

Video Communications over Dynamic Ad Hoc Networks

Sastry Venkata Subrahmanya Kompella

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Dr. Y. Thomas Hou, Chair

Dr. Scott F. Midkiff

Dr. Luiz A. DaSilva

Dr. Jeffrey H. Reed

Dr. Hanif D. Serali

July 05, 2006

Blacksburg, Virginia

Keywords: Cross-layer design, ad hoc networks, multimedia, multipath routing, multiple
description video, optimization

© Copyright 2006, Sastry V.S. Kompella

Video Communications over Dynamic Ad Hoc Networks

Sastry V.S. Kompella

ABSTRACT

Video communications play a vital role in present and future wireless ad hoc networks. One of the key requirements for a successful deployment of multimedia applications in multihop wireless networks is the ability to provide an acceptable video quality, even under a highly dynamic and perhaps unfriendly (or hostile) environment (e.g., in the presence of frequent node/link failure, interference, shadowing, fading, and so forth). Existing ad hoc routing protocols work well for data communications, but are not optimized for video, which is sensitive to latency and packet loss. Moreover, traditional end system based error control mechanisms alone cannot guarantee a sustainable video quality. Conventional QoS approaches typically optimize one or more network layer metrics, but they are usually agnostic to any kind of application layer performance. Consequently, new methodologies must be explored to improve the performance of video applications in multihop wireless networks.

This dissertation directly addresses this important problem area by leveraging recent advances in video coding techniques along with novel cross-layer formulations and powerful optimization techniques. We follow an application centric cross-layer approach to address multimedia service provisioning over ad hoc networks. Our research efforts show that video communications over multihop wireless networks can substantially benefit from a cross-layer design principle by factoring in application layer video quality into routing algorithmic designs at the network layer. There are three components in this investigation, namely, (1) concurrent routing, (2) path selection and rate allocation, and (3) multipath routing for multiple description video. Each component addresses one or more unique challenges that hinder video communications in multihop wireless networks. Although we expect that a cross-layer approach will be more effective than a network centric (single-layer) approach in addressing application performance, it also brings in complex problems that cannot be effectively solved using traditional methods, and thus, calls for the design of customized

algorithms.

In concurrent routing, we focus on issues that arise while supporting multiple concurrent video communication sessions in an ad hoc network. These sessions compete for limited network resources (such as bandwidth) while interacting with each other. Such inter-session interactions couple the performance of an individual flow with that of other flows. Applying a video centric cross-layer design principle, we model the end-to-end video distortion as a function of network layer behavior, and formulate a network-wide optimal routing problem that minimizes the total video distortion. Results based on computational experiments performed using randomly generated network topologies establish the relative efficacy and robustness of the proposed genetic algorithm based solution approach. Specifically, we demonstrate that our approach outperforms other trajectory based metaheuristic approaches as well as with conventional network centric routing algorithms such as shortest path and disjoint shortest path routing.

The joint path selection and rate allocation problem considers not only selecting the best set of paths for video communication, but also, computing the optimal video encoding rate and partitioning it among the chosen set of paths. The end-to-end video distortion is modeled as a function of network layer resources by capturing the tight coupling that exists between the optimal encoding rate for each video session, the selection of paths for video transmission, and the allocation of traffic among these selected paths. This problem is formulated as a nonlinear nonconvex programming problem, for which a tight linear programming relaxation is constructed via the Reformulation-Linearization Technique (RLT). This construct is embedded within a specialized branch-and-bound algorithm to achieve global optimality. Computational experience is reported for various problem instances, and the results validate the robustness of the proposed algorithmic procedure. The results exhibit the advantage of the solution approach over the popularly used max-min rate allocation scheme.

The emergence of Multiple Description (MD) coding technique offers great potential for multipath routing of video in multihop wireless networks. In studying multipath routing for

MD coding, we show that MD coded video, when used in combination with multipath routing in wireless networks, has tremendous advantages over traditional layered video coding techniques. We discuss how to implement an MD video codec and formulate a cross-layer optimization problem that can find a set of optimal paths, (one for each description) such that the overall video quality at the receiver is maximized. We further devise a specialized RLT-based branch-and-bound solution procedure for the ensuing 0-1 mixed integer non-convex optimization problem. Convergence behavior of the proposed solution procedure is observed for various network topologies and the results further demonstrate the performance advantage of the proposed cross-layer approach over non-cross-layer approaches.

The scope of this research is highly interdisciplinary. It intersects video communication, networking, optimization, and algorithm design. We expect that the theoretical and algorithmic results of this investigation will serve as important building blocks in developing a comprehensive methodology for addressing complex cross-layer problems in the area of wireless ad hoc networks.

To the two most influential women in my life.

My mother who taught me the meaning of life,

and

my wife who supports me in every aspect of my life.

Acknowledgments

With my days as a graduate student drawing to an end, I would like to take this opportunity and acknowledge all the people that made a difference in my Ph.D. life at Virginia Tech. First and foremost, I express my heartfelt thanks to Dr. Thomas Hou for all his help, guidance, and support. I can say with no hesitation that he has had a profound influence on me as a teacher and researcher. I hope that I can remember, and dutifully carry with me, all the valuable lessons that I have learnt from him during the past few years. I also gratefully acknowledge the contributions and help given by the rest of my committee. Specifically, I wish to thank Dr. Scott Midkiff, for the many conversations that we have had and the wonderful advice that I have received from him. I also thank Dr. Luiz DaSilva for his numerous encouragements, and Dr. Jeff Reed for his useful insights. I would like to specially thank Dr. Hanif Sherali for taking a genuine interest in my work serving on my thesis committee. Without his invaluable help and advise both within the classroom and beyond, my research would have left far to be desired from. I have learnt so much from him, and he is truly a role model for me in teaching excellence, research excellence, humility and kindness.

I would like to thank Dr. Scott Midkiff again, for his confidence in me and his efficiency in managing my National Science Foundation Integrative Graduate Education and Research Traineeship (IGERT) fund.

I want to thank Dr. Shiwen Mao for his constant support and due patience during the early days of my Ph.D. I have learnt a great deal from him and he has had a tremendous impact on my research. In addition, I would like to thank the many others who have made

my stay at Virginia Tech a truly memorable one. Few stand out; Animesh Patcha for being a very good friend; Kerry Wood for lending me his ear whenever needed; Xiaolin Cheng for his help and understanding; Yi Shi for his help with optimization. I also thank my fellow CNSR group members and current and past IREAN fellows for their cooperation and assistance.

I am deeply indebted to my brother, soon to be Dr. Ramana Rao Kompella, who was instrumental in my returning to graduate school. Life rarely offers a second chance, and I thank him for making this happen. His constant support and enthusiasm and his passion for research has guided me through many a rough day.

There is no way to adequately thank my wife, Vasudha, for not only supporting my idea of returning to graduate school, but for all that she has done to make my life that much more easier. From moving and finding a job here in Blacksburg, VA, to understanding my need to work in the evenings and weekends, to dealing with me and my necessities, in addition to handling our wonderful son Srihari Kamesh, who can be a handful sometimes. You are the best wife in the world and I am grateful for your love everyday of my life.

Finally, I thank my parents Sri. Kameshwara Rao and Smt. Subbalakshmi for all the love, care, guidance, and support they have always given me. I have always looked up to my dad for his knowledge and wisdom. To sum it up, I have always tried to be like him even though I know it is a difficult task. Thank you dad, for being such a great role model and for always guiding me in the path of life. And mom, I thank you for always believing in my abilities and for the pride you had always taken in each step of my accomplishments. I will always remember and cherish all the conversations we had during my adolescent years. Without your love and all the sacrifice you made in your life, none of my achievements would be possible.

Sponsor Acknowledgements

This research was supported by a National Science Foundation through the Integrative Graduate Education and Research Traineeship (IGERT) grant (award DGE-9987586) titled Integrated Research and Education in Advanced Networking (IREAN). These funds gave me tremendous independence in the pursuit of my ideas, and I am very thankful to NSF for the support.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Cross-layer Design for Video Applications	4
1.3	Problem Scope and Research Contributions	6
1.3.1	Concurrent Routing	6
1.3.2	Path Selection and Rate Allocation	8
1.3.3	Multipath Routing for Multiple Description Video	10
1.4	Outline	11
2	Mathematical Background	12
2.1	Global Optimization	12
2.2	Genetic Algorithms	13
2.2.1	Selection	15
2.2.2	Crossover	16
2.2.3	Mutation	18

2.3	RLT-based Global Optimization for Solving Nonconvex Programming Problems	18
2.3.1	The Reformulation-Linearization Technique	19
2.3.2	Branch-and-Bound	21
3	Routing for Concurrent Video Sessions in Ad Hoc Networks	25
3.1	Introduction	25
3.2	Problem Formulation	27
3.2.1	Network Layer Performance Metrics	29
3.2.2	End-to-End Video Rate-Distortion Model	31
3.2.3	The Global Optimal Routing Problem	32
3.3	A Genetic Algorithm-Based Solution Procedure	34
3.3.1	GA-Based Multiple-Session Routing	35
3.3.2	Determining Video Rates	41
3.4	A Greedy Algorithm For Initial Solutions	42
3.5	Simulation Results	44
3.5.1	Dissecting End-to-End Distortion	45
3.5.2	Performance Bounds	47
3.5.3	Comparison with Trajectory Methods	47
3.5.4	Comparison with Network-centric Routing	48
3.6	Related Work	54
3.7	Summary	56

4	Path Selection and Rate Allocation for Concurrent Video Sessions	58
4.1	Introduction	58
4.2	Problem Formulation	60
4.2.1	Link and Path Statistics	61
4.2.2	Video Distortion	63
4.2.3	Mathematical Formulation	64
4.3	Branch-and-bound Algorithm for Solving Problem OPT-PSRA	66
4.3.1	Reformulation	66
4.3.2	Generating LP Relaxation Using RLT	70
4.3.3	Local Search Algorithm	72
4.3.4	Remarks	73
4.4	Simulation Studies	75
4.4.1	Performance for Various Instances	76
4.4.2	Comparison with A Network-centric Scheme: The Single Path Case	80
4.4.3	Comparison with A Network-centric Scheme: The Multi-path Case	84
4.5	Related Work	87
4.6	Summary	89
5	Multipath Routing for Multiple Description Video	91
5.1	Introduction	91
5.2	Layered Coding versus Multiple Description Coding	93
5.3	Generating MD Video	96

5.4	Problem Formulation	97
5.4.1	Network Model	97
5.4.2	Video Distortion and Path Level Statistics	98
5.4.3	Mathematical Formulation	103
5.5	Reformulation and Linearization	105
5.5.1	Reformulation	106
5.5.2	Linearization	107
5.6	Branch-and-Bound based Solution Procedure	109
5.7	Simulation Studies	113
5.7.1	Convergence Behavior	114
5.7.2	Impact of Description Rates	115
5.7.3	Comparison with Non-Cross-Layer Routing	119
5.8	Related Work	123
5.9	Summary	125
6	Implementation Considerations	126
6.1	General Approach	126
6.2	Wireless Node Architecture	128
6.2.1	Topology Database	128
6.2.2	Link Statistics Database	129
6.2.3	Cross Layer Optimization Module	130
6.3	Routing Protocol Implementation	130

6.3.1	OLSR	131
6.4	Summary	133
7	Summary and Future Work	134
7.1	Summary	134
7.2	Future Research Direction	136
A	Other Metaheuristic Algorithms	138
A.1	Simulated Annealing	139
A.2	Tabu Search	140

List of Figures

1.1	A schematic illustrating battlefield wireless ad hoc network.	2
2.1	A schematic representation of genetic algorithms.	15
2.2	Flow-chart for the GA-based solution procedure.	16
2.3	A schematic representation of the crossover operator.	17
2.4	A schematic representation of the mutation operator.	18
2.5	Illustration of branch-and-bound.	23
3.1	An example wireless ad hoc network and a feasible solution.	36
3.2	Crossover operation.	37
3.3	Impact of crossover rate θ on average distortion.	38
3.4	Mutation operation.	39
3.5	Impact of mutation rate μ on average distortion.	40
3.6	H.263 rate-distortion curve using QCIF “foreman” sequence.	43
3.7	A greedy algorithm for computing initial solutions.	44
3.8	Distortion versus decoding deadline.	46
3.9	GA versus trajectory methods.	49

3.10	Average end-to-end distortion versus decoding deadline.	51
3.11	Total distortion versus number of video sessions.	52
3.12	Average distortion values for each video session in a 10-session, 50-node network obtained by different algorithms.	53
3.13	PSNRs of decoded frames for video session Five.	54
3.14	Reconstructed frame 148 from the “foreman” sequence.	55
4.1	Polyhedral outer approximation for $y = \log(x)$ in $0 < x_0 \leq x \leq 1$	69
4.2	Branch-and-bound and RLT based algorithm for Problem OPT-PSRA.	74
4.3	Correlation among paths in a network.	77
4.4	Randomly generated networks.	78
4.5	Convergence of path selection and rate allocation algorithm for a 50-node network with six video sessions.	80
4.6	Average distortion versus decoding deadline for a 50-node network with 10 sessions and one path per session.	81
4.7	Distortion for individual video sessions in the 50-node network with ten sessions and one path per session	83
4.8	Average distortion versus decoding deadline for a 50-node network with five sessions and two paths per session.	84
4.9	Distortion for individual video sessions in the 50-node network with five sessions and two paths per session	86
4.10	PSNR of reconstructed video frames for video session Two.	86
4.11	Reconstructed frame 160 from the “foreman” sequence for video session Two.	88

5.1	Layered coding versus multiple description coding.	94
5.2	Generating multiple description video.	97
5.3	Link and path models.	104
5.4	Flowchart for the ϵ -optimal solution procedure $ALG(\epsilon)$	109
5.5	Convergence behavior and final optimal solution by the proposed solution procedure for a 50-node network.	116
5.6	Convergence behavior and final optimal solution by the proposed solution procedure for a 100-node network.	117
5.7	Impact of description rate R on $ALG(\epsilon)$ computation time.	118
5.8	Impact of description rate R on average distortion obtained by $ALG(\epsilon)$	119
5.9	Average PSNR values for different description rates.	121
5.10	Examples of paths obtained by $ALG(\epsilon)$ and 2-SP for different description rates.	122
5.11	Frame 278 from the reconstructed video sequences ($R = 128Kb/s, 256Kb/s$).	124
6.1	Wireless node architecture.	129
6.2	OLSR packet header.	131
6.3	Comparison of standard flooding and MPR broadcast.	132
A.1	Neighborhood structure used in SA and TS implementation.	139
A.2	Simulated Annealing algorithm	140
A.3	Tabu Search algorithm	140

List of Tables

1.1	A taxonomy of video-centric cross-layer problems	7
3.1	Summary of notation for Chapter 3	28
3.2	Average distortion values found by the algorithms	48
4.1	Additional notation for Chapter 4	62
4.2	Performance of the proposed algorithm for various instances of Problem OPT-PSRA	79
5.1	Summary of notation for Chapter 5	99
5.2	Performance of the proposed algorithm ($\epsilon = 0.01$)	114
5.3	Average distortion values for different network sizes ($\epsilon = 0.01$)	120
5.4	Average PSNR values(dB) for different network sizes ($\epsilon = 0.01$)	120

Chapter 1

Introduction

1.1 Motivation

As progress in the area of wireless ad hoc networks continues, there is increasing expectation that such networks will support content-rich multimedia communications (e.g., video) in addition to simple data communications. The demands for multimedia applications are indeed compelling. Real-time multimedia information, for example, with live video scenes about field situations, is far more accurate and substantive than that available from data communication. However, at present, there are significant technical barriers that prevent the widespread deployment of multimedia applications over multihop wireless networks, particularly under *extreme environments*, which are characterized by frequent node/link failures, jamming, and dynamic channel conditions.

A wireless link is usually characterized by a high transmission error rate because of physical layer constraints such as shadowing, fading, path loss, and interference from other transmitting nodes. Also, wireless nodes are limited by their maximum transmission power, which reduces their ability to communicate directly with nodes that are beyond their maximum transmission range. Increasing the transmission power will increase transmission range.

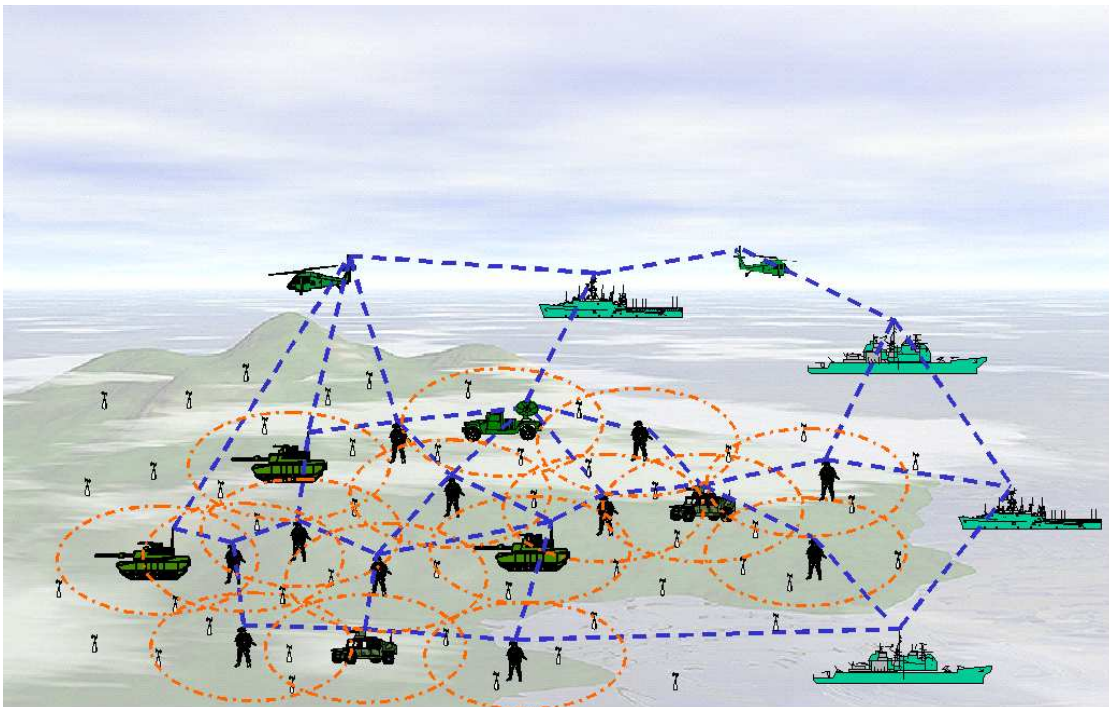


Figure 1.1: A schematic illustrating battlefield wireless ad hoc network.

But it will also create more interference to other nodes. As a result, it is more practical to have intermediate nodes cooperate with each other and route messages to their final destinations (so called multihop wireless networking). Though the idea of having nodes in the network participate in packet forwarding via multihop seems simple, a dynamic topology and potentially large search space makes it a non-trivial problem (in terms of finding and maintaining a good path in multihop wireless networks). This problem is further complicated in networks that are deployed in highly dynamic (unpredictable) or even hostile environments. Figure 1.1 shows an example of such a network in a battlefield setting.

On the other hand, video communications are usually characterized by stringent requirements with respect to packet loss, delay, and application-level performance requirements (e.g., user-perceived video quality). Although many video applications can tolerate certain degree of transmission loss, a burst of lost packets may cause loss of synchronization and error propagation that are beyond error control and error resilient mechanisms, resulting

in a severely degraded video quality. For real-time video, traditional error control mechanisms such as retransmission of lost packets may not be effective due to delay constraints. Consequently, there is a critical need to develop new methodologies to support video communications over multihop wireless networks.

Much of the early research in the area of multihop wireless networks has focused on developing routing protocols for network connectivity and data communications e.g., Ad hoc On-demand Distance Vector (AODV) routing [84], Optimized Link State Routing (OLSR) [23], Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [79], Dynamic Source Routing (DSR) protocol [52]. The success of these protocols resulted in the publication of several Internet Engineering Task Force (IETF) Request For Comments (RFCs) on wireless ad hoc routing. In addition, DSR, Dynamic MANET On-demand (DYMO) Routing, and Simplified Multicast Forwarding for MANET are currently published as Internet-drafts, and are being considered for standardization by the IETF MANET working group. There have also been active research efforts in addressing QoS issues at various layers (MAC, network, transport) [18, 64, 83, 101, 104]. For example, a conventional approach to providing QoS for multimedia applications is to devise routing protocols that optimize network layer metrics such as number of hops, link failure probabilities, and link bandwidths. Indeed, many such efficient protocols have been proposed before (e.g., [9, 18, 62, 83]) for routing in ad hoc networks. These efforts mainly focus on the optimization of one or more network layer metrics, such as throughput, delay, loss, or path correlation etc., and are typically agnostic to application layer performance. As a result, they do not suffice for networks that are required to support video, which is judged by the perceived quality at the receiver. Moreover, few of them address performance issues at the application layer *directly* (e.g., video distortion and user-perceived video quality).

In this dissertation, we present our efforts in developing enabling technologies to support video communications over dynamic ad hoc networks. Our focus is on dynamic ad hoc networks that are operating under extreme environments, where traditional techniques are unable to effectively support video communications. We offer solutions for some important

problems that are expected to arise in practice. The ideas and methodologies proposed in this dissertation would serve as a major milestone in addressing video communications in multihop wireless networks.

1.2 Cross-layer Design for Video Applications

Traditional layering approach inherent to most network designs does not provide a mechanism for network layer to adapt to specific application requirements. Meeting the end-to-end performance requirements of demanding applications such as low-latency video streaming is extremely challenging without interaction across multiple layers. Cross-layer design provides information sharing across different layers, and allows for efficient utilization of network resources, while optimizing the performance of various applications. This is especially true in the case of dynamic ad hoc networks, which, among others, are not only limited by the available bandwidth, available energy and transmission power, but are also vulnerable to frequent node/link failure, jamming and other hostile actions. In this section, we review specific applications of the cross-layer design principle that are relevant to our research effort, and lay the foundation for developing effective routing methodologies for video communications over such networks.

In [20,21], Chou and Miao proposed a rate-distortion optimized streaming framework that considered scheduling packetized media over a packet erasure channel in order to minimize an additive combination of distortion and average rate. They developed an optimization framework that was later applied to a variety of packet network models. Also, they developed a heuristic algorithm for finding the scheduling policy, which is sub-optimal. It is important to note that while an application layer parameter (video distortion) is optimized in this framework by computing the optimal policy for packet scheduling and transmission, this framework does not address issues associated with video packet routing and forwarding.

The idea of streaming layered video (live and stored) over a lossy packet network in order

to maximize the video quality at the receiver was proposed by de Cuetos and Ross in [26]. They used a framework called joint scheduling and error concealment (Joint S+EC), in which packet scheduling decisions at the sender explicitly account for the error concealment mechanism at the receiver. They formulated the problem of finding the optimal scheduling policy as a linear programming problem, which computes the optimal packet scheduling policy that minimized average distortion subject to rate constraints. The authors demonstrated their framework and solution based on Markov Decision Processes (MDP) using MPEG-4 FGS video traces. While this work combines both packet scheduling as well as error concealment, it still remains an edge-based approach which does not address the network layer explicitly.

Chakareski and Girod [15] proposed the idea of rate-distortion optimized packet scheduling for streaming video over single and multiple paths. The authors investigated a sender-driven transmission scenario where diversity is achieved by using multiple transmission paths over the network. In their proposed framework, the sender could decide at every instant which packets, if any, to transmit and over which transmission paths in order to meet a rate constraint while minimizing the end-to-end distortion. However, such paths over which video packets could be sent, are assumed to be computed *a priori*.

In the context of wireless networks, Setton *et al.* [91] explored a cross-layer framework for real-time video streaming over a lossy channel. It incorporated adaptation across most layers of the protocol stack: application, transport protocols, resource allocation at network layer and link layer techniques. In this framework, each layer is characterized by some key parameters, which are passed to the adjacent layers to help determine the operation modes that best suit the current channel, network, and application conditions. Adaptive techniques are used at the link layer to maximize the link rates under varying channel conditions while the MAC layer operates jointly with the network layer to determine the set of network flows that minimize congestion. The application layer determines the most efficient encoding rate that maximizes video quality. However, it has been aptly pointed out by Kawadia and Kumar [53] that in a wireless setting, adaptation at one layer may adversely effect other layers. If not coordinated wisely, such single-layer adaptations may degrade the overall

system performance.

A common theme in these prior works is that the video packets are assumed to be transported across a single lossy channel. But the issues related to optimal routing of these video sessions in a multihop fashion are not specifically addressed. In this dissertation, we illustrate how to optimize the performance of video applications through a cross-layer design principle by factoring in video quality (at the application layer) into routing algorithms at the network layer.

1.3 Problem Scope and Research Contributions

In this dissertation, we focus on addressing important challenges and overcoming technical barriers that hinder the future deployment of video applications over dynamic ad hoc networks. Our research addresses problems that lie at the core of the video-centric cross-layer routing. The theoretical and algorithmic aspects of this work include novel cross-layer formulation, development of effective solution approaches to the ensuing complex optimization problems, exploration of the performance limits and trade-offs, and implementation considerations for practical deployment. Specifically, we address three important problems : concurrent routing, path selection and rate allocation, and multipath routing for multiple description video. Each piece by itself contributes to the common theme, but when they are combined, they can provide enhanced video delivery over multihop wireless networks. Table 1.1 provides a taxonomy of video-centric cross-layer problems discussed in Section 1.2 and places the problems addressed in this dissertation in the context of these prior works.

1.3.1 Concurrent Routing

Existing routing policies for transporting video over ad hoc networks, typically only consider single source-destination pair. On the other hand, it is important to consider the case that

Table 1.1: A taxonomy of video-centric cross-layer problems

Problem Description	Wireline / Wireless	Sessions	Paths per Session	Video Codec	Solution Approach
[20,21] Optimal Packet Scheduling for video streaming	Wireline	Single	Single	LC	H
[15] Optimal Packet Scheduling over multiple paths	Wireline	Single	Multiple	LC	H
[26] Joint Scheduling and error concealment	Wireline	Single	Single	LC	Opt.
[5] MD video with Path Diversity	Wireline	Single	Multiple	MD	-
[91] Cross-layer framework for video streaming	Wireless	Single	Single	LC	-
[68–70] Concurrent Routing	Wireless	Multiple	Single	LC	H (GA)
[58,60] Path Selection and Rate Allocation	Wireless	Multiple	Multiple	LC	Opt. (RLT, bb)
[59,61] Multipath Routing for MD video	Wireless	Single	Multiple	MD	Opt. (RLT, bb)

Note: LC - Layered Coding, MD - Multiple Description Coding, H - Heuristic,
Opt. - Optimal solution, bb - Branch-and-bound.

an ad hoc network must sustain multiple video sessions concurrently. In this case, network resources are shared by these concurrent sessions, and the routing decisions for all these sessions in the network are highly dependent. Moreover, video distortion is a highly complex function of *multiple* network layer metrics [102], and optimizing network layer metrics does not necessarily guarantee optimal video quality. Hence, any mechanism that aims to improve the performance of individual video sessions cannot optimize a global performance objective. By jointly consider the routing of concurrent sessions (henceforth termed as *concurrent routing*), we can optimize the network resource allocation among all the flows and achieve a global network-level performance optimization objective.

In this work [68–70], we formulate a cross-layer optimization problem by considering the application layer performance metric i.e., average video distortion as a function of network layer behavior. This formulation seamlessly unifies video distortion with packet loss due to node/link failure, and packet delay due to congestion, while jointly considering the routing for concurrent video sessions in a multihop wireless network. The problem formulation exhibits a highly complex objective function and constraint set, which renders this problem substantially more difficult to solve than traditional QoS routing problems.

We develop a competitive solution methodology based on Genetic Algorithms (GA). We show that the GA-based algorithms are well suited to solve such complex network-wide optimization problems. Specifically, we show that due to its intrinsic capability to handle a population of solutions, this GA-based approach can efficiently find a set of paths (one for each source-destination pair), that minimize the overall video distortion across all sessions. The proposed GA-based approach is a major step forward in developing a comprehensive methodology for addressing complex cross-layer optimal routing.

1.3.2 Path Selection and Rate Allocation

Any mesh topology, wireless ad hoc networks being no exception, can often be characterized by a multitude of paths between a given source and destination. Thus, a mechanism that

takes advantage of these paths is bound to perform better (i.e., in supporting QoS for real-time traffic) than traditional single path approaches. Among such mechanisms, path selection is very important for supporting video sessions over multiple paths, because the quality of the received video is highly dependent on the “quality” of the paths in terms of loss, delay and delay variations. Coupled with the path selection strategy is computing the optimal video encoding rate, as well as the allocation of this video rate across the selected paths, in a way that maximizes the video quality at the receiver. Therefore, an efficient *path selection and rate allocation* algorithm that not only makes route decisions but also proportions traffic over these paths, based on application layer performance metrics such as distortion, can significantly improve the quality of the reconstructed video.

In the presence of multiple paths between a source-destination pair, this work [58, 60] provides a guideline on choosing a set of optimal paths and rate vectors that would effectively deliver the best video quality at the receiver. The formulation not only captures the rate allocation for each video session, but also splits the optimal rate over a set of paths such that the overall quality of the reconstructed video is optimized. Such a multipath transport approach has many advantages, including bandwidth aggregation, load balancing and improved error resilience against transmission errors and link failures.

We formulate the problem as a nonlinear optimization problem, for which we derive a tight linear programming relaxation by employing a novel relaxation technique called *Reformulation-Linearization Technique* (RLT). This relaxation is embedded within a specialized branch-and-bound procedure, and the proposed method is known to converge to a global optimum. The specific partitioning strategy that is employed in the context of this branch-and-bound framework, while preserving the convergence property, is presented. Computational results are reported and comparisons with max-min rate allocation scheme are presented. Results indicate that our proposed algorithm yields significantly improved solutions when compared to these other methods.

