encoding the raw video into an MJPEG/SMOKE stream. We realize two different encoders:

1. MJPEG/SMOKE encoder with high JPEG image quality of 80% and  $N=8$ .
2. MJPEG/SMOKE encoder with low JPEG image quality of 30% and

$N=8$ .

The rest of the block diagram is similar to the block diagram of LCRDO-Beacon with difference in the controller and the storage blocks. The controller block determines packet selection and controls the stride scheduler by executing Algorithm 4. The storage block stores high quality MJPEG/SMOKE compressed video (with JPEG image quality of 80%,  $N=8$  and 10 f/s) as reference for later analysis.

The block diagram of the receiver is shown in Figure 4.18. The received signal is decoded using MJPEG/SMOKE decoder and displayed on the monitor of the ground station (laptop). At the same time, the received compressed video is stored as MJPEG/SMOKE format for later analysis.

#### 4.5.4 Distortion Measures

The state of the art metric for comparative assessment of video quality is the MOtion-based Video Integrity Evaluation (MOVIE) Index [48]. However, since we experience frame drops in our video transmissions, it is difficult for us to apply the MOVIE index directly. Frame drops result in different video duration between the reference video stored in the transmitter and the video to be evaluated in the receiver. Therefore, we measure distortion in our experiments differently. First, we measure the number of frame drops experienced in the transmission, which represents the temporal loss in the received information. Second, we measure the spatial loss in the video by slicing the video

into images/frames and comparing manually selected, representative best and worst received images/frames to their corresponding source images/frames.

We qualify temporal distortion by comparing the difference in duration between the video viewed at the receiver and the high quality reference video at the transmitter. The high quality original video file is compressed by the encoder at the source while simultaneously being transmitted to the destination. At the destination, a copy of the video received is stored in compressed form while also being viewed in real-time.

In addition, the MPEG2 decoder performs inter-frame estimation for the missing packets in the frame. The estimation process causes some of the frames to be distorted. To capture such frame distortions, we introduce an additional spatial metric. The spatial metric is intended to provide an approximate measure of the distortion in received frames when applied to the best and worst manually selected frames, as will be described in section 4.5.6. We measure the spatial distortion by applying the Structural SIMilarity (SSIM) [52] index to the best and worst frames successfully received/reconstructed at the receiver. The SSIM index is a method for measuring the humanly perceived similarity between two images. It is a reference-based metric, where the assessment of image quality is based on a distortion-free reference image. We choose SSIM because it outperforms traditional methods, such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proven to be inconsistent with human perception.

The SSIM metric is calculated on various windows of an image. The

measure between two windows  $x$  and  $y$  of common size  $N \times N$  is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4.1)$$

where  $\mu_x$  is the average of  $x$ ,  $\mu_y$  is the average of  $y$ ,  $\sigma_x^2$  is the variance of  $x$ ,  $\sigma_y^2$  is the variance of  $y$ ,  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ ,  $c_1 = (k_1L)^2$ ,  $c_2 = (k_2L)^2$  are two variables to stabilize the division with weak denominator,  $L$  is the dynamic range of the pixel-values (typically this is  $2^{\#bits \text{ per pixel}} - 1$ ), and  $k_1 = 0.01$  and  $k_2 = 0.03$  by default.

In order to evaluate image quality, this formula is applied only on *luma*. The resultant SSIM index is a decimal value between -1 and 1, and a value of 1 is only reachable in the case of two identical sets of data.

#### 4.5.5 Experimental Setup

In Horus, we consider two main network topologies:

1. Unicast network topology: one AAV moving in a circular pattern, as shown in Figure 4.19. The AAV is the source that records video of the landscape and transmits this video to the destination. The destination is a laptop in the ground station running Linux.
2. Multiple unicast network topology: two AAV moving in two separate circular paths, as shown in Figure 4.20. The AAVs are both sources that record video of the landscape and transmit them to one destination (as shown in Figure 4.1). The destination is a laptop in the ground station running Linux.



Figure 4.19: Unicast network topology.

Note that a multicast network topology is easier to implement than a multiple unicast network topology. Two sources transmitting to one destination requires twice the bandwidth of a network in which one source transmitting to two destinations. Therefore, if a multiple unicast network topology is implementable and reliable using our proposed algorithms, then multicast network topology will almost certainly be implementable and reliable.

We run the flight experiment at Lester Field [3] in Austin, Texas. Figure 4.21 (taken from Google maps) shows the relative viewing area in relation to the actual AAV path in the location used to run our experiments. In the flight



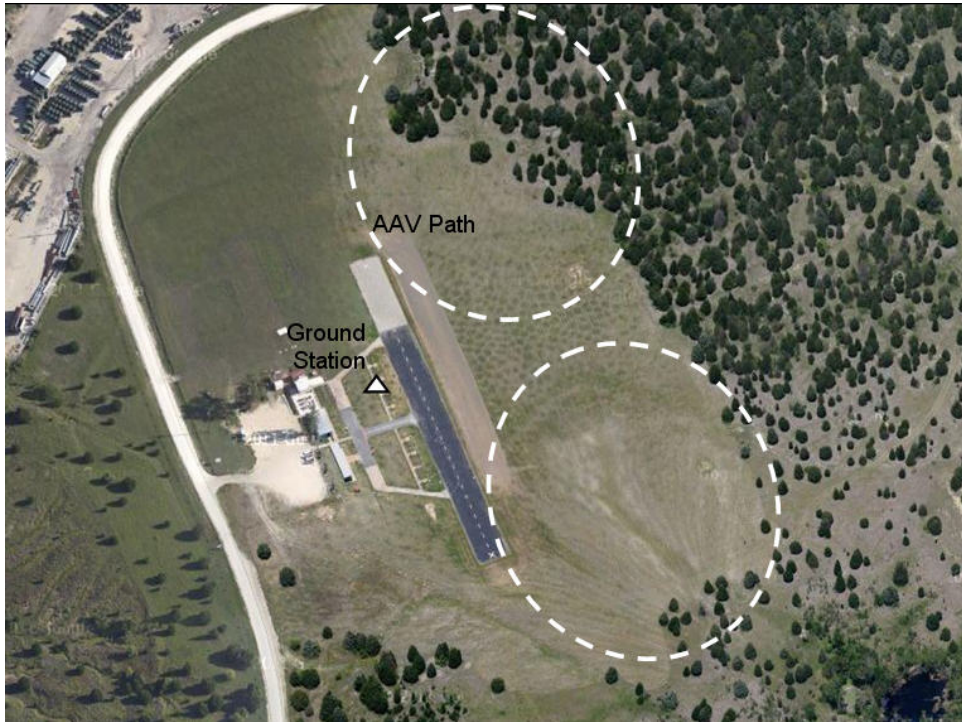


Figure 4.20: Multiple unicast network topology.

experiments, the AAV records video of the landscape and transmits this video in real-time to the ground station (laptop). Throughout each flight, AAVs pass through four different flight phases, as shown in Figure 4.22, which correspond to different characteristics of the transmitted video:

- A) The AAV is stationary on the ground and near the laptop (the ground station). This phase takes place during AAV initialization and preparation.
- B) The AAV is moving slowly and within close range of the laptop. This phase takes place when moving the AAV to the take-off runway.
- C) The AAV is moving relatively fast and with moderate range of the laptop.

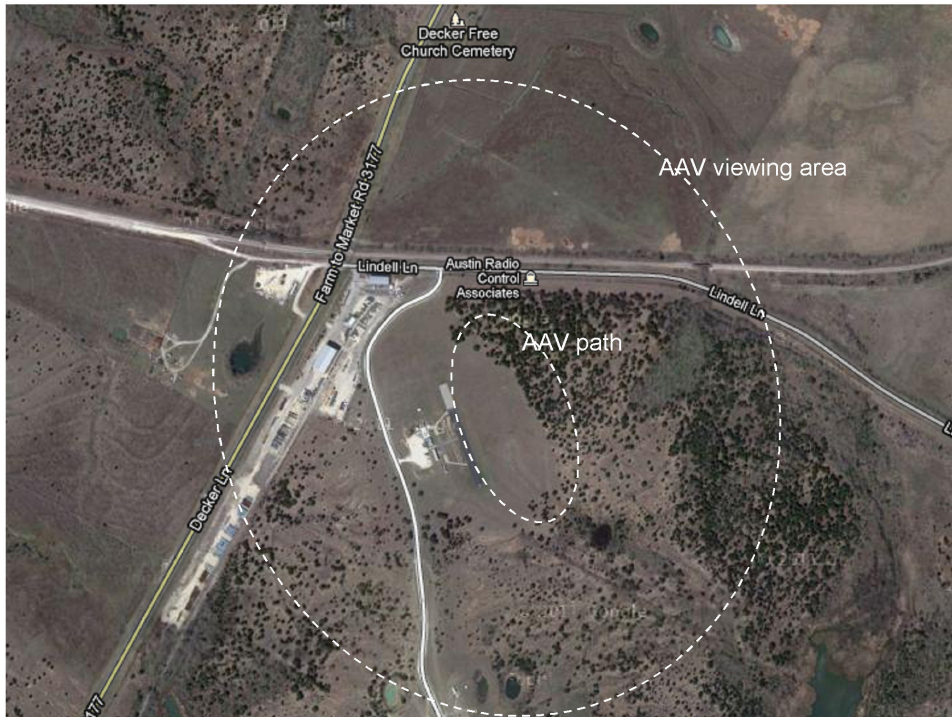


Figure 4.21: Map of the actual site used for running Horus experiments. The internal circle shows the AAV flight path. The outer circle shows the viewing area of the camera attached to the AAV.

This phase takes place at take-off and landing. In this phase, the recorded camera video is changing rapidly.

D) The AAV is moving fast and far away from the laptop. This phase takes place when the AAV is in the air.

The AAV passes through these phases every time we run a flight experiment to test different transmission algorithms. In total, we conducted three different flight experiments:

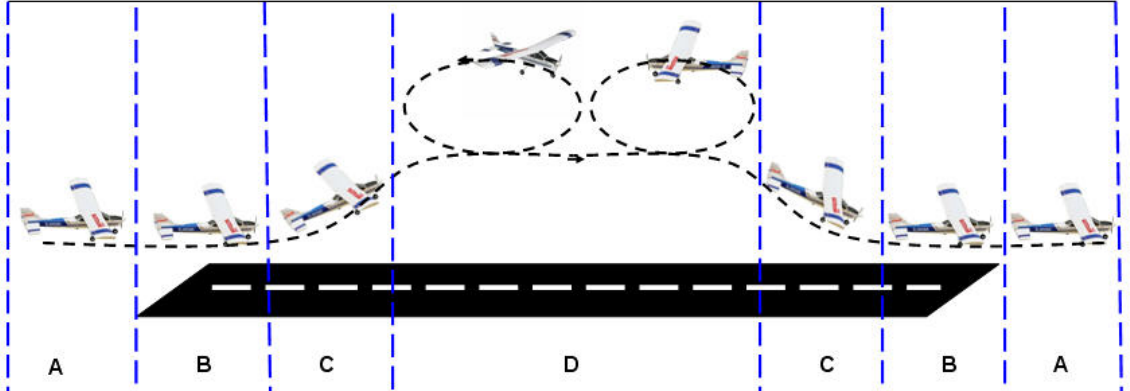


Figure 4.22: Flight phases used to calculate average SSIM index.

1. Evaluating the performance of the LCRDO-Beacon algorithm with respect to unmodified MPEG2 video transmissions. We test the following:
  - (a) *Unmodified transmitter with only  $i$  frames*: MPEG2 codec with GOF=1 and 5 f/s.
  - (b) *Unmodified transmitter with all  $i$ ,  $p$  and  $b$  frames*: MPEG2 codec with GOF=5 and 25 f/s.
  - (c) *Transmitter running the LCRDO-Beacon algorithm*: alternating MPEG2 codecs as discussed in section 4.5.1.
  
2. Evaluating the performance of the LCRDO-Adaptive algorithm for MPEG2 with respect to unmodified MPEG2 video transmissions. We test the following:
  - (a) *Unmodified transmitter with 100 kbps*: MPEG2 codec with 100 kbps transmission rate and GOF=5 and 25 f/s.

- (b) *Unmodified transmitter with unlimited rate:* MPEG2 codec with unlimited rate and GOF=5 and 25 f/s.
  - (c) *Transmitter running the LCRDO-Adaptive algorithm for MPEG2:* adaptive MPEG2 codec as discussed in section 4.5.2.
3. Evaluating the performance of the LCRDO-Adaptive algorithm for MJPEG/SMOKE with respect to unmodified MJPEG/SMOKE video transmissions. We test the following:
- (a) *Unmodified transmitter with low quality:* MJPEG /SMOKE codec with low quality of 30% and  $N=8$ .
  - (b) *Unmodified transmitter with high quality:* MJPEG /SMOKE codec with high quality of 80% and  $N=8$ .
  - (c) *Transmitter running the LCRDO-Adaptive algorithm for MJPEG:* adaptive MJPEG/SMOKE codec as discussed in section 4.5.3.

*Remark 4.5.1.* The hysteresis threshold values that we use in all the flight experiments are  $X_1=30$  and  $X_2=50$ .