1.3.3 Multipath Routing for Multiple Description Video

For ad hoc networks operating under extreme conditions, wireless links within such a network are highly fragile, and it is highly likely that any particular path within the network will not remain reliable over an extended period of time. In this environment, traditional layered video coding or single stream coding might not perform well because either scheme requires at least one relatively reliable path from the sender to the receiver.

Recently, a new video coding technique called multiple description (MD) coding has been developed, which is capable of encoding a video source into multiple independent streams (or descriptions) such that any subset of these streams at the receiver can be used to reconstruct the original video. The quality of the reconstructed video is commensurate to the information received from the number of video descriptions. This new video coding technique is drastically different from layered scalable video coding, within which the successful reconstruction of the video is highly dependent upon the most significant layer i.e., the base layer.

Multipath transport i.e., using multiple paths for a multimedia session, can take advantage of video coding schemes such as MD coded video that distribute the video information among multiple paths, possibly with some correlation between the information on the various paths, so as to protect against failure of some subset of the paths. In the worst case, information received from any path should be sufficient to reconstruct video with acceptable quality. *Multiple description video* is such a video coding technique that when used in combination with *multipath routing* has a great potential for multimedia applications in wireless ad hoc networks.

In this work [59, 61] we show that MD video coding when used in combination with multipath transport, can mitigate the problems associated with the use of single stream approaches discussed above. Further, we characterize and model a video application centric routing problem by taking a cross-layer approach that combines application layer performance and network layer multipath routing. The formulation provides a framework for

performing multipath routing such that application layer video quality is optimized. We also provide a theoretical underpinning to achieve optimal performance for MD video over multipath. We show how to achieve the theoretical optimality, by deriving tight linear programming relaxations that serve to lay the foundation for designing a specialized branch-and-bound algorithm. RLT is employed to derive the linear programming relaxation to the underlying 0-1 mixed integer nonconvex programming problem.

1.4 Outline

This chapter presents the general background for the video communications in multihop wireless networks and provides a preview of the main contributions in this dissertation. The rest of the dissertation is organized as follows. Chapter 2 provides a brief review of relevant mathematical background used in this dissertation for solving nonconvex optimization problems. In Chapter 3, we study the problem of routing for multiple concurrent video sessions in wireless ad hoc networks. We describe a solution procedure for this problem based on GA, and validate it based on extensive simulations. Chapter 4 addresses the problem of path selection and rate allocation for concurrent video sessions sustained in ad hoc networks. We design a specialized variant of the RLT-based branch-and-bound framework to solve this problem. Chapter 5 considers the problem of multipath routing for multiple description video over wireless ad hoc networks. Linear programming relaxations are derived using RLT, and the resulting construct is embedded in a specialized branch-and-bound algorithm, in order to solve this problem. In Chapter 6, we discuss practical considerations and wireless node architecture for a distributed implementation of the proposed routing algorithms. Finally, Chapter 7 provides the summary and future research direction.

Chapter 2

Mathematical Background

2.1 Global Optimization

Ever since complexity theory [35] was developed in the 1970s, it was understood that non-convex global optimization problems were NP-hard in general. So, it would be an arduous task to develop efficient solution procedures to determine the global optimum for such problems. Much of the early research was aimed at developing problem-specific as well as general heuristic methodologies that would generate good quality feasible solutions, while minimizing the possibility of being trapped at a local optimum. Metaheuristic approaches such as Genetic Algorithms (GAs) fall into this category. However, recent advances in computing technology have spurred research into developing algorithms for solving difficult nonconvex programming problems to optimality.

In this chapter we discuss two types of global optimization techniques. We begin this chapter with a brief review of genetic algorithms. Following this, we review an efficient branch-and-bound based global optimization algorithm in Section 2.3.

2.2 Genetic Algorithms

Global optimization problems in general, and specifically the problems that are explored in this dissertation, are very complex in nature and quite hard to solve by conventional optimization techniques. Genetic algorithms were introduced in the mid 1970s and have since received considerable attention in solving networking problems. Genetic algorithms not only have the potential of being an efficient optimization technique, but also have the ability to provide a *population* of feasible solutions that the user can choose from. Genetic algorithms, proposed by John Holland in 1975 [46], are probabilistic optimization algorithms based on the natural genetic processes of living organisms. These algorithms are inspired by the *survival-of-the-fittest* principle [42, 46] and simulate natural evolution through generations. They can often outperform conventional optimization methods when applied to difficult real world design optimization problems. Their intrinsic strength in dealing with a set of solutions (i.e., a population) at each generation, rather than working with a single current solution, makes genetic algorithms more desirable than other trajectory based metaheuristics. In general, population-based metaheuristics (for which GA is an example) can provide a natural means of exploration of the search space in multiple dimensions. At each iteration, genetic operators such as *selection*, *crossover*, and *mutation* are probabilistically applied to the individuals of the current population in order to generate individuals for the next generation. In general, individuals having a higher degree of fitness will have a higher probability to be chosen as members of the population for the next iteration.

Using the terminology of genetic algorithms, solutions or *phenotypes* are encoded by using individuals or *genotypes*. The fundamental idea is that genetic algorithms operate on a finite population of “chromosomes” (solutions), which are either fixed or variable size strings, with binary or integer values (“genes”) at each position (or “locus”). Each chromosome of the population is evaluated according to some fitness function. Members of the population are selectively interbred in pairs to produce offspring. Genetic operators are used to facilitate a breeding process that results in the offspring inheriting properties from their parents.

Therefore, each iteration facilitates both *self-adaptation* and *co-operation*. Self-adaptation refers to the individuals evolving independently, as enabled by the mutation operator. On the other hand, co-operation implies the sharing of the best traits among the individuals, in the form of the crossover operator. The offspring are evaluated (using a fitness function) and placed in the population, possibly replacing the weaker members of the previous generation. The search mechanism, therefore, consists of three phases: evaluation of the fitness of each chromosome in the population, selection of the parent chromosomes, and applications of mutation and crossover operators to the parent chromosomes. The new chromosomes resulting from these operations form the population for the next generation; the process is repeated until the system reaches an equilibrium (i.e., ceases to improve). The *survival-of-the-fittest* principle ensures that the overall quality of solutions improves as the algorithm progresses from one generation to the next.

The potential of GA in addressing networking problems has been recognized in recent years. GA can be highly effective when used for delay-sensitive real-time applications, because of its intrinsic ability to provide a trade-off between finding solutions that are closer to the optimal, and computational complexity. One such scenario is making routing decisions for video streaming applications. In such applications, GA can compute a set of “good” routes very quickly, which can be used by the application right away, while GA continues to evolve and compute better solutions. Eventually, the routes used by the said application can be updated with the best set of routes for optimal performance.

Figure 2.1 provides a visual representation of each of the basic building blocks of genetic algorithms, and Figure 2.2 depicts the flow-chart for GA-based solution approach. Note that both crossover and mutation are performed with certain probabilities (θ and μ , respectively) on the individual solutions. The termination condition in Figure 2.2 could be based on the total number of iterations (generations), the maximum computing time, or a threshold of desired video distortion. We conclude this section by examining each of the genetic operators in further detail. Interested readers may refer to the books authored by Goldberg [42], Gen and Cheng [36] and Back *et al.* [7] for a general discussion on genetic algorithms and

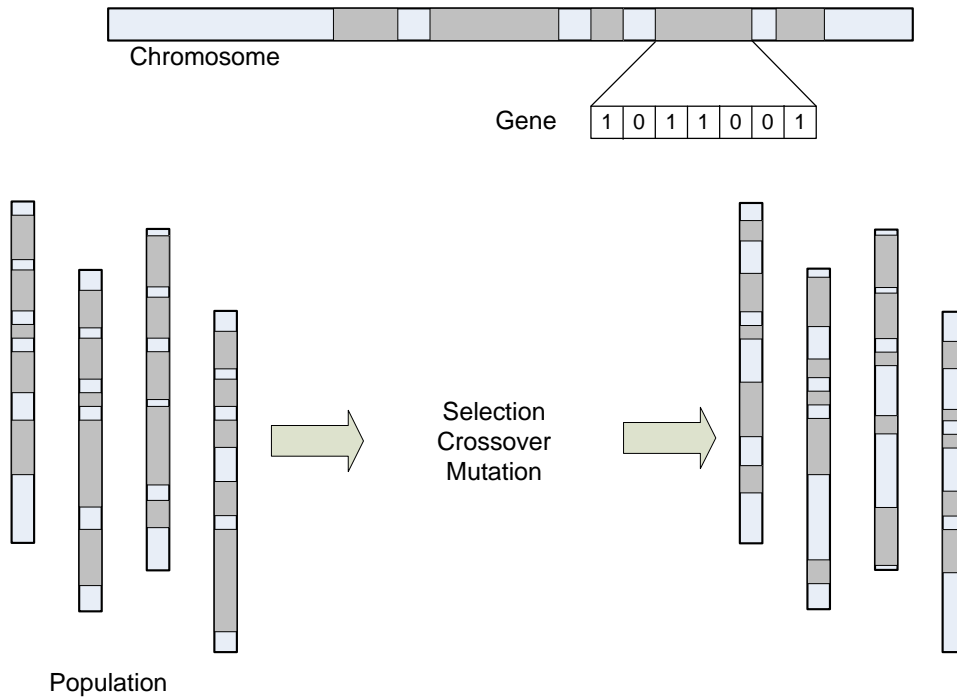


Figure 2.1: A schematic representation of genetic algorithms.

combinatorial optimization.

2.2.1 Selection

During this operation, we select high-quality individuals that have a better chance or potential to produce “good” offspring in terms of their fitness values. The selection operator is intended to improve the average quality of the population. By virtue of the selection operation, “good” genes among the population are more likely to be passed to the future generations. The selection operator thereby focuses on the exploration of promising regions in the solution space. Any selection scheme is characterized by a parameter called selection pressure. It is defined as the ratio of the probability of selection of the best chromosome in the population to that of an average chromosome. This means that having a high selection pressure enables the population to reach equilibrium very quickly, but in doing so, it inevitably sacrifices genetic diversity (i.e., results in convergence to a suboptimal solution).

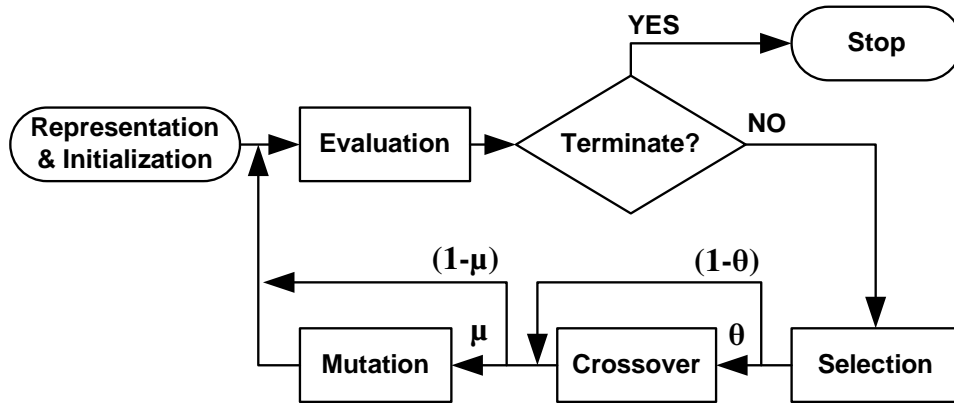


Figure 2.2: Flow-chart for the GA-based solution procedure.

Several selection schemes can be employed during the selection operation. For example, one possible scheme (known as *Roulette wheel* selection [7]) is to select an individual based on a probability in proportion to its normalized fitness value, i.e., $Pr\{\text{choosing individual } i\} = f(x_i) / \sum_j f(x_j)$, where $f(x_i)$ is the fitness of individual x_i . Another possible scheme (known as *Tournament* selection [7]) randomly chooses m individuals from the population each time, and then selects the best of these m individuals in terms of their fitness values. A third way is to use a *Ranking* selection method in which the population is sorted based on ranks given to each individuals, and the probability of selecting an individual is based on the given rank but not its fitness. Two ranking methods used are *linear* and *exponential* ranking. By repeating one of these selection procedures multiple times, a new population can be selected.

2.2.2 Crossover

Crossover is one of the most important genetic operators. It operates on two chromosomes at a time and generates offspring by combining some features of both chromosomes, as shown in Figure 2.3. The crossover between two dominant parents chosen by the selection operator provides higher probability of producing offspring having dominant traits. In its simplest form, the *single point* crossover chooses a random cutoff point in each of the two strings

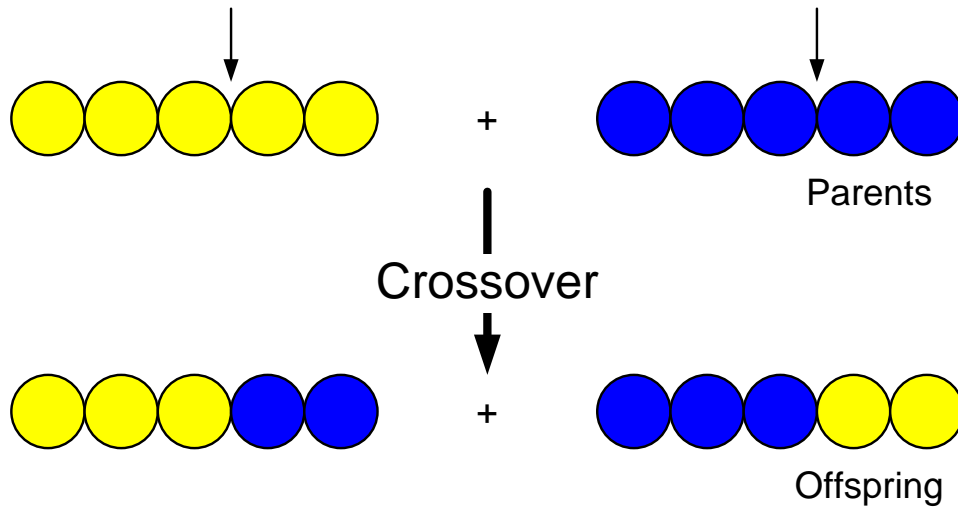


Figure 2.3: A schematic representation of the crossover operator.

(representing the parents) to form two substrings one to the left of the point, and one to the right. Next the left part of the string of one parent is spliced with the right part of the string of the other parent and vice versa, in order to generate two offspring. We can easily imagine that instead of one point crossover, we may have two, three or more point crossover. Two or more point crossover is implemented by choosing two or more random points in the selected pair of strings and exchanging the sub-strings defined by the chosen points. Some studies concluded that multipoint crossover decreases the effectiveness of the genetic algorithm and this decrease is greater if more points are considered [27]. However in some cases it may provide better results [108].

In the context of routing problems, crossover plays the role of physically exchanging each partial route of two chosen chromosomes in such a manner that the offspring produced by the crossover represents only one route. One partial route connects the source node to an intermediate node, and the other partial route connects the intermediate node to the destination node.

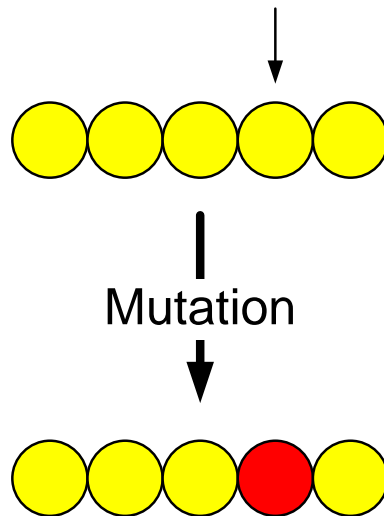


Figure 2.4: A schematic representation of the mutation operator.

2.2.3 Mutation

Mutation is an operator that produces spontaneous random changes in various chromosomes and thus it introduces some extra variability into the population in order to avoid local optima, as shown in Figure 2.4. In doing so, mutation may induce a subtle bias in the population. In the context of routing problems, mutation generates an alternative partial-route from the mutated node to the destination node. A topological information database might have to be utilized by the mutation function for this purpose.

In Chapter 3, we describes a GA-based solution procedure for solving a nonconvex optimization problem for video communications over ad hoc networks.

2.3 RLT-based Global Optimization for Solving Non-convex Programming Problems

In the previous section, we discussed how a popular metaheuristic like genetic algorithms can be used as a viable global optimization technique. However, as pointed out earlier, while

GA can produce good quality set of feasible solutions quickly, the optimal solution remains unknown. Motivated by this fact, along with a rapid growth in computation power in recent years, a significant amount of research has been dedicated to developing exact algorithms that can solve nonconvex programming problems to optimality. In this section we present a global optimization algorithm based on a novel relaxation technique called *Reformulation-Linearization Technique* (RLT) of Sherali and Tuncbilek [97], which can efficiently solve this class of nonconvex polynomial problems to global optimality. The key to obtaining global optimal solution is to embed the RLT construct in a branch-and-bound framework as described below. In the rest of this chapter, we discuss the concepts of branch-and-bound and RLT, which form the foundations of the optimization algorithm.

2.3.1 The Reformulation-Linearization Technique

A polynomial programming problem seeks to minimize a polynomial objective function subject to a set of polynomial constraint functions, where all the functions are defined in terms of some bounded, continuous decision variables. A polynomial programming problem can be mathematically stated as follows.

$$\begin{aligned}
 \mathbf{PP}(\Omega): \quad & \mathbf{Minimize} \quad \{\phi_0(x) : x \in Z \cap \Omega\} \\
 \text{where:} \quad & Z = \{x : \phi_r(x) \geq \beta_r, r = 1, \dots, R_1, \phi_r(x) = \beta_r, r = R_1 + 1, \dots, R\} \\
 & \Omega = \{x : 0 \leq l_j \leq x_j \leq u_j < \infty, j = 1, \dots, n\} \\
 \text{and where:} \quad & \phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt} \left[\prod_{j \in J_{rt}} x_j \right], r = 0, 1, \dots, R. \tag{2.1}
 \end{aligned}$$

Here T_r is an index set for the terms defining $\phi_r(\cdot)$, and α_{rt} are the real coefficients for the polynomial terms $(\prod_{j \in J_{rt}} x_j)$, $t \in T_r$, $r = 0, 1, \dots, R$. Note that a repetition of indices is permitted within each multi-set J_{rt} . For example, if $J_{rt} = \{1, 2, 3, 3\}$, then the corresponding polynomial term is $x_1 x_2 x_3^2$. Denote $N = \{1, \dots, n\}$, and define $\bar{N} = \{N, \dots, N\}$

to be composed of δ replicates of N , where δ is maximum degree of any polynomial term appearing in $PP(\Omega)$. Then each $J_{rt} \subseteq \bar{N}$, with $1 \leq |J_{rt}| \leq \delta$, for $t \in T_r$, $r = 0, 1, \dots, R$.

Determining a global optimum to a polynomial program, as defined in (2.1), is a computationally difficult task (theoretically, this is NP-Hard), and thus requires the use of specialized algorithms. Several solution approaches having been developed for such polynomial programming problems with varying degrees of success. Chang and Chang [17] proposed a concise method to solve the 0-1 mixed-integer polynomial programming problem using additional 0-1 variables and auxiliary constraints. Sherali and Adams [93,95] developed a hierarchy of representations and related relaxations for 0-1 pure and mixed-integer polynomial programs, leading to their convex hull representation. Of particular interest is the global optimization algorithm developed by Sherali and Tuncbilek [94,97] based on *Reformulation-Linearization Technique* (RLT), which is capable of solving general polynomial programs to optimality, as explained below.

RLT is a relaxation technique that can be used to produce tight *polyhedral outer approximations* or *linear programming relaxations* for an underlying nonlinear, nonconvex polynomial programming problem. In essence, RLT can provide a tight lower bound on a minimization problem [94,97]. In the RLT procedure, nonlinear implied constraints are generated by taking the products of bounding terms of the decision variables, up to a suitable order and also, possibly products of other defining constraints of the problem. The resulting problem is subsequently linearized by variable substitutions, one for each nonlinear term appearing in the problem, including both the objective function and the constraints. The mechanics of RLT automatically creates outer linearizations that approximate the closure of the convex hull of the feasible region Ω . For solving a general polynomial program the RLT-based approach operates as follows. Given a solution space Ω , in order to construct the bounded linear programming problem $LP(\Omega)$ using RLT, implied bound-factor product constraints are generated by using *distinct* products of the bounding factors $(x_j - l_j) \geq 0$ and $(u_j - x_j) \geq 0$, $j \in N$, taken δ at a time. These product constraints can be expressed as follows.

$$F_\delta(J_1, J_2) \equiv \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \geq 0, \quad (2.2)$$

where $J_1 \cup J_2 \subseteq \bar{N}$, and $|J_1 \cup J_2| = \delta$.

The RLT process proceeds as follows. In the reformulation phase, constraints (2.2) are included in the problem $PP(\Omega)$. In the linearization phase, the substitution,

$$X_J = \prod_{j \in J} (x_j), \quad \forall J \subseteq \bar{N} \quad (2.3)$$

is applied to linearize the resulting problem, where the indices in J are assumed to be sequenced in non-decreasing order, and where $X_{\{j\}} \equiv x_j$, $\forall j \in N$, and $X_\emptyset \equiv 1$.

To solve $PP(\Omega)$ to global optimality, $LP(\Omega)$ is embedded in a branch-and-bound algorithm to compute lower bounds on the underlying polynomial program. This procedure proposed by Sherali and Tuncbilek [97,98] essentially involves the partitioning of the original set Ω into sub-hyperrectangles, each of which is associated with a node of the branch-and-bound tree. A partitioning rule, geared towards identifying the variable that contributes the most to the discrepancy between a new RLT variable that contains it and the associated corresponding nonlinear product that this RLT variable represents, is followed. The motivation is to drive all such discrepancies or *relaxation errors* to zero by creating partitions that would induce the variables to achieve their bounds, leading to a global optimal solution.

2.3.2 Branch-and-Bound

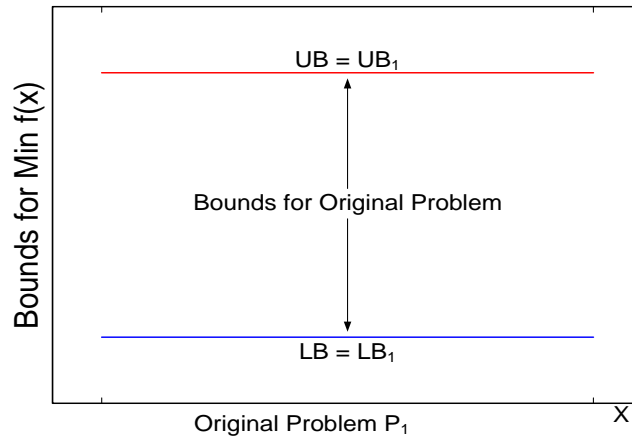
Branch-and-bound [76] is an iterative relaxation algorithm that seeks to produce an optimal solution to a nonlinear programming (NLP) problem by partitioning the original solution space into sub-hyperrectangles [94]. In branch-and-bound, the original problem $PP(\Omega)$, is first relaxed using a suitable *relaxation technique* (such as RLT) to obtain an easier-to-solve, lower-bounding problem $LP(\Omega)$. The optimal solution to this LP relaxation, $LP(\Omega)$, provides

a lower bound LB for the original problem. Since $LP(\Omega)$ usually yields an infeasible solution to $PP(\Omega)$, a *local search algorithm* should be employed to find a feasible solution, using the infeasible lower bounding solution as a starting point. The resulting feasible solution then provides an upper bound UB for the original problem $PP(\Omega)$ (see Figure 2.5 for an example).

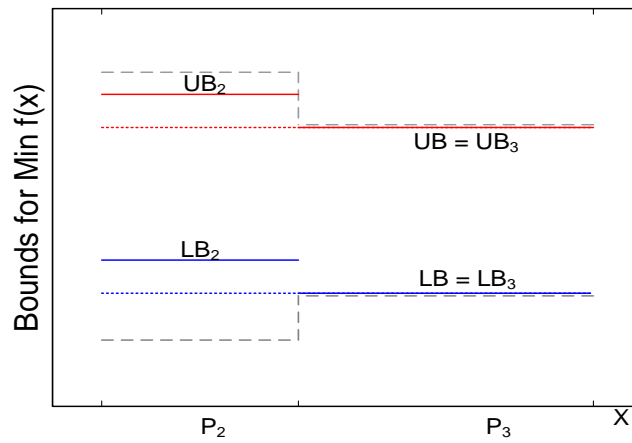
The branch-and-bound procedure is based on the idea of *divide-and-conquer*. That is, the original problem $PP(\Omega)$ is partitioned into sub-problems, each having a smaller feasible solution space, based on the solution provided by $LP(\Omega)$. New sub-problems are organized as a *branch-and-bound tree*, while this partitioning or branching process is carried out recursively to obtain two new sub-problems at each node of the tree. The sub-problems are also inserted into a *problem list* L , which records the active nodes in the branch-and-bound tree structure. More specifically, in the beginning, the problem list L is initialized with the original problem $PP(\Omega)$ (or P_1 in Figure 2.5(a)). At any given iteration, the lower and upper bounds for $PP(\Omega)$ are computed as

$$\begin{cases} LB = \min\{LB_k : \text{Problem } k \in L\} \\ UB = \min\{UB_k : \text{all explored nodes } k \text{ thus far}\}. \end{cases} \quad (2.4)$$

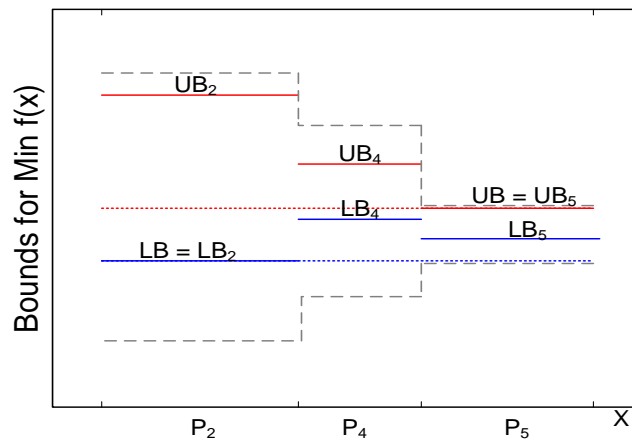
The method proceeds by choosing the next problem to partition from the problem list. In our approach, the problem $k \in L$ having the smallest LB_k is chosen. This problem k is then partitioned into two sub-problems k_1 and k_2 , which replace problem k in L . Every time a problem k is added to the list, LB_k and UB_k are computed, and the LB and UB for the original problem $PP(\Omega)$ are updated. For example, in Figure 2.5(b), the original problem (denote as problem P_1) is divided into two new problems (denote as problem P_2 and problem P_3). Since the relaxations in problems P_2 and P_3 are both tighter than that in problem P_1 , we have $UB_2, UB_3 \leq UB_1$ and $LB_2, LB_3 \geq LB_1$. The upper bound of the original problem is updated from $UB = UB_1$ to $UB = \min\{UB_2, UB_3\}$ and the lower bound of the original problem is updated from $LB = LB_1$ to $LB = \min\{LB_2, LB_3\}$. As a result, we now have smaller gap between UB and LB . At any given iteration, if $LB \geq (1 - \epsilon) \cdot UB$, the procedure terminates and we have an ϵ -optimal solution, where $\epsilon \geq 0$ is some selected



(a)



(b)



(c)

Figure 2.5: Illustration of branch-and-bound.

optimality tolerance ($\epsilon \equiv 0$ if an exact optimum is desired). Otherwise, we choose a problem with the minimum lower bound (Problem P_3 in Figure 2.5(b)) and partition this problem into two new sub-problems (P_4 and P_5 in Figure 2.5(c)). Also, for any problem k in the problem list, if $LB_k \geq (1 - \epsilon) \cdot UB$, no globally optimal solution that improves beyond the ϵ -tolerance can exist in the subspace of the feasible region represented by this node. Therefore, this node can be removed (or *fathomed*) from the branch-and-bound tree. In this manner, the branch-and-bound process can fathom certain branches or nodes of the tree, eliminating them from further exploration. The effectiveness of the branch-and-bound procedure depends strongly on that of the employed fathom strategy.

While the solution framework discussed thus far is well suited for solving polynomial programming problems, it is worth mentioning that the specific problems explored in this dissertation fall into the category of *0-1 mixed integer nonconvex, non-polynomial* programming problems. In order to use RLT in this situation, we will need to insert an intermediate level of approximation that relaxes the original problem into a polynomial programming problem, thereby facilitating the application of the above mentioned solution procedure.

Chapter 3

Routing for Concurrent Video Sessions in Ad Hoc Networks

3.1 Introduction

Enabling video service in wireless ad hoc networks [38, 41, 71] has been an area of intense research in recent times. An important milestone was a successful video demonstration over ad hoc network testbeds in [48]. Most of the prior efforts mainly focus on end system-based techniques, but do not consider optimal routing as a means to improve the quality of the video sessions. On the other hand, many efficient protocols have also been proposed for quality of service (QoS) routing in ad hoc networks, e.g., [9, 18, 62]. These efforts mainly focus on the optimization of one or more network layer metrics, such as throughput, delay, loss, or path correlation, and we therefore term them *network centric* routing throughout this chapter. Although these protocols are shown to be quite effective for data communications, they may not be optimal for video communication due to the fact that video distortion is a highly complex function of multiple network layer metrics [102]. Optimizing network layer metrics does not necessarily guarantee optimal video quality. Furthermore, an important consideration that is often ignored is the fact that multiple video sessions can be sustained

by an ad hoc network concurrently. Multiple video sessions compete among each other for the limited network resources, and such interactions couple the performance of an individual flow with that of other flows. Therefore, by jointly considering the routing of concurrent video sessions, we can optimize the network resource allocation and maximize performance objective across all flows.

In this chapter, we consider the problem of supporting multiple concurrent video sessions in ad hoc networks. We formulate the optimization problem from a cross-layer perspective by considering the application layer performance metric (i.e., average video distortion) as a function of network layer behavior (routing of each session). This formulation unifies video distortion with packet loss (due to node/link failures) and delay (due to congestion), while jointly considering routing for concurrent sessions. The problem formulation exhibits a highly complex objective function and constraints, which renders this problem substantially more difficult than traditional QoS routing problems.