#### 4.5.6 Results

For our experiments, we analyze both temporal and spatial distortion. For temporal distortion, we compare the video at the receiver/destination and the video at the transmitter/source with respect to video duration. For spatial distortion, we compare the SSIM index of 10 highest and 10 lowest spatially distorted video frames received at the destination. The frame selection is done

Table 4.2: Received video duration for temporal distortion

Experiment	video duration
1(a) Send only i frames	68 %
1(b) Send all i, p, and b frames	53 %
1(c) LCRDO-Beacon	76 %
2(a) Send at 100kbps	71 %
2(b) Send at unlimited rate	58 %
2(c) LCRDO-Adaptive for MPEG2	76 %
3(a) Send at low quality	43 %
3(b) Send at high quality	27 %
3(c) LCRDO-Adaptive for MJPEG	49 %

manually by eye inspection. The number of frames selected from the flight phases A, B, C, and D are 2, 4, 4, and 10, respectively. The SSIM index is calculated for each frame with respect to the corresponding source frame using equation 4.1. Then, the average, i.e. arithmetic mean, of SSIM indices for all the selected 20 frames is calculated for each case in each flight experiment.

Table 4.2 and Figure 4.23 show the temporal distortion results for our experiments. For every experiment, the percentage of the received video duration is highest for case (c). Hence, we can conclude that a better optical flow of received video is observed when applying the LCRDO algorithms compared to the other two cases for each experiment. Therefore, the temporal distortion observed by the user is minimal in the LCRDO algorithms. This corresponds to our experiences during real-time field observation.

*Remark 4.5.2.* When we run the same case in the same experiment under the same conditions multiple of times, we observe a 5% difference in the results. Therefore, we can conclude that all the results presented here have an accuracy

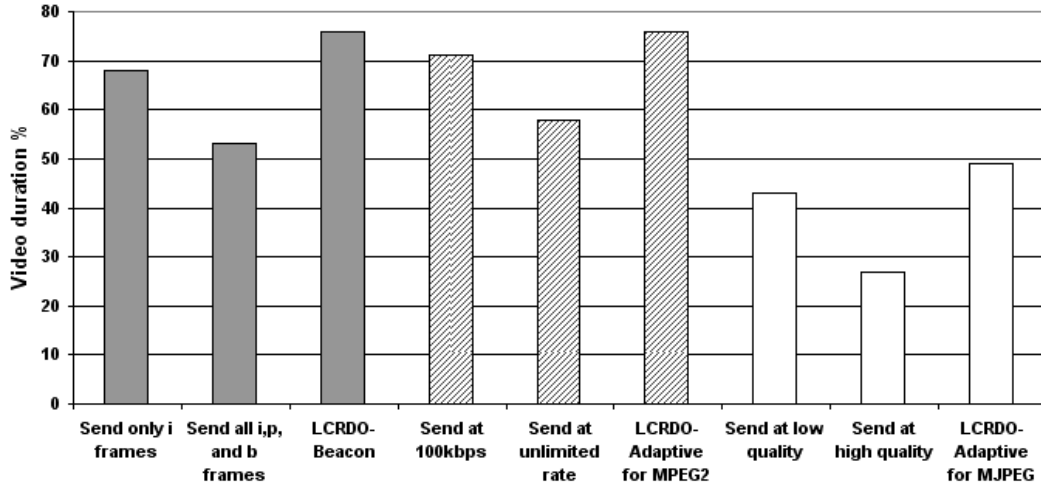


Figure 4.23: Received video duration for temporal distortion.

around 95%.

Table 4.3: Average SSIM indices for spatial distortion

Experiment	Phases				Overall average SSIM
	A	B	C	D	
1(a) Send only i frames	1	0.72895	0.902525	0.69312	0.772855
1(b) Send all i, p, and b frames	1	0.839925	0.84155	0.66746	0.770025
1(c) LCRDO-Beacon	1	0.679675	0.727875	0.79116	0.77709
2(a) Send at 100kbps	0.9686	0.71815	0.75375	0.65156	0.71702
2(b) Send at unlimited rate	1	0.765425	0.90285	0.71039	0.78885
2(c) LCRDO-Adaptive for MPEG2	1	0.841275	0.7911	0.67154	0.762245
3(a) Send at low quality	0.93955	0.948525	0.943925	0.93317	0.93903
3(b) Send at high quality	1	1	1	1	1
3(c) LCRDO-Adaptive for MJPEG	1	1	0.931575	0.94515	0.95889

In terms of spatial distortions, we observed that the average SSIM indices for case (a), (b), and (c) within each experiment are approximately equal, see Table 4.3. We conclude that the average SSIM index is more dependent on the compression type compared to the transmission algorithm. Nevertheless, spatial distortions in the RDO case are always less than when streaming with a low quality encoder, approaching the level of high quality encoding yet with much better temporal behavior. The average SSIM indices for the proposed RDO algorithms is summarized in Table 4.3 and Figure 4.24. The MJPEG/SMOKE codec outperforms MPEG2 codec in terms of its average SSIM index. This is due to inter-frame estimation performed in the MPEG2 decoder, which causes partially received frames to be viewed as distorted frames. By contrast, partially received frames are dropped in the MJPEG/SMOKE decoder, leading to higher temporal distortions instead. Some samples of observed frame distortions are shown in Figure 4.5.6.

Finally, we conducted a flight experiment for the multiple unicast network topology mentioned in 4.5.5. The two AAVs transmit two different video signals simultaneously to a common ground station (laptop) using the LCRDO-Adaptive for MJPEG algorithm. The main observation is that we can display both videos with acceptable optical flow. Successful reception of multiple unicast flows should open more possibilities for future work in building larger AAV networks that can transmit/receive videos to/from multiple destinations/sources. As discussed previously, any implementation of the more complex multiple unicast case should easily transfer into a multicast environ-

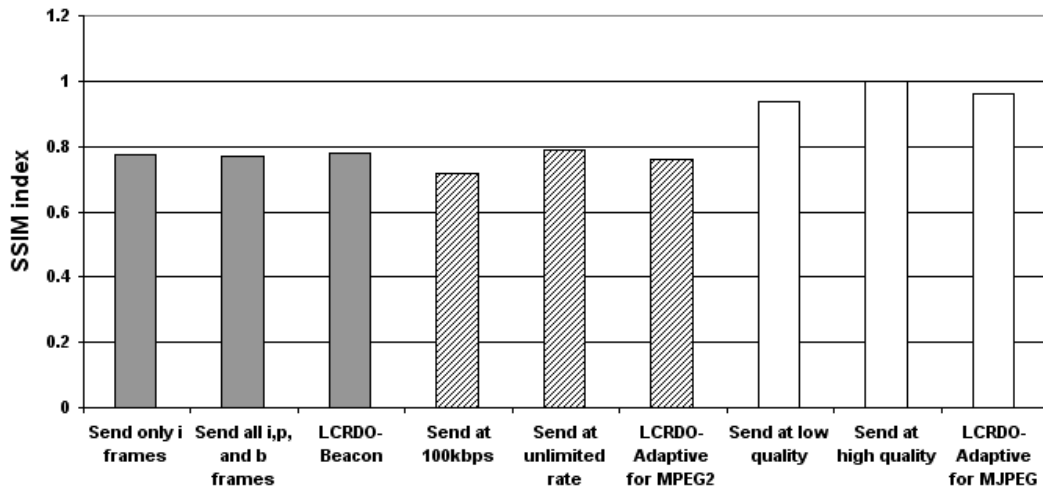


Figure 4.24: Average SSIM indices for spatial distortion.

ment.

A summary of the comparison between the three proposed RDO algorithms is shown in Table 4.4.



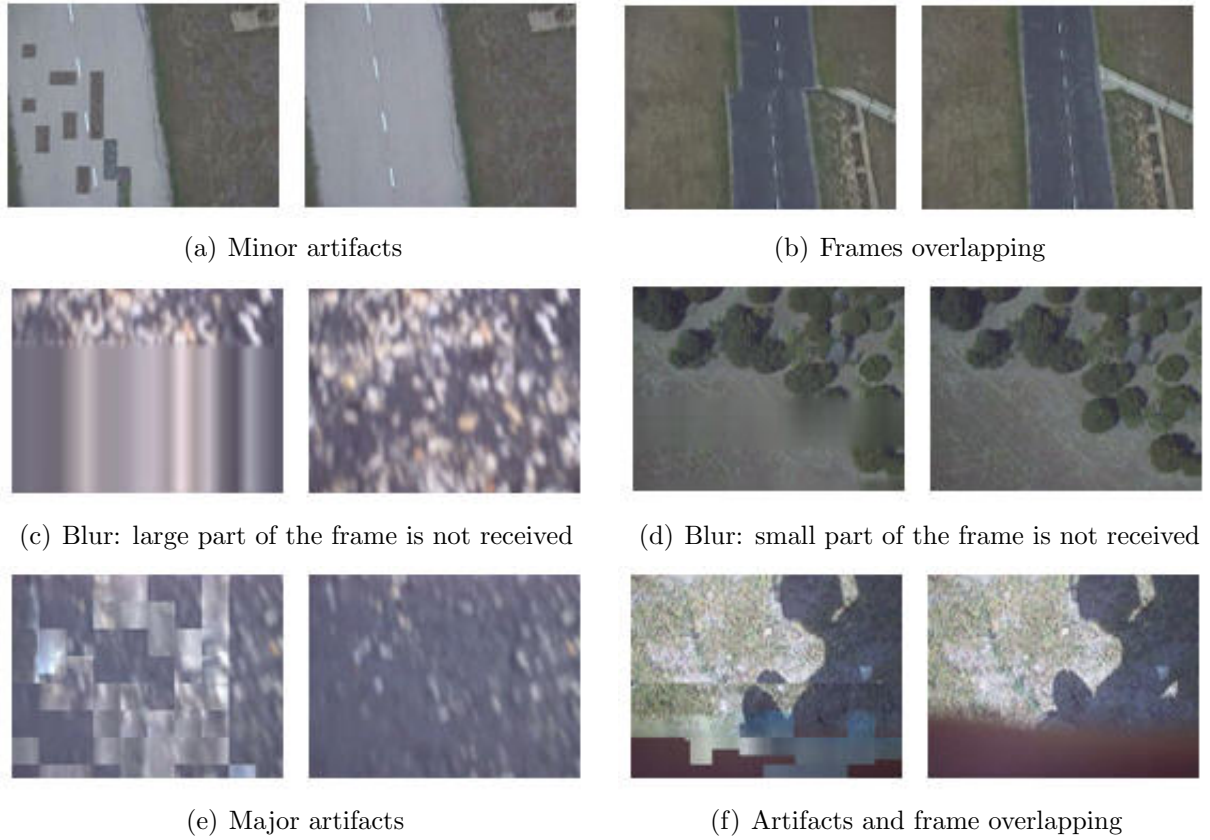


Figure 4.25: Samples of spatial frame/image distortion (frame from source video is on the right and frame from destination video is on the left).

Table 4.4: Comparison between proposed RDO algorithms

Comparison points	LCRDO-Beacon	LCRDO-Adaptive for MPEG2	LCRDO-Adaptive for MJPEG
Video optical flow	average	average	above average
Compression ratio	moderate	moderate	low
Bandwidth requirements	moderate	moderate	high
Computation requirements	moderate	moderate	low
Compatibility with Horus	moderate	moderate	best
Observed distortion intensity	apparent	apparent	not apparent (negligent)
Observed video duration	76%	76%	49 %
Observed video lag	high	high	moderate
SSIM index	0.78	0.76	0.96
Types of observed distortion, see Figure 4.5.6	artifacts, frame overlapping, blur (missing part of frame)	artifacts ,frame overlapping, blur (missing part of frame)	minor artifacts
Video reconstruction	inter-frame estimation	inter-frame estimation	independent frames
Overall observed quality	moderate	moderate	high
Observed optical flow	moderate	moderate	high

## Chapter 5

### Conclusion

This dissertation considers fundamental theoretical and practical research for multicast wireless networks due to the importance of these networks for more efficient wireless communications. Towards achieving this goal, first, we aim our research to better understand the capacity of multicast networks. In particular, our results provide both new capacity scaling laws and outer-bound for two hop networks. We analyze the impact of mobility on multicast communication in a wireless network. We establish that the growth-rate of the throughput in the all-mobile multicast network is order-wise higher compared to the all-static multicast network. We pair an interference model (either protocol or physical) with an unrestricted mobility model, and determine mechanisms for transferring information from source to destination. We establish that mobility enhances the order of growth of throughput of the hybrid network when the order of growth of mobile nodes in the network is greater than a threshold.

Next, we propose research on the alignment schemes for equal-length multihop multicast networks. This is important to fully utilize the wireless channel. We investigate the achievable rates for cases where number of relay

nodes are less and more than source-destination pairs. We determine alignment mechanisms for two-hop communication for multicast networks when the number of relays exceeds the number of sources/destinations. We combine elements of the two-hop and ergodic alignment when the number of relays is smaller than the number of sources/destinations.