We then proceed to develop a highly competitive solution method based on genetic algorithms (GAs) [7, 36, 42]. GA-based algorithms have an intrinsic capability to handle a population of solutions, which perfectly suits the nature of our cross-layer concurrent routing problem. GA-based approaches have the unique strength of identifying promising search regions and have less of a tendency to be trapped at a local optimum, as compared with other single-solution based trajectory methods [7]. We find that the complex network-wide optimization problem provides the perfect setting for a GA-based solution method. The complexity due to the interaction among concurrent sessions can be handled rather naturally by GAs since they are intrinsically parallel. Multiple session routing and flow interactions increase computational effort only linearly as compared with the single session routing. We demonstrate the performance of the GA-based approach over other network-centric approaches through extensive simulations.

The chapter begins with mathematical modeling of various network layer parameters followed by discussion of the empirical rate-distortion model at the application layer, in

Section 3.2. Based on the models at various layers, the application-centric optimal routing problem for multiple concurrent video sessions is formulated. Continuing further, we present a GA-based solution procedure in Section 3.3 along with a fast greedy heuristic algorithm in Section 3.4, that is used in the initialization of the GA-based solution procedure. Simulation results are presented that show the performance of the GA-based algorithm in Section 3.5. Related work is discussed in Section 3.6 and Section 3.7 summarizes this chapter.

3.2 Problem Formulation

In this section, we formulate the routing problem for multiple concurrent video sessions in a wireless ad hoc network. We assume that a wireless link exists between nodes i and j if nodes i and j can communicate with (i.e., within the radio transmission range of) each other. Consequently, the ad hoc network can be modeled as a time-varying, directed graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of vertices, representing mobile nodes, and \mathcal{L} is the set of wireless links in the network.

In the graph, we characterize each wireless (directed) link $\{i, j\} \in \mathcal{L}$ by the following two parameters: (i) c_{ij} : the available bandwidth of link $\{i, j\}$; and (ii) p_{ij} : the mean packet loss probability of link $\{i, j\}$, due to transmission errors or link failures. We assume that lower layer dynamics, e.g., interference and fading, could be translated into the network layer metrics.

Consider a set of concurrent video sessions, denoted as \mathcal{E} . Each session $\sigma \in \mathcal{E}$ has a source node s_σ and a destination node d_σ . The rate of a video stream, R_σ , is bounded by $\underline{R}_\sigma \leq R_\sigma \leq \overline{R}_\sigma$, $\sigma \in \mathcal{E}$. The lower and upper bounds of R_σ are determined by the specific video coder and the video sequence encoded at the source node s_σ , or the user requirement on received video quality. The decoding deadline for packets associated with session σ is Δ_σ . The optimal routing problem aims to find a set of paths for the video sessions, such that the total distortion of all the video sessions is minimized.

Table 3.1 summarizes the notation used in the chapter.

Table 3.1: Summary of notation for Chapter 3

Symbol	Definition
$\mathcal{G}\{\mathcal{N}, \mathcal{L}\}$	Graph representation of an ad hoc network.
\mathcal{N}	Set of vertices.
\mathcal{L}	Set of edges.
$\{i, j\}$	A wireless link from node i to node j .
c_{ij}	Available bandwidth of link $\{i, j\}$.
p_{ij}	Packet loss probability of link $\{i, j\}$.
λ_{ij}	Average aggregate traffic load on link $\{i, j\}$.
ρ_{ij}	Utilization of link $\{i, j\}$.
\mathcal{E}	Set of video sessions.
$\{s_\sigma, d_\sigma\}$	Source/destination nodes of session σ .
x_{ij}^σ	Index variable defined in (3.12).
\mathcal{P}_σ	Path of session σ , from s_σ to d_σ .
p_σ	End-to-end loss rate of session σ .
R_σ	Rate of video session σ .
$\{\underline{R}_\sigma, \bar{R}_\sigma\}$	The min/max rates of video session σ .
Δ_σ	Decoding deadline of session σ .
t_{ij}	Delay on link $\{i, j\}$.
$f_{ij}(y)$	Probability density function of t_{ij} .
$M_{ij}(s)$	Moment generating function of t_{ij} .
T_σ	End-to-end delay of session σ .
$M_\sigma(s)$	Moment generating function of T_σ .
D_σ^e	End-to-end distortion of session σ .
D_σ^{enc}	Encoding distortion of session σ .
D_σ^{cg}	Session σ distortion due to congestion.
D_σ^{loss}	Session σ distortion due to packet loss.
θ	Crossover rate in the GA-based approach.
μ	Mutation rate in the GA-based approach.

3.2.1 Network Layer Performance Metrics

Load on a Link

Let $\bar{\mathcal{P}}_\sigma^{ij}$ denote the upstream *partial* path of \mathcal{P}_σ from the source node s_σ to the link $\{i, j\}$, exclusive. Note that $\bar{\mathcal{P}}_\sigma^{ij} = \emptyset$ if link $\{i, j\} \notin \mathcal{P}_\sigma$. Then, the average aggregate traffic load on any link $\{i, j\} \in \mathcal{L}$ is:

$$\lambda_{ij} = \sum_{\sigma \in \mathcal{E}} R_\sigma \cdot \prod_{\{m,n\} \in \bar{\mathcal{P}}_\sigma^{ij}} (1 - p_{mn}). \quad (3.1)$$

That is, the average traffic load of link $\{i, j\}$ is the sum of the average rates of the video sessions that pass through this link, decreased by the losses incurred in their upstream links before reaching link $\{i, j\}$. The average capacity utilization of link $\{i, j\}$ is $\rho_{ij} = \lambda_{ij}/c_{ij}$, $\{i, j\} \in \mathcal{L}$. For stability, a feasible set of routes $\{\mathcal{P}_\sigma\}_{\sigma \in \mathcal{E}}$ should satisfy $\rho_{ij} < 1$, $\{i, j\} \in \mathcal{L}$.

Delay on a Link

We model each link $\{i, j\}$ as a general queuing system with an average input rate λ_{ij} (defined in (3.1)) and a service capacity c_{ij} . Let the queueing delay on link $\{i, j\}$ be t_{ij} and its probability density function be $f_{ij}(y)$. We assume that all the moments of t_{ij} are finite, which is true for most queueing systems. For example, when the video traffic is a constant bit rate (CBR) that exhibits short-range dependent (SRD) characteristics, we could model the queueing delay via an exponential distribution, i.e.,

$$f_{ij}(y) = \alpha_{ij} \cdot e^{-\alpha_{ij}y}, \quad \text{for } y \geq 0, \quad (3.2)$$

where $\alpha_{ij} \stackrel{\text{def}}{=} (c_{ij} - \lambda_{ij})$ is the link's residual capacity. On the other hand, for a variable bit rate (VBR) video that exhibits long-range dependent (LRD) characteristics, we could model the link as a fractional Brownian motion (fBm) queueing system, where t_{ij} has a heavy-tailed Weibull distribution [78].

End-to-End Delay

The end-to-end delay of session σ , denoted by T_σ , $\sigma \in \mathcal{E}$, is the sum of the queueing delay on each link along path \mathcal{P}_σ , i.e.,

$$T_\sigma = \sum_{\{i,j\} \in \mathcal{P}_\sigma} t_{ij}, \quad \forall \sigma \in \mathcal{E}. \quad (3.3)$$

We apply the large deviation approximation to obtain an accurate estimate of the overdue probabilities. For the sum of a small number of independent random variables, an estimate based on the *Chernoff Bound* [19] is known to be accurate and computationally efficient [30]. In the following, we illustrate such an approximation when link delays are exponentially distributed.

We first derive the moment generating function of t_{ij} as:

$$M_{ij}(s) = \mathbb{E}[e^{st_{ij}}] = \frac{\alpha_{ij}}{\alpha_{ij} - s}, \quad \text{for } s < \alpha_{ij}. \quad (3.4)$$

Assuming delays on the links are independent, the moment generating function of T_σ is:

$$M_\sigma(s) = \prod_{\{i,j\} \in \mathcal{P}_\sigma} M_{ij}(s), \quad \text{for } s < \min_{\{i,j\} \in \mathcal{P}_\sigma} \{\alpha_{ij}\}. \quad (3.5)$$

Define a function $F_\sigma(s)$ as:

$$F_\sigma(s) = s\Delta_\sigma - \sum_{\{i,j\} \in \mathcal{P}_\sigma} \log M_{ij}(s). \quad (3.6)$$

Since $F_\sigma''(s) < 0$, for $s < \min_{\{i,j\} \in \mathcal{P}_\sigma} \{\alpha_{ij}\}$, $F_\sigma(s)$ is a strictly concave function with a unique maximum at s_σ^* . If $\Delta_\sigma > \mathbb{E}(T_\sigma)$ (i.e., the decoding deadline is larger than the average end-to-end delay on the path, we can determine s_σ^* by solving:

$$F_\sigma'(s) = \Delta_\sigma - \sum_{\{i,j\} \in \mathcal{P}_\sigma} \frac{1}{\alpha_{ij} - s} = 0. \quad (3.7)$$

Note that since $F_\sigma'(\min_{\{i,j\} \in \mathcal{P}_\sigma} \{\alpha_{ij}\}) = -\infty < 0$ and $F_\sigma'(0) = \Delta_\sigma - \mathbb{E}(T_\sigma) > 0$, we have that $0 < s_\sigma^* < \min_{\{i,j\} \in \mathcal{P}_\sigma} \{\alpha_{ij}\}$. From the Chernoff Bound [30], the tail distribution of T_σ can be approximated as:

$$Pr\{T_\sigma \geq \Delta_\sigma\} \approx \frac{\exp\{-F_\sigma(s_\sigma^*)\}}{s_\sigma^* \delta(s_\sigma^*) \sqrt{2\pi}}, \quad (3.8)$$

where $\delta^2(s) = \frac{\partial^2 \log M_\sigma(s)}{\partial s^2}$.

Note that the moment generating function of a heavy-tailed Weibull random variable does not exist (although all of its moments are well-defined). Therefore, the above Chernoff Bound approach cannot be applied to delays having such distributions. However, the overdue probability can be computed by taking advantage of the *sub-exponential* property. For example, we have that $\Pr[\sum_{k=1}^n X_k > x] \approx \Pr[\max_{\{1 \leq k \leq n\}} \{X_k\} > x] \approx n \cdot \Pr[X_1 > x]$ [43], for an i.i.d. sequence of heavy-tailed Weibull random variables $\{X_1, \dots, X_n\}$.

End-to-End Loss Rate

Assuming that the packet loss processes on the links are independent, the end-to-end loss probability of session σ can be computed as:

$$p_\sigma = 1 - \prod_{\{i,j\} \in \mathcal{P}_\sigma} (1 - p_{ij}), \quad \forall \sigma \in \mathcal{E}. \quad (3.9)$$

3.2.2 End-to-End Video Rate-Distortion Model

In [102], Stuhlmüller *et al.* developed an empirical rate-distortion model for a hybrid motion compensated video encoder. They found that theoretically founded rate-distortion models often cannot describe experimental results very accurately due to simplistic assumptions. To avoid these limitations without an increase in model complexity, this particular rate-distortion model uses a simple equation that relates the distortion at the encoder D^{enc} to the relevant parameters. In the simulation scenarios that were considered by the authors, they found two parameters with a significant impact on D^{enc} , namely the source rate R_{enc} that is allocated to the video encoder, and second, the percentage of INTRA coded macroblocks (INTRA rate) β that is enforced by the coding control to improve error robustness. One drawback of this approach as described in [102], is that the necessary model parameters cannot be derived from commonly used signal statistics, like variance, correlation, or the power spectral density. Instead, the parameters need to be estimated by fitting the model to

a subset of measured data points from the rate-distortion curve. Since this rate-distortion model uses only six parameters (see below), the necessary subset is relatively small and can be obtained with reasonable complexity.

Now we discuss the empirical rate-distortion model in detail. For a video sequence encoded at a target coding rate R_σ , the average end-to-end distortion D_σ^e consists of the encoding distortion caused by the lossy video coder, D_σ^{enc} , and the distortion due to transmission errors, including the distortion caused by overdue packets (i.e., congestion), D_σ^{cg} , and the distortion caused by lost packets (i.e., caused by link failures or transmission errors), D_σ^{loss} . That is,

$$D_\sigma^e = D_\sigma^{enc} + D_\sigma^{cg} + D_\sigma^{loss}. \quad (3.10)$$

From [102] and the results derived in Section 3.2.1, we have

$$D_\sigma^e = D_0 + \underbrace{\frac{\omega}{R_\sigma - R_0}}_{D_\sigma^{enc}} + \underbrace{\kappa \cdot (1 - p_\sigma) \cdot Pr(T_\sigma > \Delta_\sigma)}_{D_\sigma^{cg}} + \underbrace{\kappa \cdot p_\sigma}_{D_\sigma^{loss}}, \quad (3.11)$$

where D_0 , ω , R_0 , and κ are constants for a specific video codec (with fixed encoding parameters) and video sequence, which can be determined by training and curve matching. Since the model in (3.10) takes into account the effects of INTRA coding and spatial loop filtering, it matches simulation results closely [102].

3.2.3 The Global Optimal Routing Problem

For delineating an end-to-end path \mathcal{P}_σ from s_σ to d_σ , $\sigma \in \mathcal{E}$, we define the following index variables:

$$x_{ij}^\sigma = \begin{cases} 1, & \text{if } \{i, j\} \in \mathcal{P}_\sigma, \quad \{i, j\} \in \mathcal{L} \\ 0, & \text{otherwise, } \quad \{i, j\} \in \mathcal{L}. \end{cases} \quad (3.12)$$

We can now mathematically formulate the problem of application-centric optimal routing for multiple concurrent video sessions.

OPT-CLR

$$\text{Minimize: } D = \sum_{\sigma \in \mathcal{E}} D_{\sigma}^e \quad (3.13)$$

subject to:

$$\underline{R}_{\sigma} \leq R_{\sigma} \leq \overline{R}_{\sigma}, \text{ for } \sigma \in \mathcal{E} \quad (3.14)$$

$$\rho_{ij} \leq 1 - \epsilon, \quad \{i, j\} \in \mathcal{L}, \text{ for some stability tolerance } \epsilon \quad (3.15)$$

$$\sum_{j:\{i,j\} \in \mathcal{L}} x_{ij}^{\sigma} - \sum_{k:\{k,i\} \in \mathcal{L}} x_{ki}^{\sigma} = \begin{cases} 1, & \text{if } i = s_{\sigma} \\ -1, & \text{if } i = d_{\sigma} \\ 0, & \text{otherwise} \end{cases}, \quad i \in \mathcal{N}, \sigma \in \mathcal{E} \quad (3.16)$$

$$\sum_{j:\{i,j\} \in \mathcal{L}} x_{ij}^{\sigma} \begin{cases} \leq 1, & \text{if } i \neq d_{\sigma} \\ = 0, & \text{if } i = d_{\sigma} \end{cases}, \quad i \in \mathcal{N}, \sigma \in \mathcal{E} \quad (3.17)$$

$$x_{ij}^{\sigma} \in \{0, 1\}, \{i, j\} \in \mathcal{L}, \sigma \in \mathcal{E}. \quad (3.18)$$

In Problem OPT-CLR, the objective function (3.13) is the sum of the average distortion of all the concurrent video sessions. Minimizing (3.13) achieves a best utilization of network resources, as well as the best overall quality for the video sessions. When optimizing the performance of multiple users, efficiency and fairness are usually orthogonal objectives, i.e., maximizing one may lead to significant decrease in the other. In this work, we choose efficiency as our optimization objective in order to better utilize the scarce network resources in ad hoc networks. It is worth noting that choosing a different objective function, such as $\min \max\{D_{\sigma}^e\}$ or an objective function in the form of a utility function $\sum_{\sigma} f(D_{\sigma}^e)$, does not change the solution procedure, which will be presented in the next section.

There are two sets of optimization variables that form the space of feasible solutions: (i) the set of routing vectors $\{\mathbf{X}_{\sigma}\}_{\sigma \in \mathcal{E}}$; and (ii) the set of rates of video sessions $\{R_{\sigma}\}_{\sigma \in \mathcal{E}}$. The set of inequalities in (3.14) gives the range of feasible rates for each video session. In the case of streaming stored video, we have that $\underline{R}_{\sigma} = R_{\sigma} = \overline{R}_{\sigma}$ since the rate is fixed. Inequality (3.15) is the stability constraint, which ensures that the link delays are bounded. The remaining constraints (i.e., (3.16), (3.17), and (3.18)) guarantee that each path \mathcal{P}_{σ} is

loop-free.¹

The objective function (3.13) is a highly complex ratio of high-order polynomials of x -variables. The objective evaluation of a set of feasible paths involves identifying the joint and disjoint links of the paths (in order to compute the traffic load on each link), which is only possible when all the paths are completely determined. Wang and Crowcroft [115] proved that QoS routing problems having multiple additive and/or multiplicative metrics are NP-complete. Our problem has an additive delay metric and a multiplicative loss metric. In addition, our problem has much more complex relationships pertaining to the contribution of any link to the objective function, as well as coupled session delays (rather than constant link delay metrics as in [115]). As a result, we conjecture that Problem OPT-CLR is NP-complete. In the rest of this chapter, we present an effective heuristic algorithm to address this problem.

3.3 A Genetic Algorithm-Based Solution Procedure

As described in Chapter 2, GA is a *population-based* metaheuristic that is inspired by the *survival-of-the-fittest* principle, as derived from its natural evolution context. It has the intrinsic strength of dealing with a set of solutions (i.e., a population) at each step, rather than working with a single, current solution. In each iteration, a number of genetic operators are applied to the individuals of the current population in order to generate individuals for the next generation. In particular, GA uses genetic operators known as *crossover* to recombine two or more individuals to produce new individuals, and *mutation* to achieve a randomized self-adaptation of individuals. The driving force in GA is the *selection* of individuals based on their fitness (in the form of an objective function). Individuals with a higher degree of fitness will be more likely to be chosen as members of the population for the next generation.

¹Note that a feasible solution to these constraints could admit circuits whose edges are disconnected from the produced loop-free paths. However, the objective function would automatically prohibit this occurrence.

The basic assumption within this paradigm is that good solutions often share parts with optimal solutions. The survival-of-the-fittest principle ensures that the overall quality of the population increases as the algorithm progresses from one generation to the next.

The potential of GA is best explained by comparing GA with alternative approaches, such as *simulated annealing* (SA) [1] and *tabu search* (TS) [40]. Although these methods have specific strengths in solving complex optimization problems, their performance, when applied to our problem, is sensitive to the neighborhood structure definition. In addition, such approaches have a stronger tendency of being trapped at a local optimum and have slow convergence, which could be attributed to the fact that only a single solution is handled at each iteration. We will illustrate this point in Section 3.5. The solution procedure is explained further in [69, 70].

3.3.1 GA-Based Multiple-Session Routing

Chapter 2 briefly reviewed genetic algorithms and a flow-chart for a GA-based approach was described in Figure 2.2. Note that both crossover and mutation are performed with certain probabilities (θ and μ , respectively) on the individual solutions. The termination condition in Figure 2.2 could be based on the total number of iterations (generations), the maximum computing time, or a threshold of desired video distortion. In what follows, we use the example ad hoc network in Figure 3.1(a) to illustrate the steps in the GA approach. There are three video sessions in the network, with source-destination pairs $\{1, 9\}$, $\{2, 10\}$, $\{3, 8\}$, respectively.

Representation and Initialization

In order to *encode* a feasible solution in the genetic format, we need to define a *gene* first and then map a solution to a sequence of genes (*chromosome*). Naturally, we define a node as a gene and an end-to-end path can be represented as a sequence of genes. Then, for the

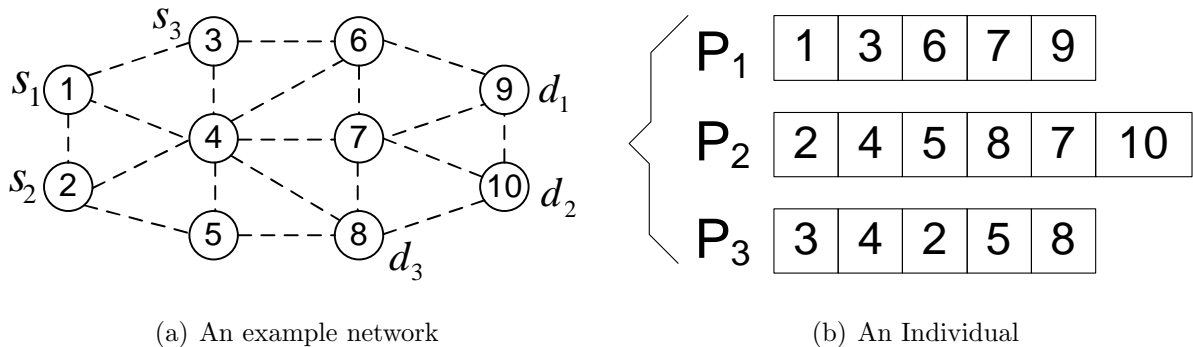


Figure 3.1: An example wireless ad hoc network and a feasible solution.

concurrent routing problem, each feasible solution (or *individual*) consists of a number of paths and, thus, a set of chromosomes, e.g., see Figure 3.1(b).

Then we need to generate a set of initial solutions (or a *population*). A simple approach would be to randomly append feasible elements (i.e., nodes with connectivity) to a partial solution. Under this approach, a construction process would start with the source node s_σ . It would then randomly choose a link incident to the current end-node of the partial path and append this link with its corresponding head-node to augment the path, until the destination node d_σ is reached. It is important to ensure that the intermediate partial path is loop-free during the process. After generating a certain set of paths for each $\{s_\sigma, d_\sigma\}$ pair independently, a population of individuals for our problem can be constructed by randomly selecting paths from the set and verifying for stability conditions. A few individuals in the population can also be initialized with the results obtained using a fast greedy heuristic, as discussed in Section 3.4. Our numerical results show that a properly-designed GA has a good exploratory power, and is not very sensitive to the quality of the individuals in the initial population.

Evaluation

The fitness function $h(\bar{x})$ of an individual, i.e., $\bar{x} = [\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3]$, is closely related to its objective function value (i.e., the total distortion D). Since the objective is to minimize the

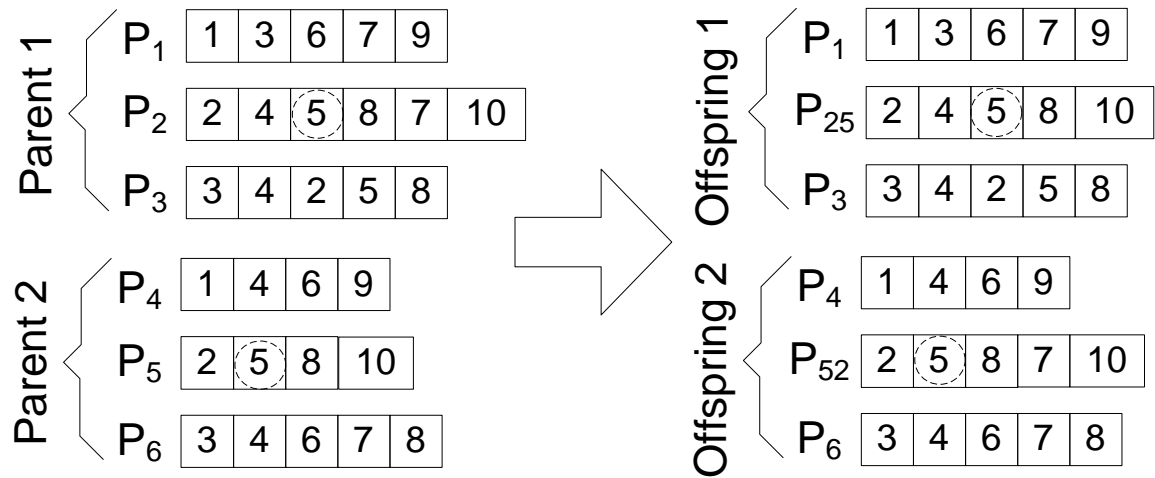


Figure 3.2: Crossover operation.

total distortion (see (3.13)), we have adopted a fitness function that is defined as the inverse of the distortion value, i.e., $h(\bar{x}) = 1/D(\bar{x})$.

Selection

During this operation, we select individuals that have a better chance or potential to produce “good” offspring in terms of their fitness values. We use the popular *Tournament* selection scheme [7], which randomly chooses m individuals from the population each time, and then selects the best of these m individuals in terms of their fitness values. By repeating this procedure multiple times, a new population can be selected.

Crossover

Crossover mimics the genetic mechanism of reproduction in the natural world, in which genes from parents are recombined and passed to offspring. The crossover operation may create new individuals, exposing the search process to a new area of the fitness landscape. Figure 3.2 illustrates one possible crossover implementation. The decision of whether or not

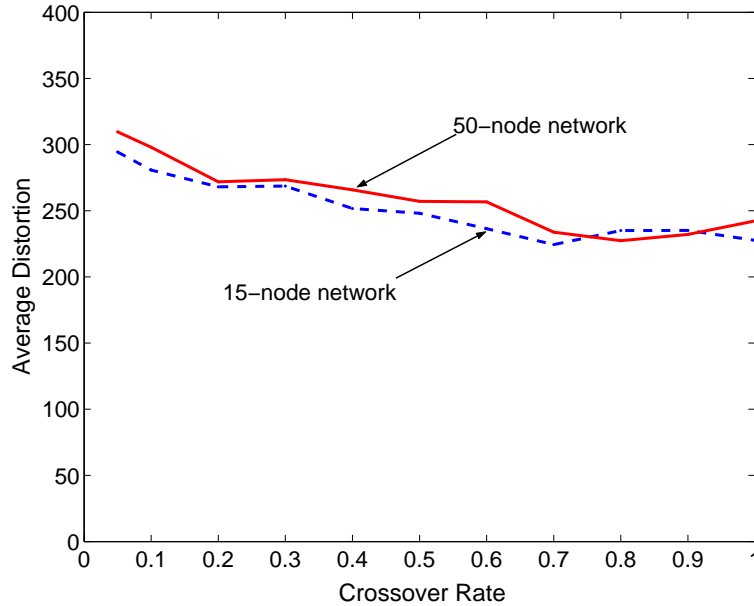


Figure 3.3: Impact of crossover rate θ on average distortion.

to perform a crossover operation is determined by the *crossover rate* θ .

For two parent individuals $x_1 = [\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3]$ and $x_2 = [\mathcal{P}_4, \mathcal{P}_5, \mathcal{P}_6]$, we could randomly pick a session, say Session 2 (\mathcal{P}_2 in x_1 and \mathcal{P}_5 in x_2). If one or more common nodes exist in these two chosen paths, we could select the first such common node that exists in \mathcal{P}_2 , say g_r , $g_r \notin \{s_2, d_2\}$ (node 5 in Figure 3.2). We can then concatenate the nodes $\{s_2, \dots, g_r\}$ from \mathcal{P}_2 with the nodes $\{g_{r+1}, \dots, d_2\}$ in \mathcal{P}_5 (where g_{r+1} denotes the next downstream node of g_r in \mathcal{P}_5) to produce a new path \mathcal{P}_{25} . Likewise, using the first such node $g_{r'}$ in \mathcal{P}_5 that repeats in \mathcal{P}_2 (which may be different from g_r), we can concatenate the nodes $\{s_2, \dots, g_{r'}\}$ from \mathcal{P}_5 with the nodes $\{g_{r'+1}, \dots, d_2\}$ in \mathcal{P}_2 to produce a new path \mathcal{P}_{52} . The two new individuals generated in this manner are $[\mathcal{P}_1, \mathcal{P}_{25}, \mathcal{P}_3]$ and $[\mathcal{P}_4, \mathcal{P}_{52}, \mathcal{P}_6]$, as illustrated in Figure 3.2. If \mathcal{P}_2 and \mathcal{P}_5 are disjoint, we could swap the entire path \mathcal{P}_2 with \mathcal{P}_5 instead.

Figure 3.3 provides an idea of how the crossover rate θ impacts the average distortion in the OPT-CLR problem. This way, we can identify the best setting for the crossover rate for future experiments. Here, we fixed the initial population size to 20 and mutation rate to

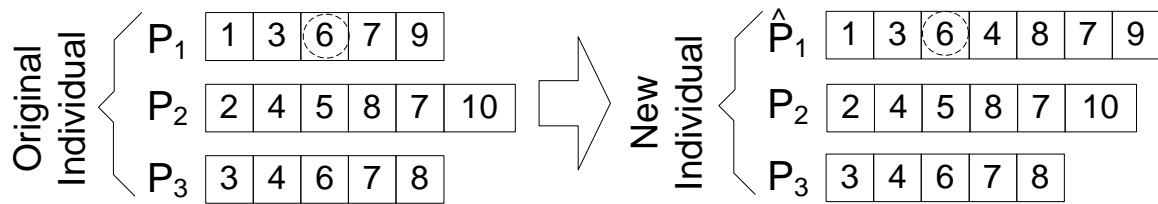


Figure 3.4: Mutation operation.

0.5, while the crossover rate is increased in steps of 0.1, from 0.1 to 1. The results are shown here for a 15-node network with 4 concurrent video sessions and a 50-node network with 8 concurrent video sessions. It can be seen from Fig. 3.3 that the average distortion gradually decreased as the crossover rate increased for both the networks. Based on this result, we could set the crossover rate between 0.4 and 0.7 for the subsequent simulations, since further increases in the crossover rate only achieves a marginal reduction in distortion, but with an increased computational burden.

Mutation

The objective of the mutation operation is to *diversify* the genes of the current population, which helps prevent the solution from being trapped in a local optimum. This is a significant advantage over traditional trajectory methods. However, just as some malicious mutations could happen in the natural world, mutation in GA may produce individuals that have worse fitness values than the current solutions. In such cases, some “filtering” operation is needed (e.g., the selection operation) to reject such “bad” genes and to drive GA toward optimality.

Mutation is performed on an individual with probability μ (called *mutation rate*). For better performance, we propose a schedule to vary the mutation rate within $[\mu_{min}, \mu_{max}]$ over iterations (rather than using a fix μ). The mutation rate is first initialized to μ_{max} ; then as generation number k increases, the mutation rate gradually decreases to μ_{min} , i.e.,

$$\begin{cases} \mu_0 = \mu_{max} \\ \mu_k = \mu_{max} - \frac{k \cdot (\mu_{max} - \mu_{min})}{T_{max}}, \end{cases} \quad (3.19)$$

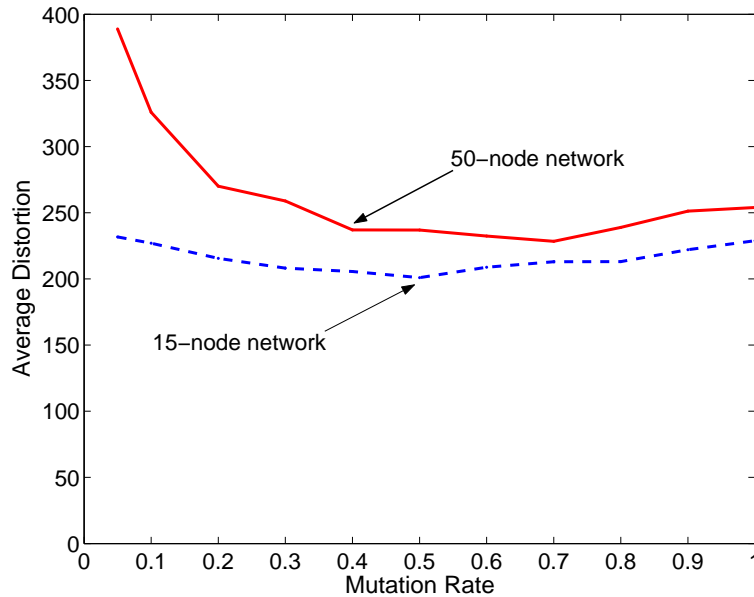


Figure 3.5: Impact of mutation rate μ on average distortion.

where T_{max} is the maximum number of generations. Our results show that varying the mutation rates over generations significantly improves the on-line performance of the GA-based routing scheme. In essence, such schedule of μ is similar to the cooling schedule used in SA. Such a *hybridized* GA yields better convergence performance than a pure GA.

Figure 3.4 illustrates the mutation of an individual $\bar{x} = [\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3]$. First, we choose a path \mathcal{P}_σ , $\sigma \in \{1, 2, 3\}$, from \bar{x} using equal probabilities of selection. Then, we randomly select an integer value r in the interval $[2, |\mathcal{P}_\sigma| - 1]$, where $|\mathcal{P}_\sigma|$ denotes the cardinality of \mathcal{P}_σ , and let the partial path $\{s_\sigma, \dots, g_r\}$ be \mathcal{P}_σ^u , where g_r is the r -th node along \mathcal{P}_σ . Finally, we could use any constructive approach to build a partial path from g_r to d_σ , denoted as \mathcal{P}_σ^d , that does not repeat any node in \mathcal{P}_σ^u (other than g_r). If no such alternative segment exists between g_r and the destination node d_σ , we keep the path intact. Otherwise, a new path can now be created by concatenating the two partial paths as $\mathcal{P}_\sigma^u \cup \mathcal{P}_\sigma^d$. For the example in Figure 3.4, \mathcal{P}_1 is chosen for mutation and node 6 is chosen to be the mutation point, yielding a perturbed path $\hat{\mathcal{P}}_1$ which replaces \mathcal{P}_1 . The new individual thus created is $\hat{x} = [\hat{\mathcal{P}}_1, \mathcal{P}_2, \mathcal{P}_3]$.

Figure 3.5 provides an idea of how the mutation rate μ impacts the average distortion in the OPT-CLR problem. As before, we fixed the initial population size to 20 and crossover rate to 0.7, while the mutation rate is increased in steps of 0.1, from 0.1 to 1. The results are shown here for a 15-node network with 4 concurrent video sessions and a 50-node network with 8 concurrent video sessions. It can be observed from Fig. 3.5 that the average distortion curves have a unimodal shape (more obvious for the 50-node case). Here again, the computational complexity increases with the mutation rate μ .

3.3.2 Determining Video Rates

As discussed in Section 3.2, the search space of Problem OPT-CLR is the Cartesian product of the set of feasible paths and the set of feasible video rates. The optimal values of these parameters jointly produce the lowest total distortion. In the GA-based approach, the optimal session rates are determined when evaluating the individuals. More specifically, we first use the procedure described in Figure 2.2 to evolve a population, assuming that each session uses its minimum rate \underline{R}_σ , $\sigma \in \mathcal{E}$. Then, during each iteration, we determine the corresponding optimal rates for each individual and use this rate to compute its fitness value.

Since an individual is a solution with a *given* set of feasible paths $\{P_\sigma\}_{\sigma \in \mathcal{E}}$, the problem of finding optimal video rates for a set of given paths (OPT-Rate) can be expressed as:

OPT-Rate

$$\mathbf{Minimize:} \quad D(x_k) = \sum_{\sigma \in \mathcal{E}} D_\sigma^\epsilon \quad (3.20)$$

$$\mathbf{subject\ to:} \quad \underline{R}_\sigma \leq R_\sigma \leq \overline{R}_\sigma, \quad \forall \sigma \in \mathcal{E} \quad (3.21)$$

$$\rho_{ij} \leq 1 - \epsilon, \quad \forall \{i, j\} \in \mathcal{L}. \quad (3.22)$$

Note that we do not need to solve OPT-Rate for streaming stored video, since the video rates are fixed for such applications. OPT-Rate is a nonlinear optimization problem with nonlinear constraints. It can be efficiently solved using an iterative procedure based on

the *Sequential Quadratic Programming (SQP) method* [81], which is considered one of the most effective methods for solving nonlinear programming problems due to its superlinear convergence rate.

3.4 A Greedy Algorithm For Initial Solutions

In this section, we present an efficient greedy algorithm for solving Problem OPT-CLR. The algorithm is based on the observation of key characteristics of the video distortion model. This algorithm computes low loss and low congestion paths for the video sessions using an empirical compound routing metric.

Before describing the greedy algorithm, we first examine the total end-to-end distortion D_σ^e of a session $\sigma \in \mathcal{S}$ (see (3.10) and (3.11)). The distortion due to encoder, D_σ^{enc} , is a monotonically *decreasing* function of video rate R_σ . The distortion due to congestion, D_σ^{cg} , on the other hand, is a monotonically *increasing* function of R_σ , as well as the rates of all other sessions R_i , that share one or more links with session σ . Both of these two terms are constrained by the stability constraint (3.15) and are thus determined by the available bandwidths of the path links. The third term D_σ^{loss} , the distortion caused by lost packets, is an increasing function of the link loss probabilities. In order to minimize the video distortion for session σ , we need to find paths having the highest end-to-end bandwidth, the minimal congestion, and the lowest end-to-end loss rate.

We also observe that the D_σ^{enc} curve is concave: when R_σ increases beyond a certain threshold, further increasing R_σ will only cause marginal reduction in D_σ^{enc} . For example, we plot D_σ^{enc} for an H.263 coder with typical settings (e.g., Intra Rate 1/15 and frame rate 30 fps) using the Quarter Common Intermediate Format (QCIF) formatted ‘‘Foreman’’ sequence, in Figure 3.6. We observe that there is a decrease of about 100 in D_σ^{enc} when R_σ increases from 40 Kb/s to 150 Kb/s. When R_σ further increases from 150 Kb/s to ∞ , the corresponding total reduction in D_σ^{enc} is only about 20. A high rate will cause congestion

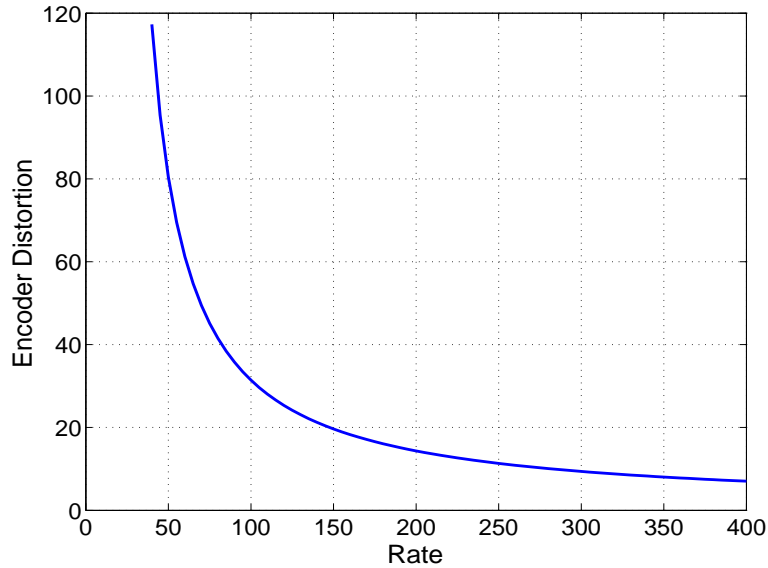


Figure 3.6: H.263 rate-distortion curve using QCIF “foreman” sequence.

in the bottleneck link, resulting in a much larger increase in D_σ^{cg} . For practical R_σ values, reducing congestion conditions in the network would be more effective than increasing video rates in improving the overall video quality.

Based on these observations, we describe a greedy heuristic (called GREEDY) in Figure 3.7, for Problem OPT-CLR. In the GREEDY algorithm, an empirical compound link cost $c_{ij}(1 - p_{ij})$, which we call the *effective available bandwidth*, is used. For a given path, its end-to-end effective available bandwidth is the minimum among those of its links. By computing the path with the maximum effective available bandwidth, GREEDY finds the current “widest” path for a session, which has the potential of supporting higher video rates and having low loss rates. Since both link capacity and loss probability are considered in the compound link cost, GREEDY may produce near-optimal solutions to Problem OPT-CLR, as will be shown in Section 3.5. For each session, the maximum effective-available-bandwidth path could be computed using the algorithm presented in [65], with a time complexity of $O(|\mathcal{L}| \cdot \log^* |\mathcal{N}|)$, where $\log^* n$ is the *iterated logarithm function*. The overall time complexity of GREEDY is $O(|\mathcal{S}| \cdot |\mathcal{L}| \cdot \log^* |\mathcal{N}|)$. This algorithm is explained further in [68].

Algorithm GREEDY

Set the cost of each link $\{i, j\}$ to $c_{ij}(1 - p_{ij})$, $\forall \{i, j\} \in \mathcal{L}$;
 For every video session $\sigma \in \mathcal{S}$;
 Use the algorithm in [65] and use the costs defined earlier to find the path
 having the maximum end-to-end cost. Let this path be \mathcal{P}_σ ;
 Decrease the bandwidth of every link on \mathcal{P}_σ by \underline{R}_σ , i.e., setting the link
 costs as $(c_{ij} - \underline{R}_\sigma) \cdot (1 - p_{ij})$, $\forall \{i, j\} \in \mathcal{P}_\sigma$;
 After the paths for all sessions are found, solve OPT-Rate to determine the
 rates for the sessions.

Figure 3.7: A greedy algorithm for computing initial solutions.

For the set of computed paths $x_k = \{P_\sigma\}_{\sigma \in \mathcal{S}}$ (which potentially has the minimal congestion and path loss), we solve the nonlinear optimization Problem OPT-Rate, which further reduces the overall video distortion by finding the near-optimal video rates for the sessions. It can be easily verified that the path set found by GREEDY is realizable, i.e., it satisfies all the constraints of Problem OPT-CLR. Therefore, the resulting distortion is an upper bound of the optimal distortion. The computational complexity of GREEDY is much lower than GA. It could be used to compute a good initial solution for GA and thus speed up the GA convergence. In practice, GREEDY can be used to quickly compute a set of near-optimal paths for the video sessions. Then, the GREEDY solution could be included into the GA initial population to be further improved, if possible. When GA terminates, the video sessions could switch to the refined routes for better performance.

3.5 Simulation Results

In this section, we present simulation results. In each experiment, an ad hoc network is generated at random within a rectangular region. Each video session has a rate between 100 Kb/s and 400 Kb/s. We use an H.263+ codec and the 400-frame, QCIF “Foreman” video sequence. The video is encoded with an Intra rate of 1/15 and a frame rate of 30 frames/second. The corresponding rate-distortion parameters are obtained from [102]. Failure probabilities

of the wireless links are uniformly distributed between [1%, 10%]; the bandwidth of a link is uniformly distributed between [100 Kb/s, 400 Kb/s]. For all the results reported in this section, the exponential model (3.2) is used.

As discussed, there are three key parameters for the proposed GA approach: the *population size*, the *crossover rate* θ , and the *mutation rate* μ . Through simulation studies, we find that the performance of the GA-based routing is quite stable for a wide range of parameter settings. For the results reported in this section, the population size is seven, $\theta = 0.4$, and $\mu = 0.2$. The greedy algorithm presented in Section 3.4 is used to generate an initial solution, while the remaining initial solutions are generated using the random constructive method discussed in Section 3.3.1.

3.5.1 Dissecting End-to-End Distortion

In Figure 3.8, we plot the average distortion versus decoding deadline for a 15-node network with four concurrent video sessions. We set the same decoding deadline value for all of the sessions. It can be observed that the average distortion is a decreasing function of decoding deadline. For small decoding deadline values, most of the video packets are overdue, resulting in high distortion. As decoding deadline increases, the average distortion quickly decreases since more and more packets are now received in time, contributing to an improved video quality. As the decoding deadline further increases, the real-time application essentially reduces to an elastic data application, where any received packet is useful for improving video quality. The same trend exists for all cases we simulated, although the specific “knee-point” depends on the network and video parameters.

As discussed, the end-to-end distortion of a video session consists of three components: encoding distortion D^{enc} , distortion due to packet losses D^{loss} , and distortion due to congestion D^{cg} . In Figure 3.8, the dash-dotted (lowest) curve is for D^{enc} . From the empirical distortion model (3.11), D^{enc} is a function of the target encoding bit rate, which is determined by the end-to-end bandwidth and the congestion condition of the path (i.e., the rate

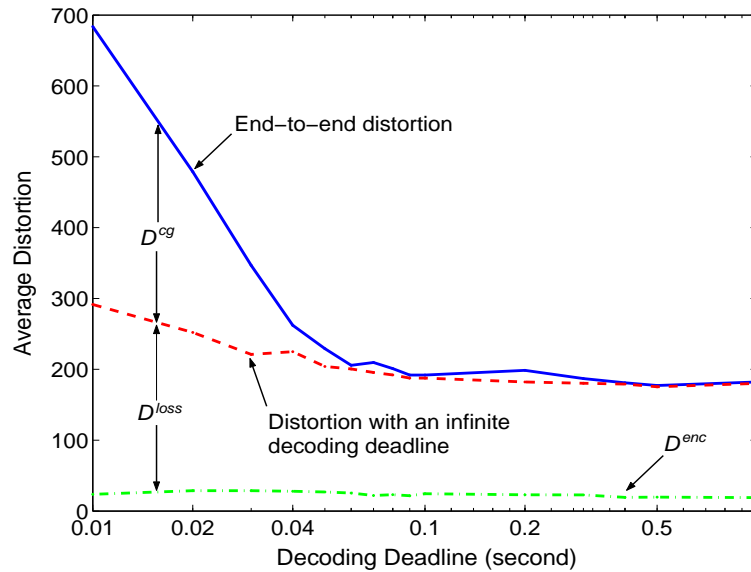


Figure 3.8: Distortion versus decoding deadline.

is computed by solving Problem OPT-Rate). It is relatively constant for various decoding deadlines in this experiment.

In Figure 3.8, the dashed curve is the average distortion computed using both packets that are received in time and packets that are overdue. The difference between the two lower curves corresponds to D^{loss} , which is determined by the end-to-end loss rate. Note that D^{loss} decreases gradually as decoding deadline increases. This is because for very tight delay constraints, the GA-based routing may give more preference to delay over loss in order to meet the delay constraint. That is, GA-based routing may choose a path having a lower end-to-end delay, even though the path may have a higher end-to-end loss. As the delay constraint gets relaxed, GA is allowed to choose paths having a lower end-to-end loss rate while still satisfying the delay constraint. The difference between end-to-end distortion and the dashed curve is D^{cg} . When congestion occurs, packets experience large delays and many of them arrive beyond the decoding deadline, causing high distortions. As decoding deadline increases, the end-to-end distortion curve converges to the dashed curve, which indicates that congestion has little impact on video distortion when the delay constraint is relaxed.

3.5.2 Performance Bounds

One interesting question regarding the GA-based routing is how close its solutions are to the optimal solution. For small networks, it is possible to find the optimal solution in a reasonable amount of time by using an exhaustive search (ES), which, however, is infeasible when network size becomes large. In Table 3.2, we compare the GA solutions to the global optimal solutions computed by exhaustive search, for small networks with three concurrent video sessions. The decoding deadline Δ_σ is set to 0.1 s for all the video sessions. GA runs for about 50 iterations in each simulation and each GA distortion value is the average of 30 runs. In Table 3.2, the normalized difference is computed as $|GA - ES|/ES$.

We find that GA performs consistently well in comparison to the optimal solution. In every case, GA either finds the exact global optimal or finds a near-optimal solution with a negligible normalized difference. Moreover, for the distortion values obtained by 30 executions using the same network, the standard deviation is very small for all the cases examined, indicating that the GA performance is very stable. Finally, the GA computation time (a few hundred milliseconds, on a Pentium4 2.4 GHz computer with 512 MB memory) is only a tiny fraction of the time required to perform the exhaustive search (from 2.5 to 9.1 hours).

We also present the GREEDY results in Table 3.2. We find that the GREEDY solutions are also quite competitive. In Topologies 2 and 4, GREEDY finds the exact global optima; in the remaining four cases, the GREEDY distortions are close to the the global optima.

3.5.3 Comparison with Trajectory Methods

For comparison, we implement two representative trajectory metaheuristic methods, namely, SA [1] and TS [40]. For best performance, we set the initial *temperature* c_0 in SA to 1 and the temperature decaying ratio ω to 0.5. The *tabu list* for the TS implementation is chosen to be 5-units. Implementation details for SA and TS are provided in Appendix A.

In Figure 3.9, we plot the evolutions of the total distortions obtained by GA, SA, and

Table 3.2: Average distortion values found by the algorithms

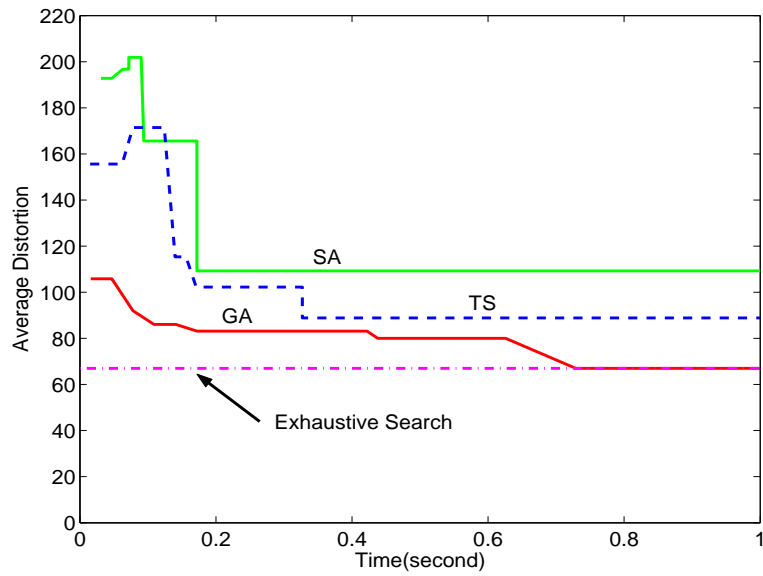
Topology	Topology 1	Topology 2	Topology 3	Topology 4	Topology 5
Network Size	9-node	9-node	11-node	11-node	11-node
GREEDY	230.20	81.77	88.29	115.32	77.55
ES	213.94	81.77	67.01	115.32	67.20
GA	214.31	81.77	67.76	115.32	67.63
Norm. Diff.	0.17%	0	1.11%	0	0.62%
Std. Dev.	0.55	0	6.93	0	3.08

TS. Figure 3.9(a) is obtained using an 11-node network with three concurrent video sessions (for which the global optimum could be found by exhaustive search), while Figure 3.9(b) is obtained for a 50-node network with 10 concurrent sessions. Simulation time for all three algorithms is 1 s. The decoding deadline is set to 0.5 s for the 50-node network simulations, and 0.1 s for the 11-node network simulations.

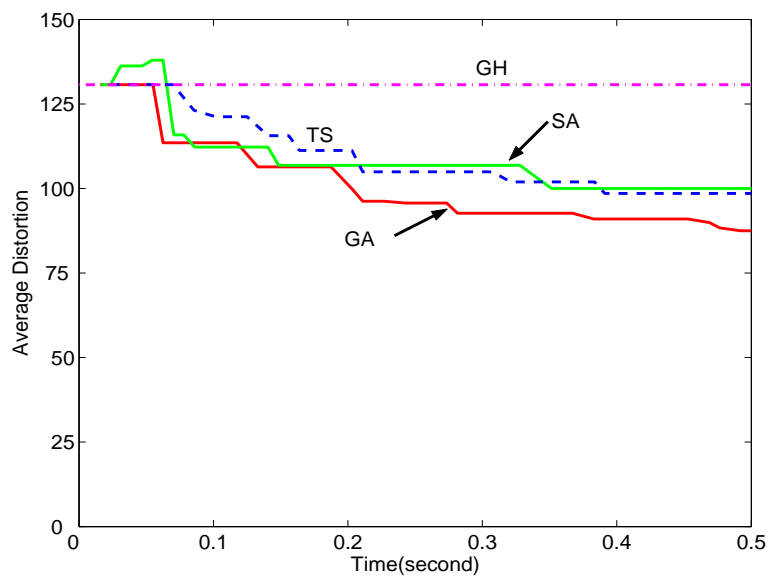
From Figure 3.9(a), we observe that GA converges to the global optimal very quickly, while both SA and TS are trapped at local optima (i.e., no improvement after a large number of iterations). This is due to the fact that GA could explore the fitness landscape in parallel by evolving a population of solutions, while trajectory methods only maintain a single solution, and have a higher tendency of being trapped at a local minimum (despite the fact that they all incorporate explicit strategies to avoid such events).

3.5.4 Comparison with Network-centric Routing

We compare GA-based routing with traditional network-centric routing in solving Problem OPT-CLR. We implement a shortest path routing algorithm (SP) using hop-count as routing metric, and a disjoint shortest path routing algorithm (DSP) using loss rate as routing metric [24], both based on Dijkstra’s Algorithm. In DSP, the link cost is set to



(a) 11-node network, four video sessions.



(b) 50-node network, 10 video sessions.

Figure 3.9: GA versus trajectory methods.

$\log(1/(1 - p_{ij}))$, $\{i, j\} \in \mathcal{L}$. As a result, the minimum cost path has the highest end-to-end success probability. In both network-centric algorithms, we first find the optimal set of paths using the minimum video rates $\{\underline{R}_\sigma\}_{\sigma \in \mathcal{S}}$, and then solve Problem OPT-Rate over the path set to compute the corresponding optimal video rates. In order to meet the link stability condition in SP, each time when a path is found, we subtract the minimum rate of the corresponding video session from the capacity of each link along this path, while the next path is found in the “reduced” graph. The computation time of GA is between 500 ms to 900 ms, while the computation time for SP or DSP ranges from tens of milliseconds to about 200 ms.

Overall Performance

Figure 3.10 plots the average distortions found by the three algorithms (i.e., GA, SP, and DSP) for various decoding deadlines. The network consists of 50 nodes with 10 video sessions. We find that for very small decoding deadlines, the delay requirement is so stringent that all the three schemes yield high distortion. On the other hand, for very large decoding deadlines, the delay requirement is so loose that all the three schemes can achieve a low total distortion, as long as the stability condition is satisfied. The more interesting region, however, lies between these two extremes, where a well-designed routing scheme can achieve a better performance by finding optimal routes for the video sessions. Within this region, GA outperforms SP and DSP by a significant margin. In Figure 3.10, the GA average distortion quickly decreases as decoding deadline increases, while the SP and DSP average distortions are persistently high for small and medium decoding deadlines (indicating that most of the video packets are overdue in these cases). For example, when $\Delta = 0.2$ s, the difference between the average distortions achieved by GA and DSP is 683.8, which translates to a 9.03 dB reduction in PSNR. Similarly, GA achieves a 449.6 reduction in total distortion over SP, which is a 7.49 dB improvement in average PSNR. These improvements are significant, since generally a half dB difference in PSNR is noticeable [116].

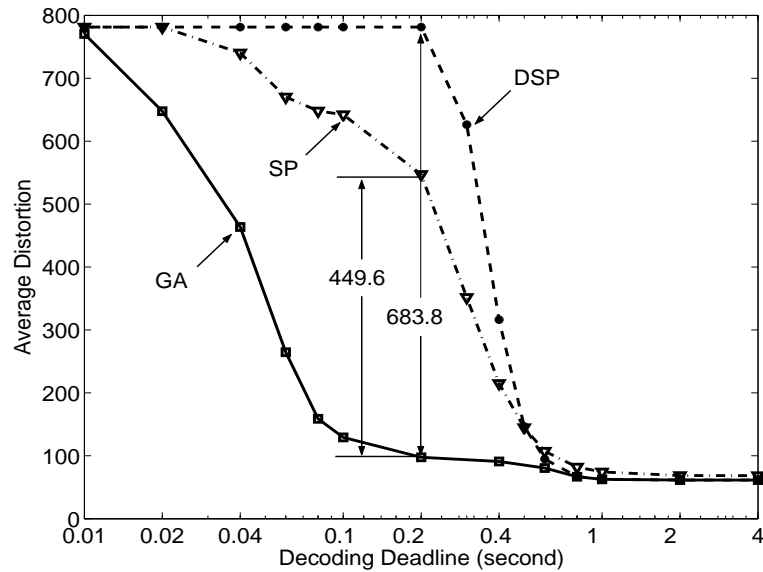


Figure 3.10: Average end-to-end distortion versus decoding deadline.

In Figure 3.11, we examine the impact of video traffic load on the routing performance. We compare the total distortions found by GA, SP, and DSP while increasing the number of video sessions in the 50-node network. The decoding deadline is 0.5 s for all of the video sessions. As expected, both SP and DSP produce higher total distortions than GA, due to the fact that they only use network layer metrics in routing. More specifically, SP does not consider the interaction of the video sessions. Although it computes the shortest path for each session, different sessions may share bottleneck links, resulting in congestion and high packet overdue rates. On the other hand, DSP goes to the other extreme by not allowing the sharing of any links, even when a link has abundant bandwidth and a low loss rate. As a result, some “bad” links (i.e., low capacity or high failure probability links) or paths having a large number of hops will be used in order to satisfy the disjointness requirement, resulting in an increased total distortion. Another interesting observation from Figure 3.11 is that the total distortion obtained by GA increases linearly with the number of sessions, which implies that the average distortion for each session is relatively constant, despite that the video traffic load has increased nearly ten folds.

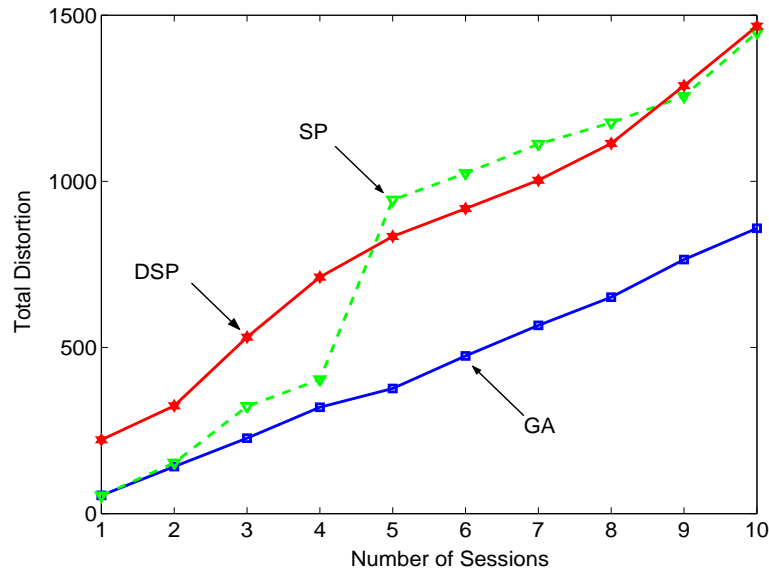


Figure 3.11: Total distortion versus number of video sessions.

Performance of Individual Sessions and Frames

So far, we have investigated the impact of optimal routing on total video distortion. Now, we examine the quality of individual video sessions. Specifically, we transmit encoded video on those paths found by GA, SP, and DSP, respectively, and compute PSNRs for decoded video frames that are possibly corrupted due to transmission errors and congestion.

The distortion values for the individual sessions obtained by the three algorithms are plotted in Figure 3.12 for a 50-node network with 10 video sessions. We find that for most of the sessions (except for Sessions 4 and 7), GA achieves a much lower distortion than the two network-centric algorithms. Although for Sessions 4 and 7, the GA distortion is higher than that of SP or DSP, the difference is negligible in both cases. The session distortions of SP and DSP are highly diverse, while the GA sessions have relatively even distortions, despite the fact that only the total distortion is minimized. The total distortion achieved by GA is 943.2, which is much lower than those achieved by SP (1956.0) and DSP (1738.4).