In addition, this work constructs a reliable wireless networks test-bed for testing new wireless networks protocols called Horus. In this test-bed, we implement the problem of streaming packetized media over a wireless network using a rate-distortion optimized algorithm. We compare different transmission algorithms both in simulation and implementation. We give a comparative study to the design trade-offs to be considered to achieve a reliable and optimized video transmission. The main intuition that emerges from Horus is that in order to provide a good real-time video transmission performance, one should consider both the computation power and the bandwidth limitations. For low computation power, we find that MJPEG/SMOKE as the best choice. For moderate computation power, we have MPEG2 as the more suitable solution. For limited bandwidth but high computation power, MPEG4-H264 could potentially be the more suitable solution because it has better utilization of the available bandwidth compared to MJPEG/SMOKE and MPEG2. This is the trade-off between bandwidth and computation limitation and the video quality that can be achieved.

## Appendices

# Appendix A

## Proofs of Chapter 2

### A.1 Proof of Lemma 2.4.1

Divide the unit square into  $\frac{1}{\sqrt{m}} \times \frac{1}{\sqrt{m}}$  square grid. Next, we consider the probability that at least  $c_1 m$  squares are empty. Let  $\bar{c}_1 = 1 - c_1$ . Using the union bound and the Stirling's approximation, we get

$$\begin{aligned} \Pr \{c_1 m \text{ or more squares empty}\} &\leq \frac{m!}{(c_1 m)! (\bar{c}_1 m)!} \bar{c}_1^m \\ &\leq \frac{(1 + O(1/m))}{\sqrt{2\pi m c_1 \bar{c}_1}} \left(\frac{\bar{c}_1}{c_1}\right)^{c_1 m} \\ &\rightarrow 0 \text{ as } m \rightarrow \infty \end{aligned}$$

if  $c_1 > 1/2$ . Now, from all the filled (non-empty) squares, we can pick at least  $1/9$  of these such that they are at least  $2/\sqrt{m}$  apart. We form the set of nodes by picking one node from each of such filled squares. This set has at least  $\bar{c}_1 m/9$  nodes that are at least  $2/\sqrt{m}$  apart. This completes the proof with  $K_1 = \bar{c}_1/9 > 0$  and  $K_2 = 2 > 0$ .

### A.2 Proof of Lemma 2.4.2

It is clear that at least a constant fraction  $K_3$  ( $1/4$ ) of the area  $A_k$  is within  $A$ . Therefore, edge effects can be easily handled by considering nodes

in this fraction of the area alone. Let  $|A_k|$  denote the number of nodes in this fraction of  $A_k$  other than the multicast source  $S_k$ , and  $K_4 = \frac{\pi K_2^2 K_3}{(2+\Delta)^2}$ . Note that  $|A_k|$  is binomial distributed with (at least)  $n - n_s$  number of trials and success probability of  $\frac{K_4}{n_s}$ . Using Chernoff's inequality applied to binomial distribution at  $\frac{K_4(n-n_s)}{2n_s}$ , we obtain

$$\Pr \left\{ |A_k| < \frac{K_4}{2} \left( \frac{n}{n_s} - 1 \right) \right\} \leq \exp \left( -\frac{K_4}{8} \left( \frac{n}{n_s} - 1 \right) \right).$$

Hence,

$$\begin{aligned} \Pr \left\{ \bigcap_k |A_k| \geq \frac{K_4}{2} \left( \frac{n}{n_s} - 1 \right) \right\} \\ &= 1 - \Pr \left\{ \bigcup_k |A_k| < \frac{K_4}{2} \left( \frac{n}{n_s} - 1 \right) \right\} \\ &\geq 1 - K_1 n_s \exp \left( -\frac{K_4}{8} \left( \frac{n}{n_s} - 1 \right) \right) \\ &\rightarrow 1 \text{ as } n \rightarrow \infty \end{aligned}$$

if  $n/n_s \rightarrow \infty$ . This completes the proof.

### A.3 Proof of Lemma 2.4.3

As mentioned before, the edge effects can be handled by looking at the constant  $K_3$  fraction of the area within the area  $A$ . Let  $R_k$  denote the location of these selected nodes,  $B_k$  be the circular area with  $R_k$  as the center and  $\frac{K_2}{(2+\Delta)\sqrt{\bar{n}}}$  as the radius, and  $K_4 = \frac{\pi K_2^2 K_3}{(2+\Delta)^2}$ .

The total number of destination nodes in  $\bigcup_k B_k$  is binomial distributed with  $\bar{n}$  number of trials and success probability of  $K_4$ . Applying Chebyshev's

inequality, we have

$$\Pr \left\{ \sum_k |B_k| < \frac{K_4 \bar{n}}{2} \right\} \leq \frac{4(1 - K_4)}{\bar{n} K_4} \rightarrow 0 \quad \text{as } n \rightarrow \infty.$$

In order to upper bound the number of selected nodes such that  $|B_k| = 0$ , we can upper bound the number of empty bins when  $k = K_4 \frac{\bar{n}}{2}$  balls are thrown randomly into  $l = K_1 \bar{n}$  ( $l = \frac{2K_1}{K_4} k$ ) bins. Let  $0 < \delta < 1$  and  $\bar{\delta} = 1 - \delta$ . Applying the union bound and the Stirling's approximation, we have

$$\Pr \{ \delta l \text{ or more bins empty} \} \leq \frac{l!}{(\delta l)! (\bar{\delta} l)!} \bar{\delta}^k \leq \frac{\bar{\delta}^k (1 + O(1/l))}{\sqrt{2\pi l \delta \bar{\delta}} (\delta^{\delta l} \bar{\delta}^{\bar{\delta} l})}.$$

Note that we can choose  $\delta$ , which does not depend on  $n$ , such that  $\log \bar{\delta} < \frac{2K_1}{K_4} (\delta \log \delta + \bar{\delta} \log \bar{\delta})$  for any  $K_1 > 0$  and  $K_4 > 0$ . For such a  $\delta$ , the above probability goes to zero as  $n \rightarrow \infty$ . Therefore,  $\bar{\delta} K_1 n_s n_d$  of the selected nodes can transmit a packet to a destination node w.h.p.

#### A.4 Proof of Lemma 2.4.4

The scheduling policy depends solely on the node locations, which are i.i.d., stationary and ergodic. Hence, all the source-destination pairs obtain the same long-term rate. Due to ergodicity, the long-term throughput between any source and any of its destinations is equal to the expected number of source-destination transmissions (directly or through relay nodes) that are scheduled. From Lemma 2.4.2 and Lemma 2.4.3, it follows that a multicast rate of  $\Omega(1)$  is achieved by all the sources.



## A.5 Proof of Lemma 2.4.5

It follows from the description of the scheme that the achievable throughput (for each multicast source) in this setting is  $1/\lceil n_s n_d/n \rceil$  times smaller compared to the setting with  $n_s n_d = n$ . This completes the proof.