The PSNRs of decoded frames for Session 5 are plotted in Figure 3.13. The frames sent on

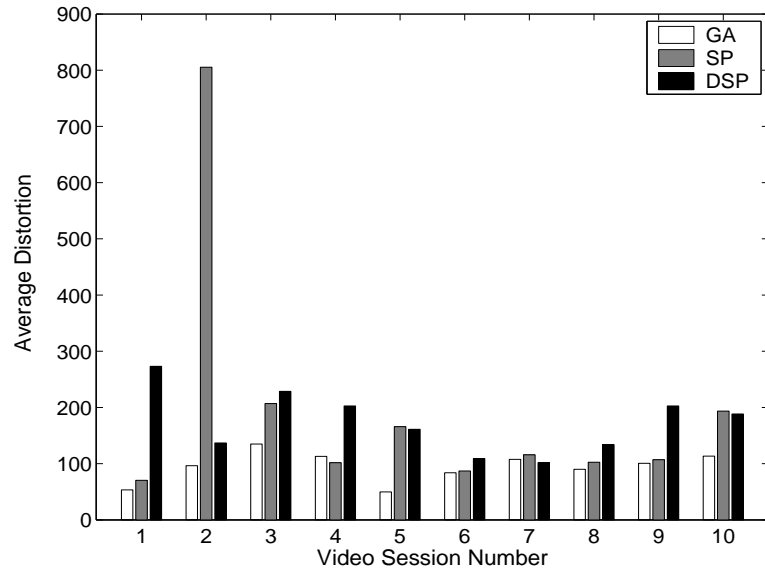


Figure 3.12: Average distortion values for each video session in a 10-session, 50-node network obtained by different algorithms.

the GA paths have much higher PSNR values than those sent on the SP or DSP paths. The average Session 5 PSNR (over the 400 frames) achieved by the GA-based routing is 31.16 dB, while the average PSNRs obtained by SP and DSP are 25.93 dB and 26.06 dB, respectively. Such significant gains (over 5 dB in both cases) are due to the fact that the application layer video quality (rather than network layer metrics) is explicitly optimized and the routing for multiple sessions is jointly optimized. To illustrate the perceived video quality, we also present decoded Frame 148 in Figure 3.14, including the original YUV formatted frame and those obtained by GA, SP and DSP. The GA frame has a slightly lower quality than the original YUV frame (Figure 3.14(a)), due to information loss caused by compression and transmission errors. The decoded frame in Figure 3.14(b) has a much better visual quality than the two frames in Figs. 3.14(c) and 3.14(d).

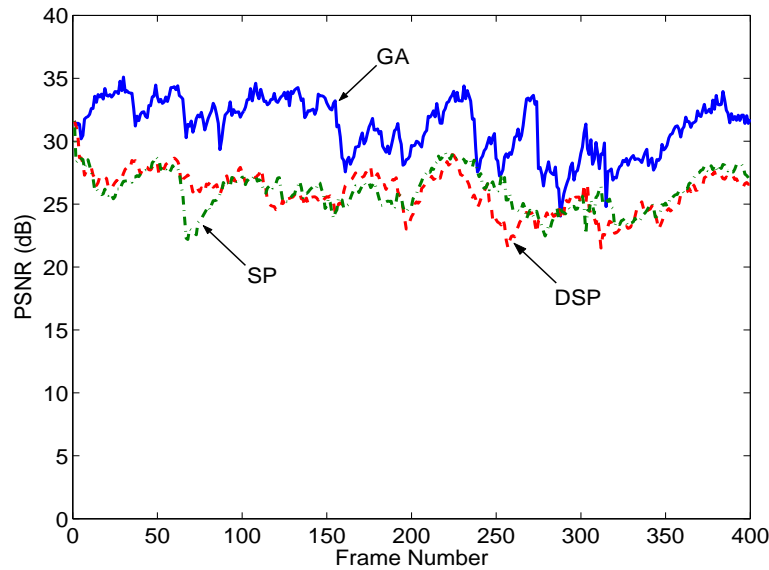


Figure 3.13: PSNRs of decoded frames for video session Five.

3.6 Related Work

Traditional network-layer QoS-routing for ad hoc networks [8,9,18,62,83] have been a topic of intense research for many years. In such problems, the focus has been on addressing network layer routing problems from various perspectives. For example, delay/bandwidth constrained least-cost routing [18], end-to-end resource guarantees [62,83], and associativity of wireless links [106]. The problems addressed in this chapter differs from the traditional QoS-routing problems in that, most of these prior efforts do not explicitly formulate the objective function as an application layer metric while the constraint set is comprised of mainly network/link layer resources and parameters.

Several path/server selection schemes have been developed for video streaming in the Internet. Specifically, in [6], the authors present three heuristics on selecting multiple description (MD) video servers in a Content Delivery Network (CDN). Although shown to be quite effective for CDNs, these algorithms only select servers based on some performance metrics. It is not clear how to determine the paths to the servers for minimizing video distortion. In a recent work [12], Begen *et al.* studied the problem of path selection for



(a) Original.



(b) GA.



(c) SP.



(d) DSP.

Figure 3.14: Reconstructed frame 148 from the “foreman” sequence.

MD video streaming in service overlay networks. The optimal routing problem is solved via exhaustive search which could have an exponential complexity. Subsequently, the authors proposed an improvement algorithm in [11] by taking advantage of the special structure of the underlying network. This improvement algorithm may not be feasible for ad hoc networks that are infrastructureless and have dynamic topologies. Finally, it is worth noting that the interaction of concurrent video flows is not considered in these prior works.

There exist several prior efforts on applying GA to address network layer problems, e.g., shortest path routing [2, 36, 100] and multicast QoS routing [10]. GA has also been explored to address various other networking problems such as admission control [117], channel assignment [39], network design [29], scheduling [77] and buffer management [32]. In particular, Ahn and Ramakrishna [2] applied GA to the shortest path routing problem and compared its performance to the Dijkstras algorithm. In addition, the authors also presented an elegant analysis on how to determine the optimal population size for the problem at hand. These efforts have made an important first step towards exploring the potential of GA for network optimization. The research presented in this chapter builds upon these earlier efforts and extends GA's potential to address the more complex cross-layer, video-centric optimization problem. The problem investigated in this chapter is substantially more difficult since it exploits the design and optimization space across multiple layers.

3.7 Summary

In this chapter, we studied the problem of optimally supporting multiple concurrent video communication sessions in an ad hoc network. We formulated a network-wide optimal routing problem that minimizes the total distortion of all video sessions. We modeled the end-to-end video distortion as a function of routing layer behavior. Our formulation seamlessly integrates the interaction of competing video flows and network layer link metrics, which allows for computing of optimal routes as well as determining the optimal rates for the video

sessions. We developed a highly effective solution procedure based on the GA framework for computing near-optimal routes for all the concurrent video sessions. Several computational experiments were performed using randomly generated network topologies ranging from 9-nodes to 50-nodes. The efficacy of the proposed solution methodology is explored by comparing with exhaustive search methods in finding the optimal. The GA-based algorithm was also compared with a fast greedy heuristic for small sized networks, and for larger networks, the fast greedy heuristic was used initialization of the population used in the GA-based algorithm. Furthermore, for larger sized networks, where exhaustive search is near impossible, the performance of the GA-based algorithm is compared with other trajectory based metaheuristics, as well as other network layer routing algorithms. The results support the robustness of the proposed approach. In particular, the GA-based algorithm consistently outperforms the trajectory based algorithms that were compared, namely simulated annealing and tabu search. The GA-based approach also outperformed the traditional network-centric routing schemes, namely the shortest path and disjoint shortest path routing algorithms, by finding paths that had considerably lower packet loss and delay.

Chapter 4

Path Selection and Rate Allocation for Concurrent Video Sessions

4.1 Introduction

Among various mechanisms for quality of service (QoS) provisioning, path selection is arguably one of the most important mechanism for supporting video sessions in multihop wireless networks. This is because the quality of a received video is highly dependent on the quality of the path(s) in terms of loss, delay, and delay variations. An efficient path selection algorithm should choose high quality paths for a video session, and would be especially appealing if it makes path selection decisions directly based on the application layer performance such as video distortion.

We consider address the following problems: (i) given a set of available paths, find an optimal subset of path(s) to be used, (ii) find the optimal rate that the video should be encoded and transmitted, and (iii) partition the rate among the chosen paths, such that the reconstructed video quality is maximized. These problems are tightly coupled: the optimal rate is determined by the path selection and traffic proportioning strategy, and *vice*

versa. Furthermore, the optimal rate vectors for all the concurrent video sessions are also closely dependent on each other: changing the rate allocation of one session may degrade the optimality of the rate vectors for other sessions. The interactions of competing video sessions in resource sharing must be addressed, and an efficient strategy should take these factors into consideration in a holistic manner. Furthermore, although heuristic algorithms could be applied to compute such rate vectors, it is more important to provide bounded *optimality gap* in many cases, i.e., the gap between a feasible solution and the global optimum. In other words, it is important to explore performance limits in such complex optimization problems.

In this chapter, we consider the problem of optimal path selection and rate allocation for concurrent video sessions in an ad hoc network. For each video session, there is a set of directed paths from the source node to the destination node. Such paths could be precomputed by an underlying routing protocol based on certain quality constraints (e.g., loss rate, available bandwidth, etc.). The sessions compete for network resources, such as bandwidth, when their paths share wireless links. Our goal is to determine the optimal rates for all the video sessions, as well as how to split the optimal rate over the paths for each session, such that the overall quality of the reconstructed video are optimized. Note that such a *multi-path transport* approach has many advantages (e.g., see [72] and references therein), including bandwidth aggregation, load balancing, and improved error resilience against transmission errors and link failures.

We formulate the above path selection and rate allocation problem, which optimizes video application performance (i.e., distortion) by seamlessly incorporating the network layer parameters. That is, we take a *cross-layer* approach by modeling the application layer performance metric, video distortion, as a function of network layer behavior (i.e., path selection and traffic proportioning). For realtime applications, each data packet is associated with a *decoding deadline*. A packet must be successfully delivered before its decoding deadline for the packet to contribute to the reconstructed video quality. The impact of congestion (i.e., packet delay distribution) should be taken into consideration, in addition to that of link failures. In our formulation, interactions among competing video sessions contribute to

the delays on shared links, and the end-to-end delay distribution is derived by applying the *Chernoff bound* approximation [19]. Such a formulation provides a guideline on choosing a set of optimal paths and rate vectors that provide the best video quality.

This formulation results in a nonconvex non-polynomial programming problem with a complex objective function and constraints. Each session's distortion is a function of the rates of all other sessions. Since non-polynomial programming problems are NP-hard in general, and our problem does not appear to possess any simplifying special structure, it is likely also NP-hard. Although this class of problems can be solved using metaheuristic algorithms (e.g., Genetic Algorithms [7]) for near-optimal solutions, we pursue to provide guarantees on the optimality gap of such near-optimal solutions. *The main contribution* of this work is a specialized branch-and-bound-based framework, predicated on RLT, that can produce ϵ -optimal solutions to the problem of optimal path selection and rate allocation for concurrent video sessions. The proposed solution procedure is computationally efficient and provides an elegant trade-off between optimality and computation complexity.

The remainder of this chapter is organized as follows. In Section 4.2, we formulate the problem of optimal path selection and rate allocation for concurrent video sessions and develop a tight linear programming relaxation of the problem through suitable transformations of variables and polyhedral approximations. Section 4.3 describes a specialized branch-and-bound and RLT-based solution procedure to solve the formulated problem. Some computational results and evaluation of alternative algorithmic strategies are presented in Section 4.4. Related work is discussed in Section 4.5, and finally Section 4.6 concludes this chapter.

4.2 Problem Formulation

As we discussed in Chapter 3, we model a wireless ad hoc network as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of vertices, representing mobile nodes, and \mathcal{L} is the set of wireless links in the network. We focus on network layer link metrics, assuming that lower layer

dynamics can be translated into these network-layer metrics. For instance, we characterize each wireless (directed) link $\{i, j\}$ by available bandwidth of link $\{i, j\}$, c_{ij} ; and the mean packet loss probability of link $\{i, j\}$, p_{ij} , due to transmission errors or link failures

Consider a set of concurrent video sessions, denoted as \mathcal{E} , sustained in this network. Each video session $\sigma \in \mathcal{E}$ has a source node z_σ and a destination node d_σ . For each session (i.e., a source-destination pair) $z_\sigma-d_\sigma$, there is a set of given paths, denoted by \mathcal{P}_σ . In practice, these paths can be precomputed by proactive routing protocols [9, 23], or discovered by a reactive routing protocol [83]. The total rate of a video stream, R_σ , originated at source node z_σ , is bounded by $\underline{R}_\sigma \leq R_\sigma \leq \overline{R}_\sigma$, $\sigma \in \mathcal{E}$, while the lower and upper bounds of R_σ are determined by the specific video encoder and the video sequence used by source node z_σ . This rate R_σ is to be allocated among all the paths in \mathcal{P}_σ .¹ Letting an element in the rate vector be R_σ^n , $n \in \mathcal{P}_\sigma$, the following conditions must be satisfied.

$$\sum_{n \in \mathcal{P}_\sigma} R_\sigma^n = R_\sigma, \quad \text{and} \quad R_\sigma^n > 0, \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E}. \quad (4.1)$$

This chapter follows the notation in Chapter 3. Additional notation used in the chapter is summarized in Table 4.1.

4.2.1 Link and Path Statistics

We derive link and path statistics in this section. These statistics will be used to compute the application layer video distortion in Section 4.2.2.

Load and Delay on a Link

As discussed in Section 3.2.1, the traffic on a link $\{i, j\}$ is an aggregate of traffic from different paths that traverse the link. To account for the potential packet loss at upstream

¹A path in \mathcal{P}_σ will be assigned a rate of zero if it is not selected. That is, path selection is made when the rate vectors are determined.

Table 4.1: Additional notation for Chapter 4

z_σ	Source node of session σ .
d_σ	Destination node of session σ .
\mathcal{P}_σ	Path set of session σ , from z_σ to d_σ .
\mathcal{P}_σ^n	A path in the set \mathcal{P}_σ , from z_σ to d_σ .
Δ_σ	Decoding deadline of session σ .
T_σ^n	End-to-end delay on path $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$.
T_σ	Average end-to-end delay for session σ .
p_σ^n	End-to-end loss rate of $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$.
R_σ^n	Rate of video session σ on path $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$.
R_σ	Rate of video session σ .
\bar{R}_σ	The maximum rate of video session σ .
\underline{R}_σ	The minimum rate of video session σ .

links, let $\bar{\mathcal{P}}_\sigma^{n,ij}$ denote the *upstream partial path* for path \mathcal{P}_σ^n up to link $\{i, j\}$, exclusively. If link $\{i, j\} \notin \mathcal{P}_\sigma^n$, we have $\bar{\mathcal{P}}_\sigma^{n,ij} = \emptyset$. Then, the average rate of the aggregate traffic on link $\{i, j\} \in \mathcal{L}$ is

$$\lambda_{ij} = \sum_{\sigma \in \mathcal{E}} \sum_{n \in \mathcal{P}_\sigma} \left[\prod_{\{l,m\} \in \bar{\mathcal{P}}_\sigma^{n,ij}} (1 - p_{lm}) \right] \cdot R_\sigma^n. \quad (4.2)$$

The utilization of link $\{i, j\}$ is $\rho_{ij} = \lambda_{ij}/c_{ij}$. For stability, we should have $\rho_{ij} \leq 1 - \epsilon$, for some stability tolerance $0 < \epsilon < 1$, for all $\{i, j\} \in \mathcal{L}$.

Similar to Section 3.2.1, we model each link $\{i, j\}$ as a general queueing system, with the input rate λ_{ij} (defined in (3.1)) and service capacity c_{ij} . Let $f_{ij}(y)$ be the probability density function for the queueing delay t_{ij} on link $\{i, j\}$. The delay on a link could then be modeled via an exponential distribution i.e.,

$$f_{ij}(y) = \alpha_{ij} \cdot e^{-\alpha_{ij}y}, \quad \text{for } y \geq 0. \quad (4.3)$$

Path Delay

Recall that for each session $\sigma \in \mathcal{E}$, there is a set of paths \mathcal{P}_σ between the source node z_σ and the destination node d_σ . Let T_σ^n denote the delay on path \mathcal{P}_σ^n and T_σ the weighted average

end-to-end delay among all the paths for Session σ , then

$$T_\sigma = \sum_{n \in \mathcal{P}_\sigma} \frac{R_\sigma^n}{R_\sigma} \cdot T_\sigma^n \quad (4.4)$$

Since the delay on path \mathcal{P}_σ^n consists of delays on all the links along the path, we have that

$$T_\sigma^n = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} t_{ij}.$$

For the end-to-end delay, we can apply the *Chernoff Bound* [19] to obtain a good approximation, similar to the problem in the previous chapter. Applying the Chernoff Bound and using the analysis in Chapter 3, the distribution of T_σ^n can be approximated as [30]:²

$$\Pr\{T_\sigma^n \geq \Delta_\sigma\} \approx \left(\frac{\exp\{-F_\sigma^n(s_{\sigma,n}^*)\}}{s_{\sigma,n}^* \delta_{\sigma,n}(s_{\sigma,n}^*) \sqrt{2\pi}} \right), \quad (4.5)$$

where $\delta_{\sigma,n}(s) = \sqrt{\frac{\partial^2 \log M_\sigma^n}{\partial s^2}}$.

End-to-End Loss Rate

Assuming that the packet loss processes on the links are independent, the end-to-end loss probability for the path $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$ can be approximated as

$$p_\sigma^n = 1 - \prod_{\{i,j\} \in \mathcal{P}_\sigma^n} (1 - p_{ij}), \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E}. \quad (4.6)$$

4.2.2 Video Distortion

As in Chapter 3, we use the rate distortion model developed by Stuhlmüller *et al.* [102], to describe the relationship between the bit rate and the achieved video distortion. Please refer to Section 3.2.2 for further information.

²The moment generating function of a heavy-tailed Weibull random variable does not exist (although all of its moments are well-defined). Therefore, the Chernoff Bound approach cannot be applied to delays having such distributions. However, the overdue probability can be computed by the *sub-exponential* property. For example, for an i.i.d. sequence of heavy-tailed Weibull random variables $\{X_1, \dots, X_n\}$, we have $\Pr[\sum_{k=1}^n X_k > x] \approx \Pr[\max_{\{1 \leq k \leq n\}} \{X_k\} > x] \approx n \cdot \Pr[X_1 > x]$ [43].

With our results on link and path statistics from Section 4.2.1, we have

$$\begin{aligned}
D_\sigma^e &= \underbrace{D_0 + \frac{\omega}{R_\sigma - R_0}}_{D_\sigma^{enc}} + \underbrace{\kappa(1 - p_\sigma)Pr(T_\sigma > \Delta_\sigma)}_{D_\sigma^{cg}} + \underbrace{\kappa p_\sigma}_{D_\sigma^{loss}} \\
&= D_0 + \frac{\omega}{R_\sigma - R_0} + \\
&\quad \kappa \sum_{n \in \mathcal{P}_\sigma} \frac{R_\sigma^n}{R_\sigma} \{p_\sigma^n + (1 - p_\sigma^n)Pr(T_\sigma^n > \Delta_\sigma)\}, \tag{4.7}
\end{aligned}$$

where D_0 , ω , R_0 , and κ are constants for a specific video codec and video sequence.

4.2.3 Mathematical Formulation

We are now ready to formulate the problem of optimal path selection and rate allocation for concurrent video sessions (OPT-PSRA), with the objective of minimizing application layer video distortion. Mathematically, Problem OPT-PSRA can be stated as in Eqs. (3.13)–(4.13) below.

In Problem OPT-PSRA, the objective function (4.8) is the sum of the average distortion of all the concurrent video sessions. The goal is to obtain the best possible rate vectors that would minimize (4.8) over a given set of paths for each video session. Alternative objective functions, such as $\max\{D_\sigma^e\}$ or an objective function in the form of a utility function $\sum_\sigma f(D_\sigma^e)$ (e.g., a logarithmic utility function), can be handled using the same solution procedure that will be discussed in Section 4.3.

The search space for Problem OPT-PSRA consists of two sets of continuous variables: (i) the set of rates for the video sessions $\{R_\sigma\}_{\sigma \in \mathcal{E}}$; and (ii) the set of rates for all paths in the path set for a given session $\{R_\sigma^n\}_{n \in \mathcal{P}_\sigma}$, for $\sigma \in \mathcal{E}$. Equation (4.9) provides the relationship between these two sets of variables. The inequalities in (4.10) bound the feasible rate for each video session, which is determined by the specific video sequence and the encoder parameters. The inequalities in (4.11) are the stability conditions, which ensure that the links are stable with finite delays. Equation (4.12) is derived from the definition of $\delta_{\sigma,n}$, while Eq. (4.13) is a

reformulation of Eq. (3.7) and is used to compute $s_{\sigma,n}^*$ for each path.

OPT-PSRA

Minimize

$$D = \sum_{\sigma \in \mathcal{E}} \left\{ D_0 + \frac{\omega}{R_\sigma - R_0} + \kappa \sum_{n \in \mathcal{P}_\sigma} \frac{R_\sigma^n}{R_\sigma} \left\{ p_\sigma^n + (1 - p_\sigma^n) \left\{ \frac{e^{-s_{\sigma,n}^* \Delta_\sigma}}{s_{\sigma,n}^* \delta_{\sigma,n}(s_{\sigma,n}^*) \sqrt{2\pi}} \right. \right. \right. \\ \left. \left. \left. \prod_{\{i,j\} \in \mathcal{P}_\sigma^n} \frac{c_{ij} - \sum_{\varphi \in \mathcal{E}} \sum_{k \in \mathcal{P}_\varphi} \left[\prod_{\{l,m\} \in \bar{\mathcal{P}}_\varphi^{k,ij} (1 - p_{lm}) \right] R_\varphi^k}{c_{ij} - \sum_{\varphi \in \mathcal{E}} \sum_{k \in \mathcal{P}_\varphi} \left[\prod_{\{l,m\} \in \bar{\mathcal{P}}_\varphi^{k,ij} (1 - p_{lm}) \right] R_\varphi^k - s_{\sigma,n}^*} \right] \right\} \right\} \quad (4.8)$$

subject to

$$R_\sigma = \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n, \quad \forall \sigma \in \mathcal{E} \quad (4.9)$$

$$\underline{R}_\sigma \leq R_\sigma \leq \bar{R}_\sigma, \quad \forall \sigma \in \mathcal{E} \quad (4.10)$$

$$\sum_{\sigma \in \mathcal{E}} \sum_{n \in \mathcal{P}_\sigma} \left[\prod_{\{l,m\} \in \bar{\mathcal{P}}_\sigma^{n,ij}} (1 - p_{lm}) \right] \cdot R_\sigma^n \leq (1 - \epsilon) \cdot c_{ij}, \quad \forall \{i,j\} \in \mathcal{P}_\sigma \quad (4.11)$$

$$\delta_{\sigma,n}^2 = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} \frac{1}{(c_{ij} - \sum_{\varphi \in \mathcal{E}} \sum_{k \in \mathcal{P}_\varphi} \left[\prod_{\{l,m\} \in \bar{\mathcal{P}}_\varphi^{k,ij} (1 - p_{lm}) \right] R_\varphi^k - s_{\sigma,n}^*)^2}, \quad (4.12) \\ \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E}$$

$$\Delta_\sigma = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} \frac{1}{c_{ij} - \sum_{\varphi \in \mathcal{E}} \sum_{k \in \mathcal{P}_\varphi} \left[\prod_{\{l,m\} \in \bar{\mathcal{P}}_\varphi^{k,ij} (1 - p_{lm}) \right] R_\varphi^k - s_{\sigma,n}^*}, \quad (4.13) \\ \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E}.$$

Observe that the objective function (4.8) and the constraints (4.12) and (4.13) are non-polynomial, non-convex functions of both $\{R_\sigma\}_{\sigma \in \mathcal{E}}$ and $\{R_\sigma^n\}_{n \in \mathcal{P}_\sigma}$. The rates of all the video sessions are closely coupled in (4.8). Since non-polynomial problems are NP-hard in general [35], and Problem OPT-PSRA does not appear to possess any simplifying special structure, it is likely also NP-hard. In the following section, we present a branch-and-bound solution procedure for Problem OPT-PSRA, predicated on the RLT construct. Our proposed solution procedure can produce a solution within a relative error of ϵ to the global optimum, where $0 < \epsilon < 1$ is an arbitrarily small value reflecting the tolerance in approximation.

4.3 Branch-and-bound Algorithm for Solving Problem OPT-PSRA

We can now design a specialized branch-and-bound algorithm to solve Problem OPT-PSRA, by embedding RLT relaxation at every node in the branch-and-bound tree, as described before in Section 2.3. But first, we shall reformulate the non-polynomial terms in Problem OPT-PSRA, so as to convert it to a polynomial problem p -PSRA. Then we apply RLT to the polynomial problem and derive the corresponding LP relaxation ℓ -PSRA. The optimal solution to this LP relaxation provides a lower bound LB , which is often infeasible to the original problem. Therefore, following the reformulation, we present a local search algorithm that generates a feasible solution by applying a rounding scheme to the LP solution. We present the detailed branch-and-bound algorithm in Figure 4.2.

4.3.1 Reformulation

Due to the existence of non-polynomial terms in Problem OPT-PSRA, our first goal is to reformulate this problem into a *polynomial programming problem*, which will simplify the objective function as well as the constraints.

In the objective function (4.8), there are three sets of non-polynomial terms. In order to transform the first two non-polynomial terms, we define new variables $u_\sigma = 1/(R_\sigma - R_0)$ and $w_\sigma = (1/R_\sigma) \cdot \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n \cdot p_\sigma^n$. Substituting u_σ and w_σ into the objective function, we get two linear terms $\omega \cdot u_\sigma$ and $\kappa \cdot w_\sigma$, respectively, and two sets of new polynomial constraints $u_\sigma \cdot (R_\sigma - R_0) = 1$ and $w_\sigma \cdot R_\sigma = \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n \cdot p_\sigma^n$.

In order to transform the third non-polynomial term in Eq. (4.8), i.e., the product of fractions, recall that α_{ij} denotes the available bandwidth on a link $\{i, j\}$, i.e.,

$$\alpha_{ij} = c_{ij} - \sum_{\sigma \in \mathcal{E}} \sum_{n \in \mathcal{P}_\sigma} \theta_{ij}^{\sigma, n} \cdot R_\sigma^n, \quad (4.14)$$

where $\theta_{ij}^{\sigma,n} = \prod_{\{l,m\} \in \bar{\mathcal{P}}_{\sigma}^{n,ij}} (1 - p_{lm})$. Note that $\theta_{ij}^{\sigma,n}$ is constant, since the paths are given.

Let g_{σ}^n denote the weighted packet overdue probability on the path \mathcal{P}_{σ}^n , i.e., $g_{\sigma}^n = \frac{R_{\sigma}^n}{R_{\sigma}} \cdot \Pr(T_{\sigma}^n > \Delta_{\sigma})$. Again, defining a substitution variable $v_{ij} = \frac{1}{\alpha_{ij} - s_{\sigma,n}^*}$ to convert the fractions in $\Pr(T_{\sigma}^n > \Delta_{\sigma})$ to polynomial form, i.e.,

$$g_{\sigma}^n = \frac{R_{\sigma}^n}{R_{\sigma}} \cdot \left[\frac{e^{-s_{\sigma,n}^* \Delta_{\sigma}}}{s_{\sigma,n}^* \delta_{\sigma,n} \sqrt{2\pi}} \prod_{\{i,j\} \in \mathcal{P}_{\sigma}^n} \alpha_{ij} \cdot v_{ij} \right]. \quad (4.15)$$

This product form motivates us to apply logarithms on both sides of Eq. (4.15), which will lead to a linear constraint. This way, the complexity on the objective function can be effectively moved into the constraints.

Taking logarithm on both sides of Eq. (4.15) and making the following substitutions,

$$\left\{ \begin{array}{l} \gamma_{\sigma,n}^g = \log(g_{\sigma}^n), \quad \forall \sigma \in \mathcal{E} \\ \gamma_{\sigma,n}^R = \log(R_{\sigma}^n), \quad \forall \sigma \in \mathcal{E} \\ \gamma_{\sigma}^R = \log(R_{\sigma}), \quad \forall \sigma \in \mathcal{E} \\ \gamma_{\sigma,n}^s = \log(s_{\sigma,n}^*), \quad \forall \sigma \in \mathcal{E} \\ \gamma_{\sigma,n}^{\delta} = \log(\delta_{\sigma,n}), \quad \forall \sigma \in \mathcal{E} \\ \gamma_{ij}^{\alpha} = \log(\alpha_{ij}), \quad \forall \{i,j\} \in \mathcal{P}_{\sigma}, \forall \sigma \in \mathcal{E} \\ \gamma_{ij}^v = \log(v_{ij}), \quad \forall \{i,j\} \in \mathcal{P}_{\sigma}, \forall \sigma \in \mathcal{E}, \end{array} \right. \quad (4.16)$$

we have,

$$\gamma_{\sigma,n}^g = \gamma_{\sigma,n}^R - \gamma_{\sigma}^R - s_{\sigma,n}^* \Delta_{\sigma} - \gamma_{\sigma,n}^s - \gamma_{\sigma,n}^{\delta} + \sum_{\{i,j\} \in \mathcal{P}_{\sigma}^n} (\gamma_{ij}^{\alpha} + \gamma_{ij}^v) - \log(\sqrt{2\pi}), \quad (4.17)$$

where throughout, $\log(\cdot)$ denotes logarithm to the base e .

Once the objective function is simplified, in order to generate a linear programming relaxation for the problem OPT-PSRA, we shall construct polyhedral outer approximations for each of these logarithmic identities as follows.

In generic notation, which can then be applied to each of the identities in (4.16), consider the following relationship:

$$y = \log(x), \quad \text{where } 0 < x_0 \leq x \leq 1 \quad (4.18)$$

We can linearize this logarithmic relationship over the bounds using a *polyhedral outer approximation* comprised of a *convex envelope* in concert with several tangential supports. For instance, if x is bounded as $0 < x_0 \leq x \leq 1$, these constraints can be written as follows.

$$\begin{cases} y \geq \frac{\log(x_0)}{1-x_0} \cdot (1-x) \\ y \leq \log(x_k) + \frac{x-x_k}{x_k}, \quad k = 1, \dots, k_{max}, \end{cases} \quad (4.19)$$

where $x_k = x_0 + (1-x_0) \cdot \frac{k-1}{k_{max}-1}$, for $k = 1, \dots, k_{max}$. A four-point tangential approximation can be obtained by letting $k_{max} = 4$, as illustrated in Figure 4.1. The corresponding convex envelope consists of a chord connecting the two end points, which is used in combination with four tangential supports at four points including the two end points. As a result, every logarithmic relationship specified in Eq. (4.16) translates to five linear constraints constituting a polyhedral outer approximation. Note that such polyhedral outer approximations will be iteratively tightened during the branch-and-bound procedure (see Section 2.3.2).

Now we have successfully reformulated the objective function into a linear form, and introduced the corresponding polynomial constraints to make the reformulation tight. To make the problem polynomial, it still remains to transform the two constraints (4.12) and (4.13) to the polynomial form. From the definition of v_{ij} , we have that $\delta_{\sigma,n}^2 = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} v_{ij}^2$ and $\Delta_\sigma = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} v_{ij}$. With the above re-formulation, we can now rewrite Problem OPT-PSRA as a *polynomial programming problem* (p -PSRA) as follows.

p -PSRA:

$$\text{Minimize } D(x_k) = \sum_{\sigma \in \mathcal{E}} \left\{ D_0 + \omega u_\sigma + \kappa w_\sigma + \kappa \sum_{n \in \mathcal{P}_\sigma} (1 - p_\sigma^n) g_\sigma^n \right\} \quad (4.20)$$

$$\text{subject to } R_\sigma = \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n, \quad \forall \sigma \in \mathcal{E} \quad (4.21)$$

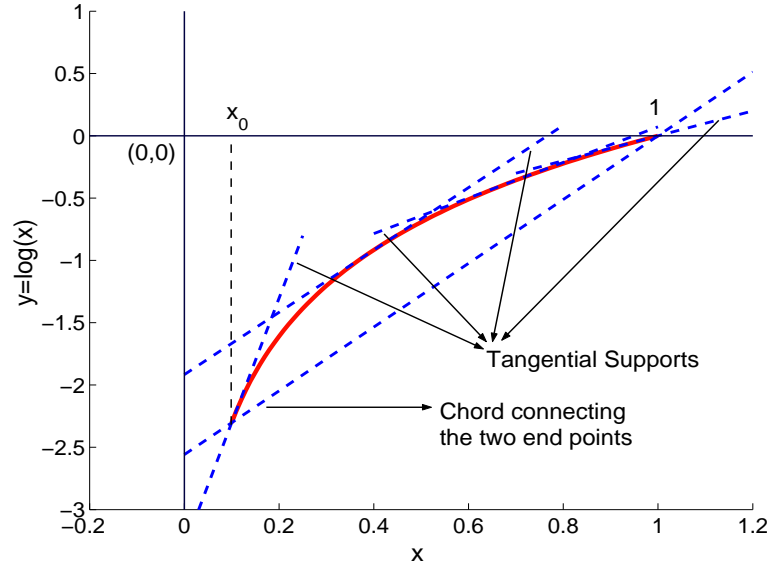


Figure 4.1: Polyhedral outer approximation for $y = \log(x)$ in $0 < x_0 \leq x \leq 1$.

$$\alpha_{ij} = c_{ij} - \sum_{\varphi \in \mathcal{E}} \sum_{k \in \mathcal{P}_\varphi} \theta_{ij}^{\varphi,k} \cdot R_\varphi^k, \quad \forall \{i, j\} \in \mathcal{P}_\sigma \quad (4.22)$$

$$\alpha_{ij} \geq \epsilon \cdot c_{ij}, \quad \forall \{i, j\} \in \mathcal{P}_\sigma \quad (4.23)$$

$$\Delta_\sigma = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} v_{ij}, \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \quad (4.24)$$

$$v_{ij} \cdot (\alpha_{ij} - s_{\sigma,n}^*) = 1, \quad \forall \{i, j\} \in \mathcal{P}_\sigma, \quad \forall \sigma \in \mathcal{E} \quad (4.25)$$

$$\delta_{\sigma,n}^2 = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} v_{ij}^2, \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \quad (4.26)$$

$$w_\sigma \cdot R_\sigma = \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n \cdot p_\sigma^n, \quad \forall \sigma \in \mathcal{E} \quad (4.27)$$

$$u_\sigma \cdot (R_\sigma - R_0) = 1, \quad \forall \sigma \in \mathcal{E} \quad (4.28)$$

$$\begin{aligned} \gamma_{\sigma,n}^g = & \gamma_{\sigma,n}^R - \gamma_\sigma^R - s_{\sigma,n}^* \Delta_\sigma - \gamma_{\sigma,n}^s - \gamma_{\sigma,n}^\delta + \\ & \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} (\gamma_{ij}^\alpha + \gamma_{ij}^v) - \log(\sqrt{2\pi}), \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \end{aligned} \quad (4.29)$$

Polyhedral outer approximations for $\gamma_{\sigma,n}^R, \gamma_\sigma^R, \gamma_{\sigma,n}^g,$

$$\gamma_{\sigma,n}^s, \gamma_{\sigma,n}^\delta, \gamma_{ij}^\alpha, \gamma_{ij}^v, \quad \forall \{i, j\} \in \mathcal{P}_\sigma^n, \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \quad (4.30)$$

$$\underline{R}_\sigma \leq R_\sigma \leq \overline{R}_\sigma, \quad \forall \sigma \in \mathcal{E} \quad (4.31)$$

$$\frac{1}{\overline{R}_\sigma - R_0} \leq u_\sigma \leq \frac{1}{\underline{R}_\sigma - R_0}, \quad \forall \sigma \in \mathcal{E} \quad (4.32)$$

Implied bounds for all other variables. (4.33)

In Problem p -PSRA, as in the case of the original problem, constraints (4.22) and (4.23) are stability constraints and constraints (4.31) are bounds on the video rates. Constraint (4.24) is a reformulation of constraint (4.13) of the original OPT-PSRA problem. Constraints (4.25)–(4.28) and (4.32) are derived from the definition of the corresponding variables. Constraint (4.29) results from a linearization of Eq. (4.15), and the constraints in (4.30) are the polyhedral outer approximations for the logarithms of the packet overdue probabilities on the paths $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$ (see Eqs. (4.15)–(4.19)).

As discussed before, Problem OPT-PSRA is now transformed into a polynomial nonlinear programming problem of order two. The highly complex objective function (4.8) is greatly simplified (i.e., linearized) and the complexity is shifted into the constraints in the polynomial form. Although this problem is simpler than the original problem, it is still a nonlinear programming problem, which is NP-hard in general.

4.3.2 Generating LP Relaxation Using RLT

Now that Problem OPT-PSRA is translated into the polynomial Problem p -PSRA, we employ RLT to generate the corresponding LP relaxation ℓ -PSRA. First, nonlinear implied constraints are generated by taking the products of bounding terms of the decision variables that constitute the polynomial terms in Problem p -PSRA. The resulting problem is subsequently linearized by variable substitutions, one for each nonlinear term appearing in the problem, including both the objective function and the constraints. It should be noted that the maximum order of the polynomials in Problem p -PSRA is two.

For instance, the second order term $u_\sigma \cdot R_\sigma$ in Eq. (4.28) can be viewed as a single term, for which we can introduce a new variable μ_σ , thereby substituting $\mu_\sigma = u_\sigma \cdot R_\sigma$. Since u_σ and R_σ are each bounded by $(u_\sigma)_L \leq u_\sigma \leq (u_\sigma)_U$ and $(R_\sigma)_L \leq R_\sigma \leq (R_\sigma)_U$, respectively, we generate the following relational constraints, which are known as *RLT bound-factor product*

constraints.

$$\begin{cases} \{[u_\sigma - (u_\sigma)_L] \cdot [R_\sigma - (R_\sigma)_L]\}_L \geq 0 \\ \{[u_\sigma - (u_\sigma)_L] \cdot [(R_\sigma)_U - R_\sigma]\}_L \geq 0 \\ \{[(u_\sigma)_U - u_\sigma] \cdot [R_\sigma - (R_\sigma)_L]\}_L \geq 0 \\ \{[(u_\sigma)_U - u_\sigma] \cdot [(R_\sigma)_U - R_\sigma]\}_L \geq 0, \end{cases} \quad (4.34)$$

where $\{\cdot\}_L$ denotes a linearization step under the substitution $\mu_\sigma = u_\sigma \cdot R_\sigma$. Note that $(R_\sigma)_L = \underline{R}_\sigma$ and $(R_\sigma)_U = \overline{R}_\sigma$. From the above relationships and by substituting $\mu_\sigma = u_\sigma \cdot R_\sigma$, we have the following RLT constraints for μ_σ .

$$\begin{cases} (u_\sigma)_L \cdot R_\sigma + (R_\sigma)_L \cdot u_\sigma - \mu_\sigma \leq (u_\sigma)_L \cdot (R_\sigma)_L \\ (u_\sigma)_U \cdot R_\sigma + (R_\sigma)_L \cdot u_\sigma - \mu_\sigma \geq (u_\sigma)_U \cdot (R_\sigma)_L \\ (u_\sigma)_L \cdot R_\sigma + (R_\sigma)_U \cdot u_\sigma - \mu_\sigma \geq (u_\sigma)_L \cdot (R_\sigma)_U \\ (u_\sigma)_U \cdot R_\sigma + (R_\sigma)_U \cdot u_\sigma - \mu_\sigma \leq (u_\sigma)_U \cdot (R_\sigma)_U. \end{cases} \quad (4.35)$$

We therefore replace the second-order term $u_\sigma \cdot R_\sigma$ with the linear term μ_σ in Eq. (4.28) and introduce the above linear bound-factor RLT constraints for μ_σ into the Problem p -PSRA formulation. Similarly, we define new variables for all the remaining nonlinear terms in Problem p -PSRA, including $\xi_{ij} = v_{ij} \cdot \alpha_{ij}$, $\psi_{ij} = v_{ij} \cdot s_{\sigma,n}^*$, $\phi_{ij} = v_{ij}^2$, $\chi_{\sigma,n} = \delta_{\sigma,n}^2$, and $\eta_\sigma = w_\sigma \cdot R_\sigma$, and make substitutions in the same manner.

Let \mathbf{R} and \mathbf{R}^n be vectors having components R_σ and R_σ^n , respectively. After replacing all non-linear terms as above and adding the corresponding RLT constraints into the Problem p -PSRA formulation, we obtain the following *linear programming relaxation* problem (ℓ -PSRA) as described in Eqs. (4.36)–(4.50), for which many efficient (polynomial-time) solution techniques and tools are available.

ℓ -PSRA:

$$\text{Minimize } D(x_k) = \sum_{\sigma \in \mathcal{E}} \left\{ D_0 + \omega u_\sigma + \kappa w_\sigma + \kappa \sum_{n \in \mathcal{P}_\sigma} (1 - p_\sigma^n) g_\sigma^n \right\} \quad (4.36)$$

$$\text{subject to } R_\sigma = \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n, \quad \forall \sigma \in \mathcal{E} \quad (4.37)$$

$$\alpha_{ij} = c_{ij} - \sum_{\varphi \in \mathcal{E}} \sum_{k \in \mathcal{P}_\varphi} \theta_{\varphi,ij} \cdot R_\varphi, \quad \forall \{i,j\} \in \mathcal{P}_\sigma \quad (4.38)$$

$$\alpha_{ij} \geq \epsilon \cdot c_{ij}, \quad \forall \{i,j\} \in \mathcal{P}_\sigma \quad (4.39)$$

$$\Delta_\sigma = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} v_{ij}, \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \quad (4.40)$$

$$\xi_{ij} - \psi_{ij} = 1, \quad \forall \{i,j\} \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \quad (4.41)$$

$$\chi_{\sigma,n} = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} \phi_{ij}, \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \quad (4.42)$$

$$\eta_\sigma = \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n \cdot p_\sigma^n, \quad \forall \sigma \in \mathcal{E} \quad (4.43)$$

$$\mu_\sigma - R_0 \cdot u_\sigma = 1, \quad \forall \sigma \in \mathcal{E} \quad (4.44)$$

$$\begin{aligned} \gamma_{\sigma,n}^g &= \gamma_{\sigma,n}^R - \gamma_\sigma^R - s_{\sigma,n}^* \Delta_\sigma - \gamma_{\sigma,n}^s - \gamma_{\sigma,n}^\delta + \\ &\sum_{\{i,j\} \in \mathcal{P}_\sigma^n} (\gamma_{ij}^\alpha + \gamma_{ij}^v) - \log(\sqrt{2\pi}), \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \end{aligned} \quad (4.45)$$

Polyhedral outer approximations for $\gamma_{\sigma,n}^R, \gamma_\sigma^R, \gamma_{\sigma,n}^g,$

$$\gamma_{\sigma,n}^s, \gamma_{\sigma,n}^\delta, \gamma_{ij}^\alpha, \gamma_{ij}^v, \quad \forall \{i,j\} \in \mathcal{P}_\sigma^n, \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{E} \quad (4.46)$$

$$\underline{R}_\sigma \leq R_\sigma \leq \overline{R}_\sigma, \quad \forall \sigma \in \mathcal{E} \quad (4.47)$$

$$\frac{1}{\overline{R}_\sigma - R_0} \leq u_\sigma \leq \frac{1}{\underline{R}_\sigma - R_0}, \quad \forall \sigma \in \mathcal{E} \quad (4.48)$$

Bound-factor product RLT constraints for the terms

$$\begin{aligned} \mu_\sigma &= u_\sigma \cdot R_\sigma, \xi_{ij} = v_{ij} \cdot \alpha_{ij}, \psi_{ij} = v_{ij} \cdot s_{\sigma,n}^*, \\ \phi_{ij} &= v_{ij}^2, \chi_{\sigma,n} = \delta_{\sigma,n}^2, \text{ and } \eta_\sigma = w_\sigma \cdot R_\sigma \end{aligned} \quad (4.49)$$

$$\text{Implied bounds for all other variables.} \quad (4.50)$$

4.3.3 Local Search Algorithm

As discussed in Section 2.3.2, in the branch-and-bound procedure, the solution to the relaxation problem is usually infeasible to the original problem. This problem can be resolved by finding a feasible solution to the original problem via a local search algorithm that starts from the infeasible solution.

For Problem OPT-PSRA, we adopt the following local search strategy that computes a feasible solution $(\mathbf{R}, \mathbf{u}, \mathbf{v}, \alpha, \delta, \mathbf{s})$ from the solution to the relaxation problem $(\hat{\mathbf{R}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\alpha}, \hat{\delta}, \hat{\mathbf{s}})$. Specifically, since the rates of the video sessions obtained from the solution to the relaxation problem are always feasible to the original problem (i.e., the stability constraints are always satisfied and the rates are always within the lower and upper bounds), we have that $\mathbf{R} = \hat{\mathbf{R}}$. From \mathbf{R} , we can compute the values of α_{ij} from Eq. (4.14), the values of $s_{\sigma,n}^*$ from Eq. (3.7), and the values of u_σ from Eq. (4.28). After obtaining α_{ij} and $s_{\sigma,n}^*$, we can compute v_{ij} from Eq. (4.25), and $\delta_{\sigma,n}$ from Eq. (4.26). Therefore, a feasible solution to the original Problem OPT-PSRA, $(\mathbf{R}, \mathbf{u}, \mathbf{v}, \alpha, \delta, \mathbf{s})$, can be obtained from the solution to the relaxed problem. The complete branch-and-bound algorithm for solving Problem OPT-PSRA presented in Figure 4.2.

4.3.4 Remarks

When optimizing the performance of rate allocation schemes for multiple users, efficiency and fairness are usually *orthogonal* objectives, i.e., maximizing one may lead to significant decreases in the other. In Problem OPT-PSRA, we minimize the sum of the average distortion of all the concurrent video sessions (see Eq. (4.8)), thus achieving the best utilization of network resources, which are limited in ad hoc networks. Alternatively, we can use an objective function $\max_{\sigma \in \mathcal{E}} \{D_\sigma^e\}$. Minimizing this objective function will equalize the performance of all the users as much as possible. To solve this new problem, we can make a simple transformation by defining a new variable as $y = \max_{\sigma \in \mathcal{E}} \{D_\sigma^e\}$. The transformed problem minimizes y , with additional constraint $D_\sigma^e \leq y$ for all $\sigma \in \mathcal{E}$. Then, our branch-and-bound/RLT-based solution procedure can be applied to this transformed problem to obtain ϵ -optimal solutions.

Furthermore, it has been shown that proportional fairness could achieve a trade-off between efficiency and fairness. For this purpose, a logarithmic utility function could be used for the users, i.e., minimizing $\sum_{\sigma \in \mathcal{E}} \log(D_\sigma^e)$. Similarly, we can first make a transformation

Path Selection and Rate Allocation Algorithm

1. Initialization:
2. Initialize the best solution $\psi^* = \emptyset$ and the best upper bound $UB = \infty$.
3. Initialize the problem list L with the original problem, Problem 1
4. Relaxation:
5. Solve the RLT relaxation ℓ -PSRA(Ω_1) for Problem 1
6. Denote the obtained relaxation solution as $(\hat{\mathbf{R}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\alpha}, \hat{\delta}, \hat{\mathbf{s}})$ and the objective value as the lower bound LB_1 .
7. Let the initial worst lower bound $LB = LB_1$.
8. Iteration:
9. Select problem k that has the minimum LB_k among all problems in the problem list.
10. Local Search:
11. Obtain a feasible solution $(\mathbf{R}, \mathbf{u}, \mathbf{v}, \alpha, \delta, \mathbf{s})$ from $(\hat{\mathbf{R}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\alpha}, \hat{\delta}, \hat{\mathbf{s}})$ by running the local search algorithm.
12. Denote the objective value obtained by using $(\mathbf{R}, \mathbf{u}, \mathbf{v}, \alpha, \delta, \mathbf{s})$ in the original problem as UB_k .
13. If $(UB_k < UB)$ {
14. Update $\psi^* = \psi$ and $UB = UB_k$.
15. If $(LB \geq (1 - \epsilon)UB)$, stop with the ϵ -optimal solution ψ^* .
16. Otherwise, remove all problems k' from the problem list that satisfy $(1 - \epsilon)UB \leq LB_{k'}$. }
17. Partition:
18. Find the maximum relaxation error among all the RLT variables such as $|\hat{u}_\sigma \hat{R}_\sigma - \hat{\mu}_\sigma|$
19. In the case that the maximum relaxation error is $|\hat{u}_\sigma \hat{R}_\sigma - \hat{\mu}_\sigma|$,
20. if $((u_\sigma)_U - (u_\sigma)_L) \cdot \min\{\hat{u}_\sigma - (u_\sigma)_L, (u_\sigma)_U - \hat{u}_\sigma\} \geq ((R_\sigma)_U - (R_\sigma)_L) \cdot$
21. $\min\{\hat{R}_\sigma - (R_\sigma)_L, (R_\sigma)_U - \hat{R}_\sigma\}$,
22. partition Ω_k into two new regions Ω_{k_1} and Ω_{k_2} by dividing $[(u_\sigma)_L, (u_\sigma)_U]$ into $[(u_\sigma)_L, \hat{u}_\sigma]$ and
23. $[\hat{u}_\sigma, (u_\sigma)_U]$;
24. Else, partition Ω_k into two new regions by dividing $[(R_\sigma)_L, (R_\sigma)_U]$ into $[(R_\sigma)_L, \hat{R}_\sigma]$ and $[\hat{R}_\sigma, (R_\sigma)_U]$.
25. Perform similar partitions if the maximum relaxation error is obtained from any other RLT variable
26. and its associated product.
27. Bounding Step:
28. Solve the RLT relaxation for the two sub-problems and obtain their lower bounds LB_{k_1} and LB_{k_2} .
29. Remove problem k from the problem list.
30. If $(1 - \epsilon)UB > LB_{k_1}$, add problem k_1 into the problem list.
31. If $(1 - \epsilon)UB > LB_{k_2}$, add problem k_2 into the problem list.
32. If the problem list is empty, stop with the ϵ -optimal solution ψ^* .
33. Otherwise, proceed to the next iteration.

Figure 4.2: Branch-and-bound and RLT based algorithm for Problem OPT-PSRA.

by defining new variables $y_\sigma = \log(D_\sigma^e)$, for all $\sigma \in \mathcal{E}$. The transformed problem minimizes $\sum_{\sigma \in \mathcal{E}} y_\sigma$, with additional constraints $y_\sigma = \log(D_\sigma^e)$, for all $\sigma \in \mathcal{E}$. Then, the polyhedral outer approximations could be applied for these logarithmic constraints (see Section 4.3.1) and the branch-and-bound/RLT-based solution procedure can be applied to solve the transformed problem.

4.4 Simulation Studies

In this section, we present simulation results for the optimal path selection and rate allocation problem. In each simulation, a wireless ad hoc network is generated by placing a number of nodes at random locations in a rectangular region. A wireless link exists if a node is within the radio range of a transmitting node. As discussed, a set of preselected paths are precomputed using a k -disjoint path routing algorithm for each source-destination pair, which are randomly chosen from the set of nodes \mathcal{N} . In order to show how the technique works and for simplicity, we assume that there is only one path connecting a source z_σ to destination d_σ , in the earlier part of this section. In the later part of the section, we expand the problem to multiple paths connecting each source destination pair.

In the simulations, each video session has a rate bounded by 20 Kb/s and 200 Kb/s. We used an H.263+ codec and the first 200 frame of the “Foreman” trace in the quarter common intermediate format (QCIF). The video was encoded at 12.5 frames per second and an intra rate of 1/9. Each group of blocks (GOB) was transmitted in a packet to make them independently decodable. The rate-distortion parameters are obtained from [102], and are found to be $D_0 = -0.3971$, $R_0 = 7.0307$, $\omega = 3448.4$, and $\kappa = 645.16$. Failure probabilities of the wireless links are chosen from a uniform distribution between [1%, 5%]; the bandwidth of a link is chosen from a uniform distribution between [50 Kb/s, 400 Kb/s]. For all the results reported in this section, the exponential model (4.3) and the Chernoff Bound approximation (4.5) are used to compute the end-to-end delay distribution. The proposed solution procedure

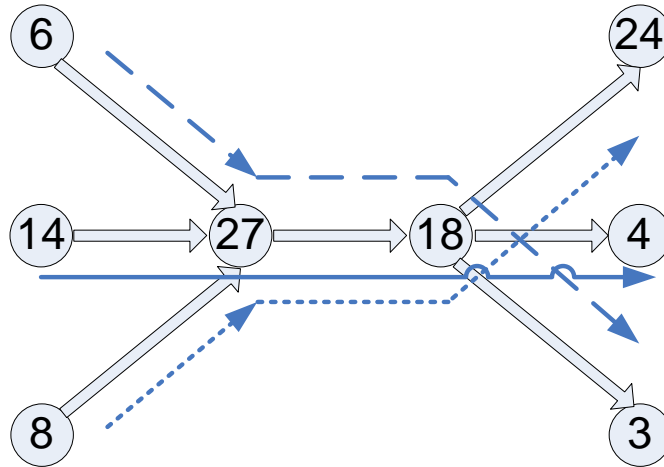
is implemented in C, and the LINDO API 3.0 is used for solving the LP relaxation Problem ℓ -PSRA. At every node in the branch-and-bound tree, the local search algorithm discussed in Section 4.3.3 is used to obtain a feasible solution from the LP relaxation solution.

4.4.1 Performance for Various Instances

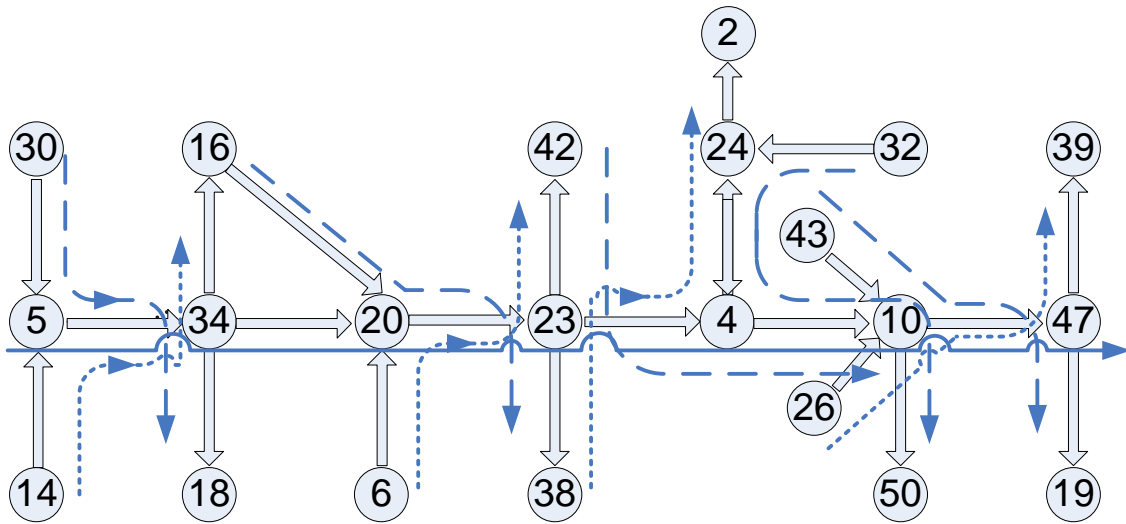
We first examine the performance of the proposed solution procedure with different instances of Problem OPT-PSRA, which are presented in Table 4.2. In the table, Cases I–III are for a 30-node network, as shown in Figure 4.4(a). There are two, three, and four sessions, respectively, and each source-destination pair is connected by one path, sharing a common bottleneck link (i.e., link $\{27, 18\}$). In Figure 4.3(a), we show the correlation among the given paths for Case II. Cases IV–VII are for a 50-Node network as shown in Figure 4.4(b). There are four, six, eight, and 10 sessions, respectively, and one path for each session. Similarly, Figure 4.3(b) depicts the path correlations for Case VII.

In these case studies, we examine the performance of the proposed solution procedure in the presence of multiple shared links and bottlenecks. The remaining Cases VIII–X are for the same 50-node network as shown in Figure 4.4(b), with three, four, and five sessions, respectively, and two paths for each session. In these cases studies, the proposed algorithm performs both rate allocation and path selection (i.e., proportional routing). The decoding deadline is 0.2 s for all the cases in Table 4.2.

The last column in Table 4.2 presents the number of subproblems examined, or nodes in the corresponding branch-and-bound tree, when the algorithm terminates. We observe that the number of nodes that are examined is an increasing function of the optimality gap ϵ , as well as the size of the problem. For example, for the same 50-node network, when the number of concurrent video sessions increases from 6 to 10, the number of nodes in the branch-and-bound tree increases from 112 to 334. The fifth column of Table 4.2 presents the total distortion values found by solving the corresponding UB for Problem 1 in the Path Selection and Rate Allocation Algorithm (i.e., the UB obtained from the first node in the

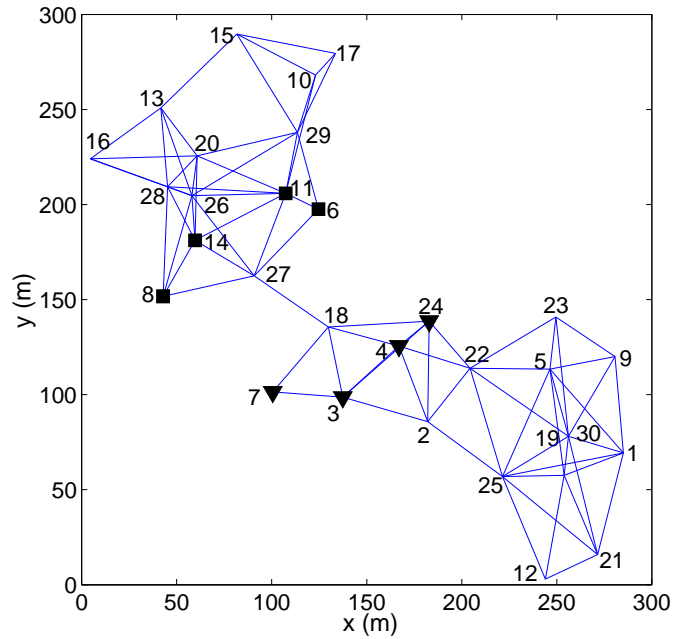


(a) 30-node network (Case II)

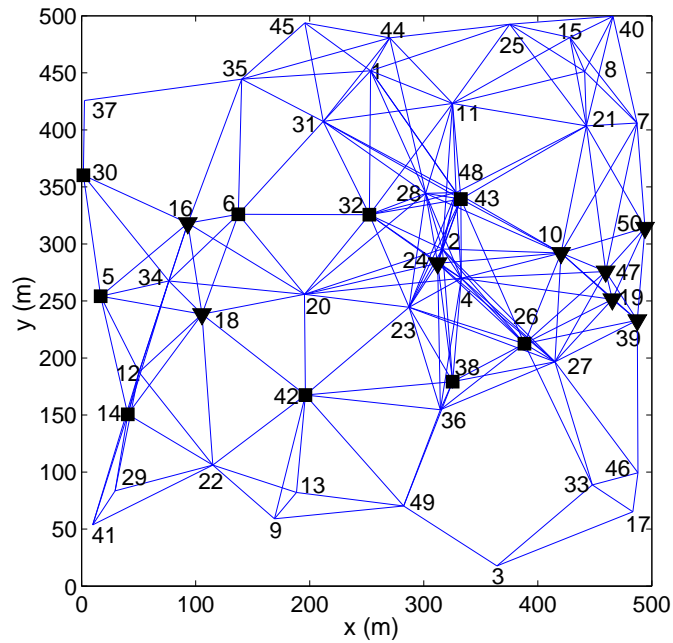


(b) 50-node network (Cases VII)

Figure 4.3: Correlation among paths in a network.