## A.6 Proof of Lemma 2.5.1

Based on (2.4) and (2.5) related to the physical model, we define the received power by node  $V$  from  $S_k$  as  $Q_k = |S_k - V|^{-\alpha_g}$ . We are interested in the SINR for this node, which is

$$\text{SINR} = \frac{\max_k Q_k}{N_0 + I}$$

where  $I$  is the sum of interference from all other sources given by  $I = \sum_k Q_k - \max_k Q_k$ .

Conditioning on  $V = v$ , we have the probability of received power greater than  $q$ , for large  $q$ , at node  $V$  is given by

$$\begin{aligned} \Pr\{Q_k > q | V = v\} &= \Pr\left\{|S_k - V| < q^{\frac{-1}{\alpha_g}} | V = v\right\} \\ &= \Pr\left\{|S_k - v| < q^{\frac{-1}{\alpha_g}}\right\} = \pi q^{\frac{-2}{\alpha_g}}. \end{aligned}$$

Note that conditioning on  $V = v$ ,  $Q_k$  are i.i.d. random variables. From the above calculation, each random variable  $Q_k \equiv Q$  conditioned on random variable  $V \equiv v$  has conditional CDF satisfying

$$\lim_{x \rightarrow \infty} \frac{1 - F_{Q|V}(x)}{1 - F_{Q|V}(lx)} = l^{2/\alpha_g}.$$

Let  $Q_{(1)}$  denote the extremum of all  $Q_k$ . From results of asymptotic distribution of extremum of i.i.d. random variables [30], we have

$$\lim_{m \rightarrow \infty} \Pr \{ Q_{(1)} \leq (\pi m)^{\alpha_g/2} x | V = v \} = \exp(-x^{-2/\alpha_g}).$$

Now, since the right hand side of the above equation is independent of  $v$  and using dominated convergence theorem, we have  $\lim_{m \rightarrow \infty} \Pr \{ Q_{(1)} \leq (\pi m)^{\alpha_g/2} x \} = \exp(-x^{-2/\alpha_g})$ . It follows that  $(\pi m)^{-\alpha_g/2} Q_{(1)} \rightarrow Q_\infty$  as  $m \rightarrow \infty$ , where  $Q_\infty$  has CDF given by  $F_{Q_\infty}(x) = \exp(-x^{-2/\alpha_g})$ ,  $x \geq 0$ .

The second random variable in SINR is the interference  $I = \sum_k Q_k - \max_k Q_k$ . We know that, for large  $q$ ,  $Q_k$  satisfies  $\Pr \{ Q_k > q | V = v \} = \pi q^{-\frac{2}{\alpha_g}}$  and  $Q_k$  are i.i.d. random variables given  $V = v$ . Using the Theory of Stable Random Variables [31], we have

$$\left( \pi \Gamma \left( 1 - \frac{2}{\alpha_g} \right) m \right)^{-\alpha_g/2} I \rightarrow I_\infty \text{ as } m \rightarrow \infty,$$

where  $\Gamma(s) = \int_0^\infty x^{s-1} e^{-x} dx$  is the gamma function, and  $I_\infty$  has CDF given by  $F_{I_\infty}(x) = \exp(-x^{-2/\alpha_g})$ ,  $x \geq 0$ .

Finally, we calculate SINR in the asymptotic limit using the above tail distributions. We have

$$\begin{aligned} \Pr \{ \text{SINR} > \zeta \} &= \Pr \left\{ \frac{Q_{(1)}}{N_0 + I} > \zeta \right\} \\ &\rightarrow \Pr \left\{ \frac{Q_\infty}{I_\infty} > \zeta_\infty \right\} > 0 \text{ as } m \rightarrow \infty, \end{aligned}$$

where  $\zeta_\infty = \zeta \left( \Gamma \left( 1 - \frac{2}{\alpha_g} \right) \right)^{\alpha_g/2}$ .

## A.7 Proof of Lemma 2.5.2

Divide the unit square into  $\frac{1}{\sqrt{m}} \times \frac{1}{\sqrt{m}}$  square grid. We consider the probability that at least  $c_2 m$  squares have exactly one transmitter node. Next, we show that there exists  $c_2 > 0$  such that this probability is greater than  $c_3 > 0$ , i.e., non-vanishing.

Consider the problem of throwing  $l$  balls randomly into  $l$  bins. The probability described above is equal to the probability of having at least  $c_2 l$  bins with exactly one ball each. Let  $k = \{1, 2, \dots, l\}$  denote the  $l$  bins,  $I_k$  be the indicator function whether bin  $l$  has exactly one ball or not, and  $X$  be the number of bins with exactly one ball each. We have, for any  $k$ ,

$$\mathbf{E}[I_k] = \Pr\{\text{bin } k \text{ has exactly one ball}\} = \left(1 - \frac{1}{l}\right)^{l-1}.$$

Since  $X = \sum_k I_k$ , we have

$$\mathbf{E}[X] = l \left(1 - \frac{1}{l}\right)^{l-1}.$$

Let  $Y = l - X$  and  $\bar{c}_2 = 1 - c_2$ . Applying Markov's inequality to non-negative random variable  $Y$ , we get

$$\begin{aligned} \Pr\{X \leq c_2 l\} &= \Pr\{Y \geq \bar{c}_2 l\} \\ &\leq \frac{\mathbf{E}[Y]}{\bar{c}_2 l} \\ &= \frac{1 - \left(1 - \frac{1}{l}\right)^{l-1}}{1 - c_2} \\ &\rightarrow \frac{1 - e^{-1}}{1 - c_2} \quad \text{as } l \rightarrow \infty. \end{aligned}$$

We can choose  $c_2 = (2e)^{-1} > 0$  and  $c_3 = \frac{1}{2e-1} > 0$ , such that  $\Pr\{X > c_2 l\} \geq c_3$ .

Now, among all these transmitter nodes, which belong to squares with exactly one transmitter node each, there are at least  $\frac{1}{9}$  of these with the following property. For this transmitter node, all receiver nodes that are within a distance of  $\frac{2}{\sqrt{m}}$  is nearer to it than all other transmitter nodes. With this result, the proof for  $\beta > 1$  follows from the proof of Lemma 2.4.2 and the proof for  $\beta = 1$  follows from the proof of Lemma 2.4.3.