(a) A 30-node network for Cases I–III



(b) A 50-node network for Cases IV–X

Figure 4.4: Randomly generated networks. Source nodes are marked with solid square; destination nodes are marked with solid triangle.

Table 4.2: Performance of the proposed algorithm for various instances of Problem OPT-PSRA

Case	Network Size	No. of Sessions	No. of Paths	Init. Feas. Solution	ϵ -optimal Solution	ϵ	No. of Nodes
I	30	2	1	106.64	106.64	0.05	3
II	30	3	1	181.85	179.26	0.05	23
III	30	4	1	249.09	240.66	0.05	25
IV	50	4	1	255.59	252.13	0.05	16
V	50	6	1	419.44	366.41	0.1	112
VI	50	8	1	537.65	500.33	0.1	145
VII	50	10	1	665.39	640.10	0.1	334
VIII	50	3	2	200.45	190.67	0.1	48
IX	50	4	2	295.01	282.60	0.1	144
X	50	5	2	369.19	301.53	0.1	403

branch-and-bound tree); the sixth column of Table 4.2 presents the ϵ -optimal solution found by the algorithm. We find that the corresponding values between these two columns are very close to each other. This clearly demonstrates that the polyhedral outer approximation and the RLT-based LP relaxations used in the solution procedure are well designed and tight. This also implies that for time-critical applications, the rate vectors computed by Problem 1 can be used as a competitive near-optimal approximation, while the refined ϵ -optimal rate vectors could be updated when the branch-and-bound algorithm terminates.

We also study the convergence performance of the proposed solution procedure, which is illustrated in Figure 4.5. In this figure, we plot the evolutions of the upper bound UB and the lower bound LB for the 50-node network (see Figure 4.4(b)) with six concurrent video sessions. The decoding deadline is 0.1 s for this result. Recall that at each iteration, the lower and upper bounds for the original problem are chosen according to Eq. (2.4). In Figure 4.5, the y -axis represents the average value of Distortion for each session, while the x -axis represents the number of iterations. We observe that even though the gap between the lower and upper bounds is initially high, the gap quickly decreases over iterations. Also,

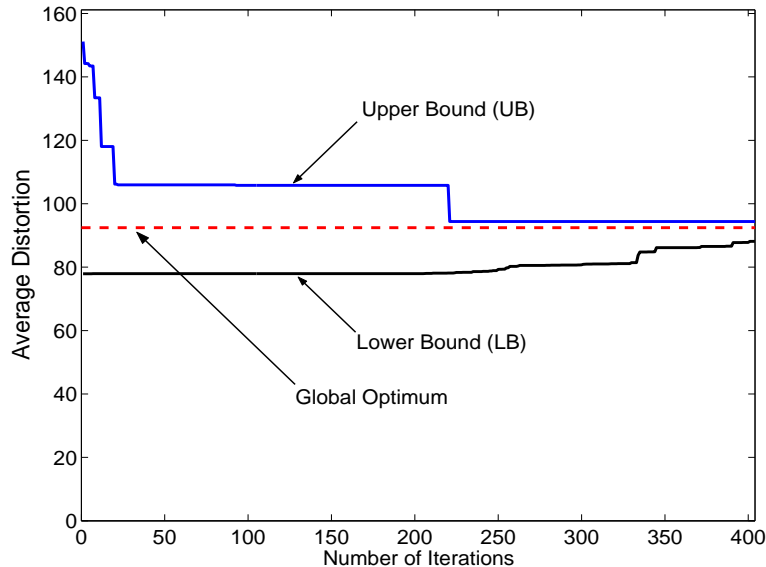


Figure 4.5: Convergence of path selection and rate allocation algorithm for a 50-node network with six video sessions.

the largest decreases in UB , i.e., the objective value of the feasible solution to Problem OPT-PSRA, occur in the first few iterations. When LB is within $\epsilon = 10\%$ of UB , the algorithm terminates. The global optimum, found by an exhaustive search, is also plotted in Figure 4.5, which is bounded by the upper and lower bounds, as expected, and is very close to the computed upper bound UB .

4.4.2 Comparison with A Network-centric Scheme: The Single Path Case

In the remainder of this section, we compare the performance of the proposed approach with a network-centric rate allocation scheme. Before presenting the comparison results for the multi-path case, we first compare the proposed algorithm with the network-centric scheme while assuming a single path for each session. The reason for doing this is two-fold. *First*, this allows us to separate rate allocation and path selection, and to focus on the rate allocation performance, since all the traffic for a session will be transmitted on the single path. We can

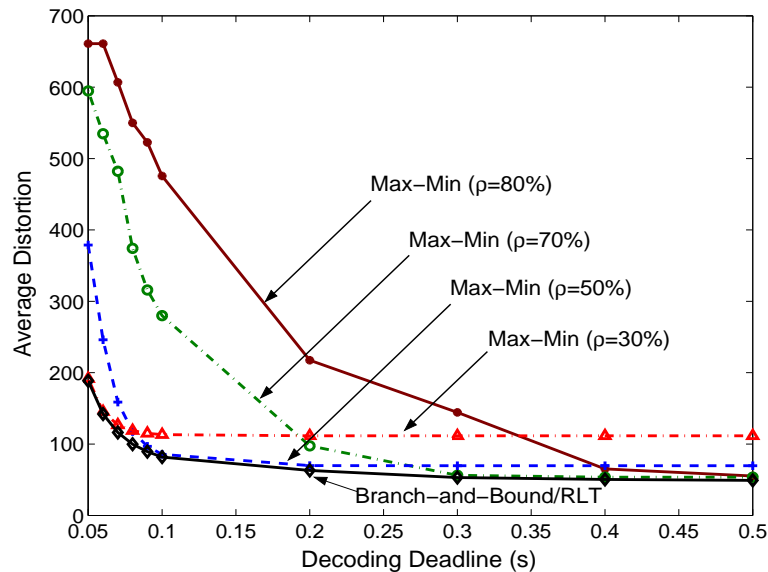


Figure 4.6: Average distortion versus decoding deadline for a 50-node network with 10 sessions and one path per session.

then examine the performance of the proposed algorithm on rate allocation only. *Second*, in an ad hoc network with a random topology, such single path scenarios are highly likely to exist, especially when mobility is allowed.

Specifically, we implement a *max-min* rate allocation scheme (called max-min throughout this chapter), which is widely regarded as a fair rate allocation policy [13]. In max-min, fairness is achieved by maximizing the minimum rate allocation in the network without exceeding its upper bound and the capacity of each link. Note that the max-min rate allocation is based on the fluid flow model and does not explicitly consider queueing delay at the links. Therefore, we need to compute the max-min fair rate allocations for a prescribed link utilization factor, which specifies the maximum percentage of capacity that can be used on a given link. Using this scheme, we find the rate allocation for each video session and then compute the total distortion using Eq. (4.8).

Figure 4.6 plots the average distortions found by the proposed algorithm and max-min for various decoding deadlines. The network consists of 50 nodes with 10 concurrent video

sessions. There is one path for each session. The max-min scheme is executed for various link utilization factors, ranging from 30% to 80%. It can be seen that the proposed approach outperforms max-min in all the cases. In fact, the branch-and-bound/RLT curve forms a *lower bounding envelope* for the set of curves produced by max-min. A closer examination reveals that for small decoding deadlines, max-min curves with lower link utilizations (e.g., $\rho=30\%$) are closer to the branch-and-bound/RLT curve, while for large decoding deadlines, max-min curves with higher link utilizations (e.g., $\rho=80\%$) are closer to the branch-and-bound/RLT curve.

This interesting observation can be well explained by Eq. (4.7). Recall that the end-to-end distortion D_σ^e consists of three components: $D_\sigma^e = D_\sigma^{enc} + D_\sigma^{cg} + D_\sigma^{loss}$. In this case, since there is only a single path associated with each session, D_σ^{loss} is the same for all the schemes. However, each of the remaining two terms will dominate in different ranges of decoding deadline. For small decoding deadlines, D_σ^{cg} is the dominating component. Thus max-min with a high link utilization suffers from severe congestion. For large decoding deadlines, D_σ^{enc} is the dominating component. Thus max-min with a low link utilization suffers from high encoding distortion. However, for the entire range of decoding deadline, the branch-and-bound/RLT algorithm can intelligently choose the ϵ -optimal rates to minimize both types of distortion. In addition, although the max-min distortion is close to that computed by the proposed algorithm for some decoding deadline values and link utilizations in this simple network, its performance will be much worse when there exist multiple paths for each session with diverse loss rates, i.e., when the third component, D_σ^{loss} , comes into play, as will be shown in Section 4.4.3.

We find that for very small decoding deadlines, the delay requirements are so stringent that all the schemes yield high distortion. On the other hand, for very larger decoding deadlines, congestion has only limited impact since only few packets are overdue. In this case, all the schemes can achieve a low total distortion that is determined by the end-to-end loss rates on the paths. The most interesting region, however, lies between these two regions, where a well designed rate allocation scheme can achieve a much better performance

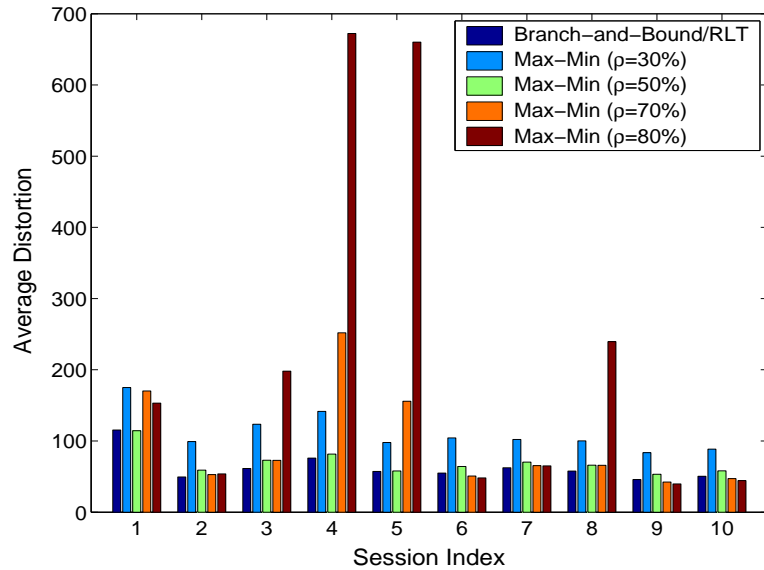


Figure 4.7: Distortion for individual video sessions in the 50-node network with 10 sessions and one path per session ($\Delta_\sigma=0.2$ s, for all $\sigma \in \mathcal{E}$).

by finding optimal rates for the video sessions. Within this region, and throughout this graph, we can see that the proposed approach consistently outperforms max-min by a significant margin. This is due to the fact that the proposed approach takes directly minimizes the distortion of the sessions. In Figure 4.5, the distortion achieved by the RLT-based procedure quickly decreases as the decoding deadline increases, while the distortion achieved by max-min stays persistently high for the small and medium ranges of decoding deadlines (implying that most video packets are overdue in these cases).

In Figure 4.7, the distortions of individual sessions computed by the proposed approach and max-min are plotted for a decoding deadline of 0.2 s. For each session index, distortion values are plotted in the following order: branch-and-bound/RLT, max-min ($\rho=30\%$), max-min ($\rho=50\%$), max-min ($\rho=70\%$), and max-min ($\rho=80\%$). We find that our algorithm achieves lower distortion than max-min for most of the sessions, except for Session Six, where the differences are negligible. The total distortion achieved by the proposed approach is 630.90. This is much lower than those achieved by max-min, which are 1115.80, 697.82,

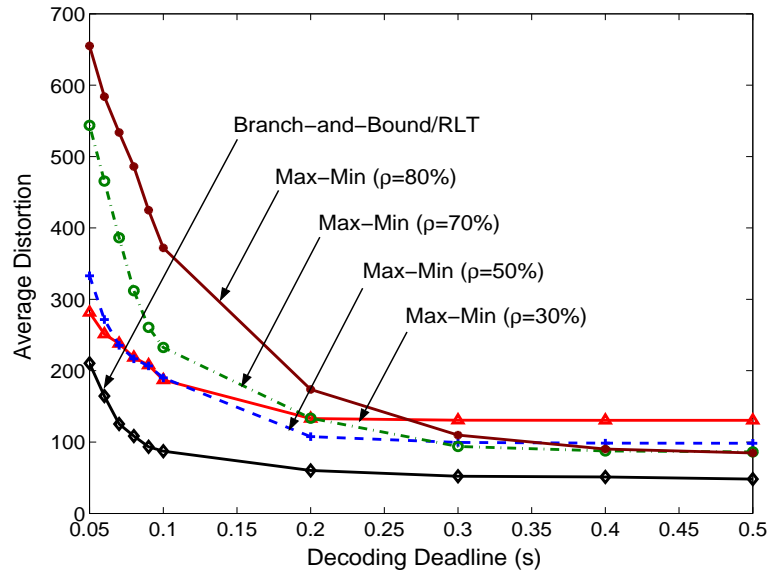


Figure 4.8: Average distortion versus decoding deadline for a 50-node network with five sessions and two paths per session.

975.01, and 2174.20 for link utilizations of 30%, 50%, 70%, and 80%, respectively.

4.4.3 Comparison with A Network-centric Scheme: The Multi-path Case

In the previous simulation studies, we examined the performance of the proposed algorithm under the condition that there is one path for each session. These will demonstrate its performance on rate allocation, i.e., how to reduce the encoding distortion without causing congestion in the network. In this section, we examine the multi-path cases with both rate allocation and path selection, where all the three components in D_g^e come into play.

In Figure 4.8, we plot the average distortion for various decoding deadlines for a 50-node network, where there are five sessions and two paths available for each session. As in Figure 4.6, we find that the average distortion quickly decreases when the delay constraint is relaxed. In contrast to Figure 4.6, the branch-and-bound/RLT approach outperforms

the max-min approach by a significant margin: there is a big gap between the branch-and-bound/RLT curve and the lower bounding envelope of the max-min curves. For example, when the decoding deadline is 0.2 s, the average distortion achieved by our approach is 60.31, which translates to a Peak-Signal-Noise-Ratio (PSNR) of 30.33 dB (computed as $10 \cdot \log_{10}(255 * 255/D_{\sigma}^e)$); the average distortion achieved by max-min when $\rho=50\%$ is 107.61, which translates to a PSNR of 27.81 dB (the best among all the max-min curves). There is a 78.4% reduction in distortion and a 2.52 dB improvement in average PSNR. Note that such an improvement is significant in terms of perceived video quality, since usually a half dB difference in PSNR is noticeable.

In order to illustrate the quality for individual sessions, we plot the distortion for each session obtained for a decoding deadline of 0.2 s in Figure 4.9. As in the single path case, for each session index, distortion values are plotted in the following order: branch-and-bound/RLT, max-min ($\rho=30\%$), max-min ($\rho=50\%$), max-min ($\rho=70\%$), and max-min ($\rho=80\%$). We observe that the proposed scheme achieves better performance for all the five sessions. The total distortion achieved by the proposed approach is 301.54. This is also much lower than those achieved by max-min, which are 663.33, 529.55, 615.43, and 824.16 for link utilizations of 30%, 50%, 70%, and 80%, respectively.

We then simulate the video transmissions in this 50-node network for the $\Delta_{\sigma} = 0.2$ s case. In the simulations, the source nodes transmit packetized video traffic along the paths according to the rate vectors computed, while the destination nodes reconstruct video frames from received packets and compute their PSNR values. For Session Two, the average PSNR is 32.27 dB for the branch-and-bound/RLT scheme, and are 28.73 dB, 30.20 dB, 29.62 dB, and 26.88 dB for max-min when link utilization is 30%, 50%, 70%, and 80%, respectively. The proposed approach achieves a 2.07 dB gain in average PSNR over max-min with $\rho=50\%$, which is the best among all the max-min schemes. In Figure 4.10, the PSNRs for the reconstructed frames are plotted for our scheme and max-min with $\rho=50\%$. We find that except for a few frames, there is a clear gap between the two PSNR curves.

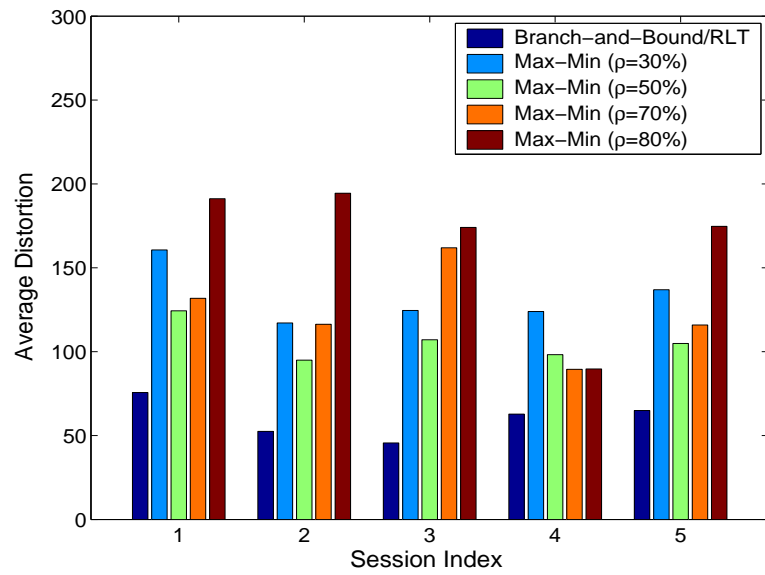


Figure 4.9: Distortion for individual video sessions in the 50-node network with five sessions and two paths for each session ($\Delta_\sigma=0.2$ s, for all $\sigma \in \mathcal{E}$).



Figure 4.10: PSNR of reconstructed video frames for video session Two.

Although PSNR provides a good means of quantifying video quality, perceived quality would be the ultimate performance measure for applications such as video. In order to illustrate the perceived quality of reconstructed video, in Figure 4.11, we present Frame 160 obtained by various schemes. Again, we find that although the frame delivered by max-min with $\rho=50\%$ is the best among all the max-min frames, it is still inferior to the frame obtained by our algorithm.

4.5 Related Work

The optimal traffic proportioning problem, i.e., given a set of calls, how to partition them to the set of given paths such that the overall throughput is maximized, has been studied extensively in the context of telephone networks. See [90] and references therein. Recently, a localized approach to the proportional routing problem is proposed in [75]. In addition, the problem of utility maximization with multi-path routing has been studied in the context of flow/congestion control for elastic data in a few recent works [45, 54, 63, 109]. In such problems, each session is associated with a concave, univariate utility function of its total rate. Distributed algorithms are developed to adapt the session rates, in order to maximize the total utility of all the sessions while subject to network resource constraints [63].

These interesting works motivate our efforts on this important problem, particularly when video quality is the optimization objective. The problem we study in this chapter differs from those analyzed by prior works in that we need information on link metrics and on all the active video sessions. The formulation is considerably more complex in that we model the end-to-end delay distributions, which is required for realtime traffic with tight decoding deadlines. There is no such nice property as convexity. A session's distortion is affected by other sessions in the network, making it impossible to break the original problem into simpler subproblems or to derive an easier-to-solve dual problem. Consequently, the prior approaches (developed for elastic data traffic based on flow models) could not be applied to



(a) Original



(b) RLT approach

(c) max-min ($\rho=30\%$)(d) max-min ($\rho=50\%$)(e) max-min ($\rho=70\%$)(f) max-min ($\rho=80\%$)

Figure 4.11: Reconstructed frame 160 from the “foreman” sequence for video session Two.

our problem.

As discussed, we focus on the joint design and optimization of the application and network layers, which differs from network-centric QoS routing problems for ad hoc networks. For example, see [8, 9, 18, 83], among others. Most of these efforts do not explicitly consider the optimization of application layer performance. Although the network layer performance could be optimized, they may not yield optimal performance at the application layer, since application performance metrics, e.g., video distortion, are usually highly complex functions of *multiple* network layer metrics (e.g., see Eq. (4.8)). Indeed, our proposed scheme is complementary to these works: the network-centric protocols could be used to compute a set of paths for a new video session, and then, the path selection and rate allocation algorithm could be applied to determine the ϵ -optimal rate vector for the video session.

Several path/server selection schemes have been developed for video communications in various network settings [6, 12, 67]. In [6], three heuristic schemes are presented for selecting a pair of MD video servers in a Content Delivery Network (CDN). In a recent work [12], Begen *et al.* study the problem of path selection for MD video streaming in service overlay networks. Finally, a multi-path congestion-based partitioning scheme is presented in [92] for optimizing received video quality, and the optimal traffic partitioning problem for multimedia data is studied in [73]. It is worth noting that this class of works focuses on a single video session, while the interactions among concurrent video sessions are not explicitly considered. In addition, except for [73], these works are approximation algorithms and there is a lack of theoretical study on the optimality of the proposed approaches.

4.6 Summary

In this chapter, we investigated the problem of path selection and rate allocation for concurrent video sessions in an ad hoc network. We formulated a network-wide optimal routing problem that minimizes the total distortion of all video sessions, by not only selecting the

best set of paths for video communication, but also, computing the optimal rate and partitioning it among the chosen set of paths. We modeled the end-to-end video distortion as a function of routing layer behavior. Our formulation captures the tight coupling that exists between the optimal encoding rate for each video session, the selection of paths for video transmission and the proportioning of traffic among these selected paths. An enhanced reformulation of this problem was presented, augmented by logarithmic convexification and RLT-based inequalities, and a specialized branch and bound algorithm was designed for solving this resulting model representation. Several computational experiments were performed using randomly generated network topologies to explore the efficacy of the proposed solution approach. The performance of the algorithm on different instances of the problem, ranging from 30-node network and 2 video sessions to 50-node network and 10 sessions is studied. Furthermore, the performance of the branch-and-bound algorithm is compared with a max-min rate allocation scheme [13], which is widely regarded as a fair rate allocation policy. The results revealed that when a single path is used for each video session, the branch-and-bound algorithm provides a lower bounding envelope of distortion values obtained using max-min scheme with any link utilization factor. In the multi-path case, when both both rate allocation and path selection are involved, we find that the branch-and-bound algorithm considerably outperforms the max-min rate allocation scheme, over the entire range of the decoding deadline.

Chapter 5

Multipath Routing for Multiple Description Video

5.1 Introduction

The problems considered thus far, focused on developing techniques to optimize the performance of multiple concurrent video sessions over multihop wireless networks. In doing so, we have not taken advantage of any specific video encoding scheme. Recently, a new video coding scheme called Multiple Description (MD) video coding is developed. This video coding technique is designed to generate substreams in such a way that the loss in one substream does not adversely affect the decoding of the other substreams. In this chapter therefore, we investigate the problem of multipath routing for multiple description video in multihop wireless networks.

MD video is an important coding technique for error resilience and control for multimedia applications [44, 114] and has been recognized as an ideal candidate for video streaming in multi-hop wireless networks [5, 6, 16, 41, 71]. Under MD coding, multiple *equivalent* streams (or descriptions) are generated for a video source for transmission. At the receiver, *any*

received subset of these streams can be combined to reconstruct the original video and the quality of the reconstructed video is commensurate with the number of received descriptions. This video coding technique is drastically different from traditional layered video coding, where video reconstruction hinges upon successful delivery of the base layer.

From a cross-layer routing perspective, the problem is to find a set of routes (or paths) in a network, one for each MD video stream such that the video distortion is minimized. The optimal multipath routing problem considered in this chapter is formulated into a *0-1 mixed-integer non-linear programming* problem. Such problems are shown to be NP-hard in general [96]. In a previous work [66], we studied this problem and solved it using genetic algorithms. Although GA remains an effective algorithm, it remains a metaheuristic which does not provide any performance bounds on how close the solution is to the optimal. As a result, a theoretical result on the multipath routing for multiple description video remains an open problem.

In this research effort, we fill in this important theoretical gap in cross-layer optimization for video communications. We design an optimization approach based on the Reformulation-Linearization Technique (RLT) to solve the multipath routing problem. The underlying 0-1 mixed-integer nonlinear programming problem is relaxed into a polynomial problem having a tight linear programming (LP) relaxation as prescribed by RLT, and a specialized branch-and-bound algorithm is designed to derive a global optimum.

The remainder of this chapter is organized as follows. We begin this chapter by comparing two contrasting video coding techniques, namely layered coding and multiple description coding in Section 5.2, followed by MD video generation in Section 5.3. In Section 5.4, we present the problem formulation. We then describe an RLT-based approach to reformulate and linearize the problem in Section 5.5, and develop a branch-and-bound-based algorithm in Section 5.6. Simulation results are presented in Section 5.7, and related work is discussed in Section 5.8. Finally, Section 5.9 concludes this chapter.

5.2 Layered Coding versus Multiple Description Coding

A common feature of wireline and wireless networks is the existence of multiple paths between nodes in a mesh topology. Video communications can take advantage of multiple paths by dividing the video stream into multiple substreams, so that each substream can be transmitted on a separate path. This reduces congestion in the network, thereby reducing congestion induced packet loss, and also allows for a better load balancing through out the entire network. To send video over multiple paths in ad hoc networks, we must have a suitable mechanism to generate sub-streams at the video source.

One way to generate multiple streams is to employ layered video coding scheme, that is often used in the case of wired networks. Layered video codecs can achieve elegant rate scalability [37, 56] i.e., they can generate layered embedded bit-streams that are decodable at different bitrates, with a gracefully degrading quality. This provides a convenient way for performing rate control required to mitigate network congestion [103]. Layered representations have become a part of the current video coding standards such as MPEG and H.263+ [49, 105], and layered representations for Internet streaming have been widely studied [47, 55, 86]. Furthermore, layered video coding has been proposed in combination with differentiated Quality-of-Service (DiffServ) [14] in the Internet [85, 99], where important layers can be delivered with better, but often more expensive QoS guarantees, while the less important layers with minimal or no QoS guarantees.

A layered codec encodes raw video into base layer and enhancement layers. The base layer (BL) provides a basic level of quality and can be decoded independently. On the other hand, the enhancement layers serve only to refine the base layer quality i.e., each additional enhancement layer (EL) can be used to further improve the video quality. But if the base layer is lost, it is not possible to reconstruct a video frame using the enhancement layers, even if they are received successfully (see Figure 5.1 (b)). In other words, the base layer

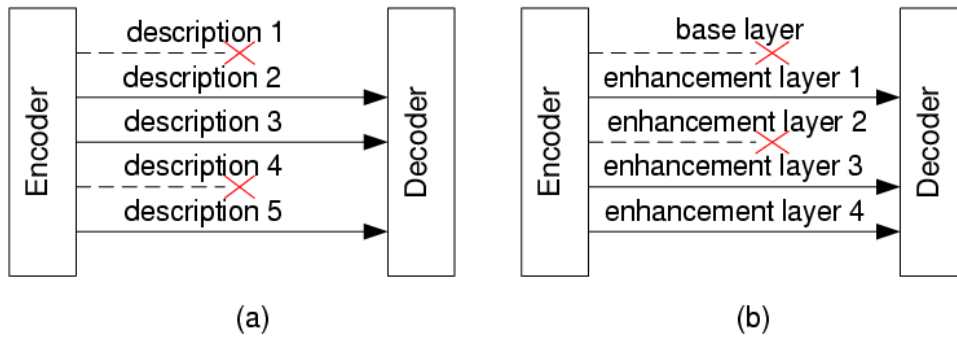


Figure 5.1: Layered coding versus multiple description coding.

represents the most critical part of the layered representation. As a result, the performance of streaming applications that employ layered representations is sensitive to losses in base layer packets. We find that transporting layered coded video over the Internet is highly successful because of the existence of a relatively reliable path from the source to the destination, such that packet losses and delays are within a range that can be effectively coped by error control and error concealment mechanisms. However, there is no guarantee of a single reliable path in wireless networks. As a result, it is neither possible to maintain a minimum quality for the perceived video, nor feasible to decode the upper enhancement layer video streams, despite the fact that upper enhancement layers might be received correctly via other paths at the receiver (see Figure 5.1 (b)). Therefore, layer video coding is not a viable approach for video transport in multihop wireless networks, especially under *extreme conditions*. For wireless networks operating under dynamic conditions, the wireless links are highly fragile, and there is a high degree of uncertainty that any particular path within the network will remain reliable over an extended period of time. In this environment, as discussed, traditional layered scalable video coding or single stream coding cannot perform well because either scheme requires at least one relatively reliable path from the sender to the receiver.

For multipath transport to be successful in sending compressed video, the video coder must be carefully designed to generate substreams so that the loss in one substream does not adversely affect the decoding of other substreams. Recently, MD coding has gained considerable attention as an alternative to layered coding for streaming over unreliable chan-

nels [44, 89, 111]. Under the MD coding paradigm, multiple “equally” important streams (or descriptions) are generated by the encoder. At the receiver, any subset of these streams can be used to reconstruct the original video (see Figure 5.1 (a)). That is, each description alone can guarantee a basic level of reconstruction quality of the source, and every additional description can further improve that quality.

In [4], Apostolopoulos introduced packet path diversity for video streaming. He proposed to send complementary descriptions of an MD coder through two different paths. Experimental results showed the potential benefits of the proposed system. Since then a number of studies have appeared that exploit the concept of packet path diversity in media communication. In [5], Apostolopoulos and Wee employed path diversity in the context of video communication using unbalanced MD coding to accommodate the fact that different paths might have different bandwidth constraints. The unbalanced descriptions are created by adjusting the frame rate of a description sent over a particular path. In [41], Gogate *et al.* studied image and video transmission in mobile radio networks. It was shown that combining MD coding and multiple path transport in such a setting provides higher bandwidth and robustness to end-to-end connections. In [71], Mao *et al.* showed the advantages of path diversity in combination with multistream coding by studying three techniques that are based on the motion compensated prediction, which is found in modern video coding standards. These schemes include feedback based reference picture selection, layered coding with selective automatic repeat request and multiple description motion compensation coding. The authors realized that each of these three video coding/transport techniques is best suited for a particular environment, depending on the availability of a feedback channel, the end-to-end delay constraint, and the error characteristics of the paths, and that a significant improvement of video quality can be achieved over standard schemes with limited additional cost.