# Appendix B

## Proofs for Chapter 3

### B.1 Proof of Lemma 3.3.1

We state the proof here for  $K_2 = K$ , but it easily straightforwardly extends to cases where  $K_2 > K$ . Recall from [37, 39] that for every full rank square matrix  $\mathbf{H}_1$  there exists a unique inverse matrix  $\mathbf{H}_1^{-1}$  which is full rank and square such that  $\mathbf{H}_1^{-1}\mathbf{H}_1 = \mathbf{I}$  where  $\mathbf{I}$  is the Identity matrix. For the given matrix  $\mathbf{L}_K$ , multiply both sides from left by  $\mathbf{L}_K$  so we have  $(\mathbf{L}_K\mathbf{H}_1^{-1})\mathbf{H}_1 = \mathbf{L}_K$  and let  $\mathbf{H}_2 = \mathbf{L}_K\mathbf{H}_1^{-1}$ . Recall [37] that for any matrix  $\mathbf{A}$  and  $\mathbf{B}$  we know that

$$\text{rank}(\mathbf{AB}) \leq \min(\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B}))$$

Under the given conditions that  $(-1)^K \prod_{i=1}^K l_i \neq 1$  and  $\mathbf{H}_1$  is full rank, we have  $\text{rank}(\mathbf{H}_1) = \text{rank}(\mathbf{L}_K) = K$  and so from the previous inequality we get

$$K \leq \min(K, \text{rank}(\mathbf{H}_2))$$

Therefore  $\text{rank}(\mathbf{H}_2) \geq K$  and since  $\mathbf{H}_2$  is a  $K \times K$  matrix. Then  $\mathbf{H}_2$  must be full rank and unique.

## B.2 Proof of Corollary 3.3.2

The statement follows directly from the fact that the addition of more than  $K$  nodes to the relay layer, increases the average rank of  $\mathbf{H}_1$  and  $\mathbf{H}_2$ . Therefore, we have

$$\Pr(\text{rank}(\mathbf{H}_1) = i | K_2 > K) \geq \Pr(\text{rank}(\mathbf{H}_1) = i | K_2 = K) \quad \forall i$$

therefore we have

$$\mathbf{E}(\text{rank}(\mathbf{H}_1) | K_2 > K) \geq \mathbf{E}(\text{rank}(\mathbf{H}_1) | K_2 = K)$$

The result follows.

## B.3 Proof of Theorem 3.3.3

This theorem is a straightforward consequence of Lemma 3.3.1.  $K$  simultaneously multiple access channels (MACs) can be sustained from the sources to the destinations. This, along with random coding arguments based on [29] gives us this result.

## B.4 Proof of Theorem 3.3.4

We combine time instances (equivalently, perform symbol extension) to transform the system into an equivalent full rank channel transformation. This is very similar to alignment schemes. Let  $\mathbf{H}_1(1), \dots, \mathbf{H}_1(p)$  denote  $p$  time instances of the channel  $\mathbf{H}_1$  which collectively are of rank  $K$ . Now, choose

$$[\mathbf{H}_2(1) \ \mathbf{H}_2(2) \ \dots \ \mathbf{H}_2(p)]^T [\mathbf{H}_1(1) \ \mathbf{H}_1(2) \ \dots \ \mathbf{H}_1(p)] = \mathbf{L}_K.$$

Note that such a collection of channel states  $\mathbf{H}_2$  is unique as given by Lemma 3.3.1. Moreover, as each channel state  $\mathbf{H}_2$  occurs equally often, such a collection of channel states is possible, and thus, over  $p$  time instances of the channel,  $K$  simultaneous multiple access channels can be supported between the sources and destinations. This grants us the result.

## Bibliography

- [1] 80A Phoenix ICE HV speed controller with Data Logging Capability, <http://www.truerc.com/index.php-main-page=product-info-cPath-15-27-products-id=242>.
- [2] ArduPilot Mega, <http://diydrones.com/profiles/blogs/ardupilot-mega-home-page>.
- [3] Austin RC, <http://www.austinrc.org/index.html>.
- [4] CM9-GP: 802.11 a/b/g 108Mbps wifi mini-PCI module, MB42/AR5213A+AR5112, <http://www.unex.com.tw/product/cm9-gp>.
- [5] GStreamer: open source multimedia framework, <http://gstreamer.freedesktop.org/>.
- [6] Hacker Brushless A40-10L, <http://www.aero-model.com/Hacker-Brushless-A40-10L.aspx>.
- [7] Hitec Eclipse 7 7-Channel FM Transmitter/Spectra, <http://www.scribd.com/doc/16070675/Hitec-Eclipse-7-Transmitter-Manual>.
- [8] Horus: A Wireless Network of AAVs, <http://theseus.ece.utexas.edu/horus/>.
- [9] HS-322 Deluxe Standard Servo, <http://www.hobby-lobby.com/hs-322-standard-servo-2786-prd1.htm>.



- [10] Logitech QuickCam Pro 4000, <http://www.amazon.com/Logitech-961239-0403-QuickCam-Pro-4000>.
- [11] MediaTek MT3329 GPS 10Hz, [http://store.diydrones.com/MediaTek - MT3329-GPS-10Hz-p/mt3329-01.htm](http://store.diydrones.com/MediaTek-MT3329-GPS-10Hz-p/mt3329-01.htm).
- [12] MiXiM OMNeT++ Framework Project, <http://mixim.sourceforge.net/>.
- [13] OMNeT++ Network Simulation Framework, <http://www.omnetpp.org/>.
- [14] Smoke Encoder, [http://www.flumotion.net/doc/flumotion /manual /en /0.6.0/html/](http://www.flumotion.net/doc/flumotion-manual-en/0.6.0/html/).
- [15] The Click Modular Router Project, <http://read.cs.ucla.edu/click/click>.
- [16] Thunder Power 3S4P 8000 mAh Li-Poly Packs TP8000-3S4P, [http:// aeromicro.com/catalog/thunder-power-3s4p -8000-mah-li-poly- packs-tp8000-3s4p-2660251.htm](http://aeromicro.com/catalog/thunder-power-3s4p-8000-mah-li-poly-packs-tp8000-3s4p-2660251.htm).
- [17] Via EPIA Nano-ITX, [http://www.via.com.tw/en/products/embedded/ ProductDetail.jsp- motherboard -id-470](http://www.via.com.tw/en/products/embedded/ProductDetail.jsp-motherboard-id-470).
- [18] XBee-PRO DigiMesh 900 Mesh RF Modules, [http://www.digi.com/products /wireless/zigbee - mesh/xbee-digimesh-900](http://www.digi.com/products/wireless/zigbee-mesh/xbee-digimesh-900).
- [19] S. O. Gharan A. Motahari and A. K. Khandani. On the degrees of freedom of the 3-user Gaussian interference channel: the symmetric case. *Proc. IEEE ISIT*, 2009.

- [20] A. Abdel-Hadi, J. Micheal, A. Gerstlauer, and S. Vishwanath. Real-time optimization of video transmission in a network of aavs. *IEEE 74th Vehicular Technology Conference, San Francisco, United States*, 2011.
- [21] A. Abdel-Hadi and S. Vishwanath. On multicast interference alignment in multihop networks. *IEEE Information Theory Workshop, Cairo, EGYPT*, 2010.
- [22] S. Aeron and S. Venkatesh. Capacity scaling in wireless ad hoc networks with  $p_e$ . *Proc. IEEE Symp. Inform. Theory*, page 469, June 2004.
- [23] D. Agrawal, T. Bheemarjuna Reddy, C. Siva, and Ram Murthy. Robust Demand-Driven Video Multicast over Ad hoc Wireless Networks. In *Proc. of BroadNets '06*, October 2006.
- [24] S. Jafar B. Nazer, M. Gastpar and S. Vishwanath. Ergodic interference alignment. *Proc. IEEE ISIT*, 2009. [arxiv.org/abs/0901.4379](http://arxiv.org/abs/0901.4379).
- [25] S. Jafar B. Nazer, M. Gastpar and S. Vishwanath. Interference alignment at finite snr: General message sets. *Proc. Allerton Conf. on Commun. Control and Computing*, 2009.
- [26] V. R. Cadambe and S. A. Jafar. Interference alignment and the degrees of freedom for the K user interference channel. *IEEE Trans. Inform. Theory*, 54(8):3425–3441, Aug. 2008.
- [27] P. Chaporkar and S. Sarkar. Wireless multicast: Theory and approaches. *IEEE Trans. Inform. Theory*, 51:1954–1972, June 2005.

- [28] P. A. Chou and Z. Miao. Rate-Distortion Optimized Streaming of Pack-  
etized Media. *IEEE Transaction on Multimedia*, May 2005.
- [29] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John  
Wiley & Sons, Inc., 2006.
- [30] H. A. David. *Order Statistics*. Wiley, 2nd edition, 1981.
- [31] W. Feller. *An Introduction to Probability Theory and Its Applications*,  
volume II. Wiley, 2nd edition, 1971.
- [32] M. Franceschetti, M.D. Migliore, and P. Minero. The capacity of wire-  
less networks: information-theoretic and physical limits. *IEEE Trans.*  
*Inform. Theory*, 2009. submitted.
- [33] M. Garetto, P. Giaccone, and E. Leonardi. Capacity scaling in delay  
tolerant networks with heterogeneous mobile nodes. In *Proc. ACM*  
*MobiHob*, Sept. 2007.
- [34] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad  
hoc wireless networks. *IEEE/ACM Trans. Networking*, pages 477–486,  
Aug. 2002.
- [35] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE*  
*Trans. Inform. Theory*, 46(3):388–404, Mar. 2000.
- [36] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger. On randomized  
network coding. *Proc. 41st Annual Allerton Conf. on Commun. Control*  
*and Computing*, 2003.

- [37] R. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press.
- [38] P. Jacquet and G. Rodolakis. Multicast scaling properties in massively dense ad hoc networks. *Proc. 11th International Conference on Parallel and Distributed Systems*, 2005.
- [39] S-W. Jeon and S-Y. Chung. Sum capacity of multi-source linear finite-field relay networks with fading. *Proc. Allerton Conf. on Commun. Control and Computing*, 2009.
- [40] J. Jose, A. Abdel-Hadi, P. Gupta, and S. Vishwanath. Impact of mobility on multicast capacity of wireless networks. *IEEE InfoCom, San Diego, CA, USA*, 2010.
- [41] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li shiuan Peh, and Daniel Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Proc. of ACM (ASPLOS-X)*, pages 96–107, October 2002.
- [42] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-demand multicast routing protocol. *Proc. Wireless Communications and Networking Conference (WCNC)*, September 1999.
- [43] X. Li, S. Tang, and O. Frieder. Multicast capacity for large scale wireless ad hoc networks. *Proc. ACM MobiCom*, 2007.

- [44] Ting Liu, Christopher M. Sadler, Pei Zhang, and Margaret Martonosi. Implementing Software on Resource-Constrained Mobile Sensors: Experiences with Impala and ZebraNet. In *Proc. of MobiSYS '04*, pages 256–269. ACM Press, 2004.
- [45] U. Niesen, P. Gupta, and D. Shah. On capacity scaling in arbitrary wireless networks. *Proc. ITA*, January 2008.
- [46] A. Ozgur, O. Leveque, and D. Tse. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Transactions on Information Theory*, 53:3549 – 3572, 2007.
- [47] E. M. Royer and C. E. Perkins. Multicast operation of the ad hoc on-demand distance vector routing protocol. *Proc. ACM MobiCom*, August 1999.
- [48] K. Seshadrinathan and A. Bovik. Motion tuned spatio-temporal quality assessment of natural videos. *IEEE Transactions on Image Processing*, pages 335–350, Feb. 2010.
- [49] S. Shakkottai, X. Liu, and S. Srikant. The multicast capacity of large multihop wireless networks. *Proc. ACM MobiCom*, Sept. 2008.
- [50] P. Sinha, R. Sivakumar, and V. Bharghavan. MCEDAR: Multicast core-extraction distributed ad hoc routing. *Proc. Wireless Communications and Networking Conference (WCNC)*, September 1999.

- [51] S. Sridharan, A. Jafrian, S. Vishwanath, and S. A. Jafar. Capacity of symmetric  $k$ -user Gaussian very strong interference channels. To appear in Globecom Communications Conference, New Orleans, Dec. 2008, arXiv:0808.2314v1, e-print.
- [52] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, pages 600–612, Apr. 2004.
- [53] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. *Proc. IEEE INFOCOM*, March 2000.
- [54] L. L. Xie and P. R. Kumar. A network information theory for wireless communication: scaling laws and optimal operation. *IEEE Trans. Inform. Theory*, 50:748–767, May 2004.
- [55] Pei Zhang, Christopher M. Sadler, Stephen A. Lyon, and Margaret Martonosi. Hardware Design Experiences in ZebraNet. In *Proc. of SenSys '04*, November 2004.

## Vita

Ahmed Mohamed Abdel-Hadi was born in Cairo, Egypt. He received the Bachelor of Science degree in Electrical Engineering from Ain Shams University, Cairo Egypt in 2003, and was assigned a teaching assistant (Mo-aid) position at the university. He received Master of Science degree in Electrical Engineering from Cairo University, Giza Egypt in 2008, and Master of Science degree in Engineering Mathematics from Ain Shams University in 2008. His research interests include wireless networks, embedded systems and system control.

Permanent address: 12349 Metric Blvd.  
Austin, Texas 78758

This dissertation was typeset with  $\text{\LaTeX}^\dagger$  by the author.

---

<sup>†</sup> $\text{\LaTeX}$  is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's  $\text{\TeX}$  Program.