MD coding matches perfectly with the wireless ad hoc network environment for multimedia applications [71]. This is because of the inherent mesh topology of such networks, and the existence of multiple paths between any source and destination pair. Indeed, within an

MD coding paradigm, as long as the link/node failure events on each path are not entirely correlated, it is possible to construct an acceptable quality video at the receiver in wireless ad hoc networks.

5.3 Generating MD Video

Multiple description coding has received considerable attention in recent times. From an information theoretic standpoint, many approaches for realizing MD objectives have been proposed. These include, using interleaved quantizers [33, 50, 107], interleaved spatial and/or temporal sub-sampling [57], pair-wise correlation transform [112, 113], correlating lapped orthogonal transform [22], correlating filter-banks [51] and coefficients splitting [88] in the discrete cosine transform (DCT) domain. These approaches vary widely in terms of their rate-distortion performance and complexity. Few of these approaches are can handle for a general source, while others are designed for specific applications, such as speech/audio or image/video (see [44] for an excellent survey).

Our interest in MD coding lies in developing an MD codec for video. A key challenge in designing an MD video coder is in the mismatch between the reference frames used in the encoder and decoder when only a single description is received in the decoder. Such a mismatch can be avoided by having independent prediction loops, each based on the reconstructed frames from a single description. Adapting the video redundancy coding (VRC) method for our cause (adopted in the H.263 standard), we partition the video frames into two descriptions using a temporal sub-sampling scheme, where two descriptions are generated by separating the even- and odd-numbered frames and encoding them separately, as shown in Figure 5.2. Therefore, for the rest of this chapter, we consider double description (DD) video. The first frame in each stream is coded in the *intra-mode* (I frame) and the following frames are coded in the *inter-mode* (P frame). a 10% macroblock level intra-refreshment is used, which has been found to be effective in suppressing error propagation for the range of

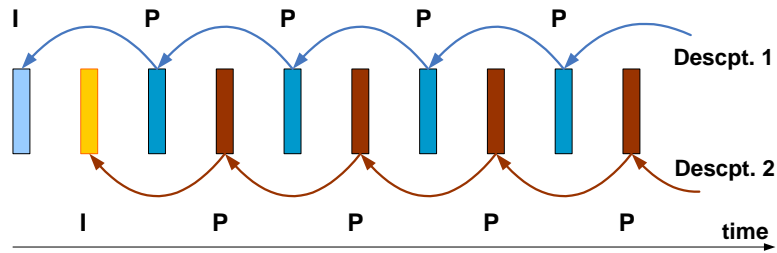


Figure 5.2: Generating multiple description video.

loss rates considered. This simple time-domain partitioning method is widely used in many video streaming studies [6, 12, 16, 71]. An H.263+ like codec is implemented to generate the two descriptions. This codec encodes the video sequence into two balanced descriptions (i.e., $R_1 = R_2$). From the network point of view, each Group of Blocks (GOB), or *slice*, is carried in a different packet. Packets from even-numbered frames are transmitted along one path while the packets from odd-numbered frames are transmitted on a different path. When a GOB is corrupted, the decoder applies a simple error concealment scheme by copying the corresponding slice from the most recent, correctly received frame. It is worth noting that the problem formulation and the solution approach discussed later in this chapter, are quite independent of the actual MD coder used, and can support other multiple description styles as well as other codecs (e.g., MPEG based codec instead of H.263+).

5.4 Problem Formulation

5.4.1 Network Model

We model a multi-hop wireless network as a directed graph $\mathcal{G}\{\mathcal{V}, \mathcal{E}\}$, where \mathcal{V} is the set of vertices representing wireless nodes and \mathcal{E} the set of edges representing wireless links. We assume that nodes are reliable during the video session, but links may be up or down with certain probabilities. For our routing problem, we focus on the network and link layer statistics, assuming that the physical and MAC layer dynamics from the underlying radio

environment are reflected in the network and link layer statistics. We characterize a link $\{i, j\} \in \mathcal{E}$ by:

- b_{ij} : the available bandwidth of link $\{i, j\}$. We assume that the impact from other wireless interference is accounted for through this link metric.
- p_{ij} : the probability that link $\{i, j\}$ is “up”.
- l_{ij} : average burst length for packet losses on link $\{i, j\}$.

These parameters may be measured at every node, and distributed throughout the network using Link State Advertisements (LSA) [23] or piggybacked in route replies (RREP) [83]. Based on these basic metrics, we can derive path-level bandwidth and failure probability, which are useful to characterize end-to-end performance at the video application layer (i.e., distortion).

Table 5.1 summarizes the notation used in this chapter.

5.4.2 Video Distortion and Path Level Statistics

Video Distortion

Consider a video session from video server s to client t . We assume that the video is encoded into two descriptions (i.e., double description video), each with a rate R_h bits/pixel, $h = 1, 2$. For video coding and communications, a distortion rate model addresses the problem of how much information R , needs to be transmitted over the communication channel such that the original video can be successfully decoded at the receiver with a given distortion d . For DD video, let d_h be the achieved distortion when only Description h is received, $h = 1, 2$, and d_0 the distortion when both descriptions are received. The rate-distortion region for a memoryless *i.i.d.* Gaussian source with a square error distortion measure was first introduced in [80]. For computational efficiency, the following distortion-rate function

Table 5.1: Summary of notation for Chapter 5

$\mathcal{G}\{\mathcal{V}, \mathcal{E}\}$	Graph representation of the network
\mathcal{V}	Set of vertices
\mathcal{E}	Set of edges
s	Source node
t	Destination node
\mathcal{P}	A path from s to t
$\{i, j\}$	A link from node i to node j
b_{ij}	Bandwidth of link $\{i, j\}$
p_{ij}	Success probability of link $\{i, j\}$
l_{ij}	Average length of loss burst on link $\{i, j\}$
R_h	Rate of Description h in bits/sample
R	For balanced descriptions, $R = R_1 = R_2$
d_0	Distortion when both descriptions are received
d_h	Distortion when only Description h is received, $h = 1, 2$
D	Average distortion
\bar{T}_{on}	Average “up” period of the joint links
Λ	“up” to “down” transition prob. for $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$
Ψ	“down” to “up” transition prob. for $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$
π_{00}	Probability of receiving both descriptions
π_{01}	Probability of receiving Description 1 only
π_{10}	Probability of receiving Description 2 only
π_{11}	Probability of losing both descriptions
$I_{ij}^{(h)}$	Routing index variables, defined in (5.3)
α_{ij}	“up” to “down” transition prob. of link $\{i, j\}$
β_{ij}	“down” to “up” transition prob. of link $\{i, j\}$
p_{jnt}	Average success prob. of joint links
$p_{dj}^{(h)}$	Average success prob. of disjoint links on \mathcal{P}_h

could be used [3, 66]:

$$\begin{cases} d_0 = \frac{2^{-2(R_1+R_2)}}{2^{-2R_1}+2^{-2R_2}-2^{-2(R_1+R_2)}} \cdot \sigma^2 \\ d_1 = 2^{-2R_1} \cdot \sigma^2 \\ d_2 = 2^{-2R_2} \cdot \sigma^2, \end{cases} \quad (5.1)$$

where σ^2 is the variance of the source. Since Gaussian source is the most difficult to encode, i.e., for a given square error distortion measure, it requires the most number of bits, the above video encoding scheme can be regarded as the worst case rate distortion region.

From end-to-end perspective, denote π_{00} as the probability of receiving both descriptions, π_{01} the probability of receiving Description 1 only, π_{10} the probability of receiving Description 2 only, and π_{11} the probability of losing both descriptions. Then, the expected average video distortion at the receiver can be approximated as:

$$D = \pi_{00} \cdot d_0 + \pi_{01} \cdot d_1 + \pi_{10} \cdot d_2 + \pi_{11} \cdot \sigma^2. \quad (5.2)$$

Finding the rate distortion region for MD video is still an open problem [44], and the MD region is well understood only for memoryless Gaussian sources with squared-error distortion measure, which bounds the MD region for any continuous-valued memoryless source with the same distortion measure. Our simulations show that although (5.2) is an approximation for DD video, significant improvement in video quality could be achieved over alternative approaches by incorporating it in the optimal routing problem formulation (see Section 5.7). Furthermore, our formulation does not depend on any specific distortion-rate function. A more accurate distortion-rate function could be easily incorporated into this formulation should it be available in the future.

Path-Level Statistics

To characterize a path \mathcal{P}_h between source node s and destination node t , we define:

$$I_{ij}^{(h)} = \begin{cases} 1, & \text{if link } \{i, j\} \in \mathcal{P}_h \\ 0, & \text{otherwise.} \end{cases} \quad (5.3)$$

An arbitrary path \mathcal{P}_h can then be represented by a vector $\mathbf{I}^{(h)}$ of $|\mathcal{E}|$ elements, each corresponding to a link and having a binary value.

For a source-destination pair $\{s, t\}$, consider two given paths $[\mathcal{P}_1, \mathcal{P}_2]$ in $\mathcal{G}\{\mathcal{V}, \mathcal{E}\}$. Since we do not mandate “disjointedness” between the two paths, \mathcal{P}_1 and \mathcal{P}_2 may share nodes and links¹. For each link $\{i, j\}$, the aggregate description rate should be bounded by its available bandwidth as

$$I_{ij}^{(1)} \cdot R_1 + I_{ij}^{(2)} \cdot R_2 \leq \rho \cdot b_{ij}, \quad (5.4)$$

where ρ is a constant. For a video with coding rate f frames/s and a resolution of $W \times V$ pixels/frame, we have $\rho = 1/(\kappa \cdot W \cdot V \cdot f)$, where κ is a constant determined by the chroma sub-sampling scheme. For example, when using quarter common intermediate format (QCIF) [176×144 Y pixels/frame, 88×72 Cb/Cr pixels/frame], we have $\kappa = 1.5$ and $\rho = 1/(1.5 \cdot 176 \cdot 144 \cdot f)$.

We now focus on how to compute the end-to-end path statistics. Similar to the approach in [6, 12, 66], we classify the links into three sets: set one consisting of links shared by both paths, denoted as $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$, and the other two sets consisting of disjoint links on the two paths, denoted as $\bar{\mathcal{J}}(\mathcal{P}_h)$, $h = 1, 2$, respectively. For disjoint portion of the paths, it suffices to model the packet loss as a Bernoulli event, since losses of the two descriptions are assumed to be independent on disjoint portions. The success probabilities on the disjoint portions

¹We argue that although stipulating disjoint paths may reduce the complexity of the problem considerably, purely pursuing disjointedness may lead to the use of low quality links (e.g., high loss rates, low bandwidths, or large delays). This inturn may result in a worse video quality than allowing the sharing of some “good” links in wireless ad hoc networks.

are:

$$p_{dj}^{(h)} = \begin{cases} \prod_{\{i,j\} \in \bar{\mathcal{J}}(\mathcal{P}_h)} p_{ij}, & \text{if } \bar{\mathcal{J}}(\mathcal{P}_h) \neq \emptyset, h = 1, 2 \\ 1, & \text{otherwise, } h = 1, 2. \end{cases} \quad (5.5)$$

On the joint portion of the paths, losses on the two streams are correlated. In order to model such correlation, we model each shared link $\{i, j\}$ as an on-off process modulated by a discrete-time Markov chain, as shown in Figure 5.3(a). There is no packet loss when the link is “up”; all packets are dropped when the link is “down”. Transition probabilities, $\{\alpha_{ij}, \beta_{ij}\}$, can be computed from the link statistics as $\beta_{ij} = 1/l_{ij}$ and $\alpha_{ij} = (1 - p_{ij})/(p_{ij}l_{ij})$. Note that we have $p_{ij} = \Psi/(\Lambda + \Psi)$, which is the equilibrium probability that the Markov chain is in the “up” state.

If there are K shared links, the aggregate failure process of these links is a Markov process with 2^K states. In order to simplify the computation, we follow the well-known Fritchman model [34] in modeling the aggregate process as an on-off process. Since a packet is successfully delivered on the joint portion if and only if all joint links are in the “up” state, we can lump up all the states with at least one link failure into a single “down” state, while using the remaining state where all the links are in good condition as the “up” state.

Let T_{on} be the length of the “up” period, then the average value \bar{T}_{on} can be computed as follows.

$$\begin{aligned} \bar{T}_{on} &= \sum_{i=1}^{\infty} i \cdot Prob\{T_{on} = i\} \\ &= \sum_{i=1}^{\infty} \left[i \cdot \prod_{k=1}^K (1 - \alpha_k)^{i-1} (1 - \prod_{k=1}^K (1 - \alpha_k)) \right] \\ &= 1 / \left[1 - \prod_{k=1}^K (1 - \alpha_{ij}) \right]. \end{aligned}$$

Therefore, in the context of the current problem,

$$\bar{T}_{on} = \frac{1}{1 - \prod_{\{i,j\} \in \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)} (1 - \alpha_{ij})}. \quad (5.6)$$

The transition probabilities of the aggregate on-off process can be computed as

$$\Lambda = \frac{1}{\bar{T}_{on}}, \quad \Psi = \frac{p_{jnt}}{[\bar{T}_{on}(1 - p_{jnt})]}, \quad (5.7)$$

where p_{jnt} is the average success probability of the joint portion, and

$$p_{jnt} = \begin{cases} \prod_{\{i,j\} \in \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)} p_{ij}, & \text{if } \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) \neq \emptyset \\ 1, & \text{otherwise.} \end{cases} \quad (5.8)$$

Note that $\Lambda = 0$ and $\Psi = 0$ if $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) = \emptyset$.

The consolidated path model is illustrated in Figure 5.3(b), where $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$ is modeled as a two-state Markov process with parameters $\{\Lambda, \Psi\}$, and $\bar{\mathcal{J}}(\mathcal{P}_h)$ is modeled as a Bernoulli process with parameter $(1 - p_{dj}^{(h)})$, $h = 1, 2$. With the consolidated path model, the joint probabilities of receiving the descriptions are:

$$\begin{cases} \pi_{00} = p_{jnt} \cdot (1 - \Lambda) \cdot p_{dj}^1 \cdot p_{dj}^2 \\ \pi_{01} = p_{jnt} \cdot p_{dj}^1 \cdot [1 - (1 - \Lambda) \cdot p_{dj}^2] \\ \pi_{10} = p_{jnt} \cdot [1 - (1 - \Lambda) p_{dj}^1] \cdot p_{dj}^2 \\ \pi_{11} = 1 - p_{jnt} \cdot [p_{dj}^1 + p_{dj}^2 - (1 - \Lambda) \cdot p_{dj}^1 \cdot p_{dj}^2]. \end{cases} \quad (5.9)$$

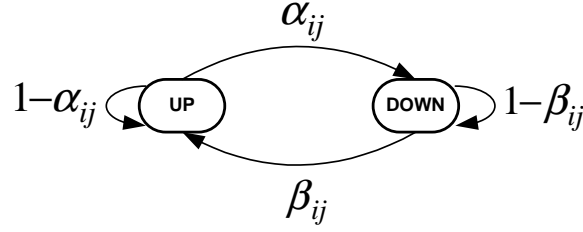
5.4.3 Mathematical Formulation

We can now formulate the problem of multipath routing for MD video into a mathematical programming problem as follows.

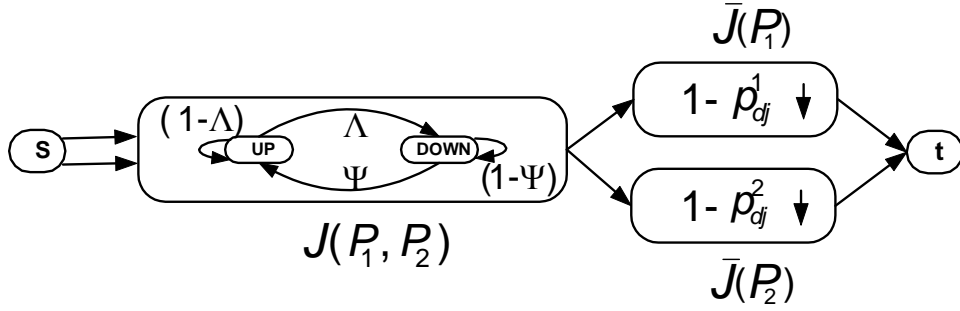
OPT-MR:

$$\text{Minimize} \quad D = \pi_{00} \cdot d_0 + \pi_{01} \cdot d_1 + \pi_{10} \cdot d_2 + \pi_{11} \cdot \sigma^2 \quad (5.10)$$

$$\text{subject to} \quad \sum_{j \in \mathcal{V}} I_{ij}^{(h)} - \sum_{j \in \mathcal{V}} I_{ji}^{(h)} = \begin{cases} 1, & \text{if } i = s, \quad i \in \mathcal{V}, h = 1, 2 \\ -1, & \text{if } i = t, \quad i \in \mathcal{V}, h = 1, 2 \\ 0, & \text{otherwise, } i \in \mathcal{V}, h = 1, 2 \end{cases} \quad (5.11)$$



(a) The Gilbert two-state link model.



(b) A path model characterizing joint and disjoint links.

Figure 5.3: Link and path models.

$$\sum_{i \in \mathcal{V}} I_{ij}^{(h)} \begin{cases} \leq 1, & \text{if } i \neq t, \quad j \in \mathcal{V}, \quad h = 1, 2 \\ = 0, & \text{if } i = t, \quad j \in \mathcal{V}, \quad h = 1, 2 \end{cases} \quad (5.12)$$

$$I_{ij}^{(1)} \cdot R_1 + I_{ij}^{(2)} \cdot R_2 \leq \rho \cdot b_{ij}, \quad \{i, j\} \in \mathcal{E} \quad (5.13)$$

$$I_{ij}^{(h)} \in \{0, 1\}, \quad \{i, j\} \in \mathcal{E}, \quad h = 1, 2. \quad (5.14)$$

In Problem OPT-MR, $\{I_{ij}^{(h)}\}$ are binary optimization variables (incorporated in π_{00} , π_{01} , π_{10} , and π_{11}). Constraint (5.11) guarantees that the paths originate at the source s and terminate at the destination t , and constraint (5.12) ensures that the paths are loop-free. Constraint (5.13) guarantees that the link capacity limit is observed. For a given pair of paths, the average video distortion D is determined by the end-to-end statistics and the correlation of the paths, as given in (5.1) and (5.9). The impact of each of the link characteristics on the average distortion is captured by the problem formulation. Specifically, with larger end-to-end bandwidth, the video rate for each description could be higher and therefore the impact of the encoder on distortion is relatively low (i.e., d_0 , d_1 , and d_2 are all

decreasing functions of the description rates). With a lower end-to-end loss rate, fewer video frames will be corrupted. This is modeled in (5.10), where σ^2 is usually much larger than d_0 , d_1 , and d_2 , and d_h is usually larger than d_0 , $h = 1, 2$. Finally, the impact of path correlation is actually considered in the derivation of the joint probabilities of receiving the description. In Problem OPT-MR, all the three elements are integrated in the objective function (5.10), and are jointly optimized in routing.

The objective function (5.10) is a complex ratio of high-order exponentials of the I -variables. The objective evaluation of a pair of paths involves identifying the joint and disjoint portions, which is only possible when both paths are completely determined. Since such problems are NP-hard in general [35], and Problem OPT-MR does not appear to possess any special simplifying structure, it is likely to be NP-hard.

In the following sections, we present an algorithm based on branch-and-bound framework [76] for Problem OPT-MR, predicated on RLT. Our proposed solution procedure can produce a solution within a relative error of ϵ with regard to the global optimum, where $\epsilon \in (0, 1)$ can be made arbitrarily small depending on required accuracy.

5.5 Reformulation and Linearization

Our solution approach to Problem OPT-MR is to embed RLT in a branch-and-bound framework [94, 97]. In this section, we show how to reformulate Problem OPT-MR to obtain an easier-to-solve linear relaxation. We then develop a branch-and-bound-based algorithm that finds ϵ -optimal solution in the next section. We first reformulate Problem OPT-MR into a *0-1 mixed-integer polynomial programming* problem P-MR. Then, we replace all the non-linear terms and add the corresponding RLT constraints into the problem formulation, so as to obtain a *linear programming relaxation* of Problem OPT-MR, denoted as L-MR.

5.5.1 Reformulation

As discussed, the objective function of Problem OPT-MR is a complex function of exponential terms of the I -variables. Our first goal is to reformulate these terms, which will simplify the objective function and the constraints. Without loss of generality, we set $\sigma^2 = 1$ to simplify notation. Note that σ^2 only affects the absolute value of distortion, but not optimal routing selection.

In (5.9), there are four high order terms that need to be reformulated, namely, p_{jnt} , $p_{dj}^{(1)}$, $p_{dj}^{(2)}$, and Λ . From their definitions in (5.5) and (5.8), we can rewrite the success probabilities as:

$$\begin{cases} p_{jnt} = \prod_{\{i,j\} \in \mathcal{E}} p_{ij}^{\{I_{ij}^{(1)} \cdot I_{ij}^{(2)}\}} \\ p_{dj}^{(1)} = \prod_{\{i,j\} \in \mathcal{E}} p_{ij}^{\{I_{ij}^{(1)} \cdot (1 - I_{ij}^{(2)})\}} \\ p_{dj}^{(2)} = \prod_{\{i,j\} \in \mathcal{E}} p_{ij}^{\{I_{ij}^{(2)} \cdot (1 - I_{ij}^{(1)})\}} \end{cases} \quad (5.15)$$

Taking logarithms on both sides, we can convert the high order terms on the right-hand-side (RHS) of (5.15) into summations of quadratic terms of the I -variables, i.e.,

$$\begin{cases} \log(p_{jnt}) = \sum_{\{i,j\} \in \mathcal{E}} \left[I_{ij}^{(1)} \cdot I_{ij}^{(2)} \cdot \log(p_{ij}) \right] \\ \log(p_{dj}^{(1)}) = \sum_{\{i,j\} \in \mathcal{E}} \left[I_{ij}^{(1)} \cdot (1 - I_{ij}^{(2)}) \cdot \log(p_{ij}) \right] \\ \log(p_{dj}^{(2)}) = \sum_{\{i,j\} \in \mathcal{E}} \left[I_{ij}^{(2)} \cdot (1 - I_{ij}^{(1)}) \cdot \log(p_{ij}) \right] \end{cases} \quad (5.16)$$

Similarly, we can rewrite Λ according to (5.6) and (5.7) as

$$\Lambda = 1 - \prod_{\{i,j\} \in \mathcal{E}} \left[1 - \frac{1 - p_{ij}}{p_{ij} \cdot l_{ij}} \right]^{\{I_{ij}^{(1)} \cdot I_{ij}^{(2)}\}} \quad (5.17)$$

Letting $\phi = 1 - \Lambda$ and taking logarithms on both sides, we have

$$\log(\phi) = \sum_{\{i,j\} \in \mathcal{E}} \left[I_{ij}^{(1)} \cdot I_{ij}^{(2)} \cdot \log(h_{ij}) \right], \quad (5.18)$$

where $h_{ij} = 1 - (1 - p_{ij}) / (p_{ij} \cdot l_{ij})$ is a constant for all $\{i, j\} \in \mathcal{E}$.

Having simplified the high-order terms, we now deal with the resulting constraints of the form $y = \log(\lambda)$, as shown in (5.16) and (5.18). We can linearize this logarithmic relationship

in a manner similar to the discussion in Section 4.3.1. Note that such polyhedral outer approximations will be iteratively tightened during the branch-and-bound procedure (see Section 2.3.2).

Substituting ϕ , we can rewrite (5.9) as

$$\begin{cases} \pi_{00} = p_{jnt} \cdot p_{dj}^{(1)} \cdot p_{dj}^{(2)} \cdot \phi \\ \pi_{01} + \pi_{00} = p_{jnt} \cdot p_{dj}^{(1)} \\ \pi_{10} + \pi_{00} = p_{jnt} \cdot p_{dj}^{(2)} \\ \pi_{00} + \pi_{01} + \pi_{10} + \pi_{11} = 1, \end{cases} \quad (5.19)$$

and the objective function can be rewritten as

$$\begin{aligned} D = & \pi_{00} \cdot d_0 + (p_{jnt} \cdot p_{dj}^{(1)} - \pi_{00}) \cdot d_1 + (p_{jnt} \cdot p_{dj}^{(2)} - \pi_{00}) \cdot d_2 + \\ & (1 + \pi_{00} - p_{jnt} \cdot p_{dj}^{(1)} - p_{jnt} \cdot p_{dj}^{(2)}). \end{aligned} \quad (5.20)$$

This reduces OPT-MR into an approximated *0-1 mixed-integer polynomial programming problem* P-MR, with the objective being minimizing D in (5.20). The constraints of P-MR include the original constraints (5.11)–(5.14), the reformulated constraint (5.19), and the new constraints derived from reformulating the logarithmic terms (5.16) and (5.18) [in the form of (4.19)].

5.5.2 Linearization

Although greatly simplified, Problem P-MR is still a polynomial programming problem, which is NP-hard in general [94]. In this section, we linearize Problem P-MR by employing RLT which involves variable substitutions and introducing linear RLT bound-factor constraints.

Consider a quadratic product term of the form $(p_{dj}^{(1)} \cdot p_{dj}^{(2)})$. By introducing a new variable $z_0 = p_{dj}^{(1)} \cdot p_{dj}^{(2)}$, we can substitute the $(p_{dj}^{(1)} \cdot p_{dj}^{(2)})$ terms in (5.19) and (5.20) with z_0 , thus removing this quadratic term from the objective function and constraints. Assuming $p_{dj}^{(1)}$

and $p_{dj}^{(2)}$ are each bounded as $\left(p_{dj}^{(1)}\right)_L \leq p_{dj}^{(1)} \leq \left(p_{dj}^{(1)}\right)_U$ and $\left(p_{dj}^{(2)}\right)_L \leq p_{dj}^{(2)} \leq \left(p_{dj}^{(2)}\right)_U$, respectively, we can add the following relational constraints, which are known as the *RLT bound-factor product constraints*:

$$\left\{ \begin{array}{l} \left\{ \left[p_{dj}^{(1)} - \left(p_{dj}^{(1)}\right)_L \right] \cdot \left[p_{dj}^{(2)} - \left(p_{dj}^{(2)}\right)_L \right] \right\}_{LS} \geq 0 \\ \left\{ \left[p_{dj}^{(1)} - \left(p_{dj}^{(1)}\right)_L \right] \cdot \left[\left(p_{dj}^{(2)}\right)_U - p_{dj}^{(2)} \right] \right\}_{LS} \geq 0 \\ \left\{ \left[\left(p_{dj}^{(1)}\right)_U - p_{dj}^{(1)} \right] \cdot \left[p_{dj}^{(2)} - \left(p_{dj}^{(2)}\right)_L \right] \right\}_{LS} \geq 0 \\ \left\{ \left[\left(p_{dj}^{(1)}\right)_U - p_{dj}^{(1)} \right] \cdot \left[\left(p_{dj}^{(2)}\right)_U - p_{dj}^{(2)} \right] \right\}_{LS} \geq 0, \end{array} \right.$$

where $\{\cdot\}_{LS}$ denotes a *linearization step* under the substitution $z_0 = p_{dj}^{(1)} \cdot p_{dj}^{(2)}$. From the above relationships and by substituting $z_0 = p_{dj}^{(1)} \cdot p_{dj}^{(2)}$, we obtain the following RLT constraints for z_0 .

$$\left\{ \begin{array}{l} \left(p_{dj}^{(1)}\right)_L \cdot p_{dj}^{(2)} + \left(p_{dj}^{(2)}\right)_L \cdot p_{dj}^{(1)} - z_0 \leq \left(p_{dj}^{(1)}\right)_L \cdot \left(p_{dj}^{(2)}\right)_L \\ \left(p_{dj}^{(1)}\right)_L \cdot p_{dj}^{(2)} + \left(p_{dj}^{(2)}\right)_U \cdot p_{dj}^{(1)} - z_0 \geq \left(p_{dj}^{(1)}\right)_L \cdot \left(p_{dj}^{(2)}\right)_U \\ \left(p_{dj}^{(1)}\right)_U \cdot p_{dj}^{(2)} + \left(p_{dj}^{(2)}\right)_L \cdot p_{dj}^{(1)} - z_0 \geq \left(p_{dj}^{(1)}\right)_U \cdot \left(p_{dj}^{(2)}\right)_L \\ \left(p_{dj}^{(1)}\right)_U \cdot p_{dj}^{(2)} + \left(p_{dj}^{(2)}\right)_U \cdot p_{dj}^{(1)} - z_0 \leq \left(p_{dj}^{(1)}\right)_U \cdot \left(p_{dj}^{(2)}\right)_U. \end{array} \right.$$

By adding the linear RLT bound-factor constraints for z_0 into the problem formulation, we can therefore replace the second-order term $p_{dj}^{(1)} \cdot p_{dj}^{(2)}$ with the linear term z_0 in (5.19) and (5.20).

Similarly, we define new variables for all the remaining non-linear terms that are found in the reformulated problem OPT-MR(p), including $z_1 = p_{jnt} \cdot p_{dj}^{(1)}$, $z_2 = p_{jnt} \cdot p_{dj}^{(2)}$, and $z_3 = z_0 \cdot \phi$, and make substitutions in the same manner. We can then rewrite the objective function (5.20) and constraints (5.19) as

$$D = \pi_{00} \cdot d_0 + (z_1 - \pi_{00}) \cdot d_1 + (z_2 - \pi_{00}) \cdot d_2(1 + \pi_{00} - z_1 - z_2) \quad (5.21)$$

and

$$\left\{ \begin{array}{l} \pi_{01} + \pi_{00} = z_1 \\ \pi_{10} + \pi_{00} = z_2 \\ \pi_{00} + \pi_{01} + \pi_{10} + \pi_{11} = 1. \end{array} \right. \quad (5.22)$$

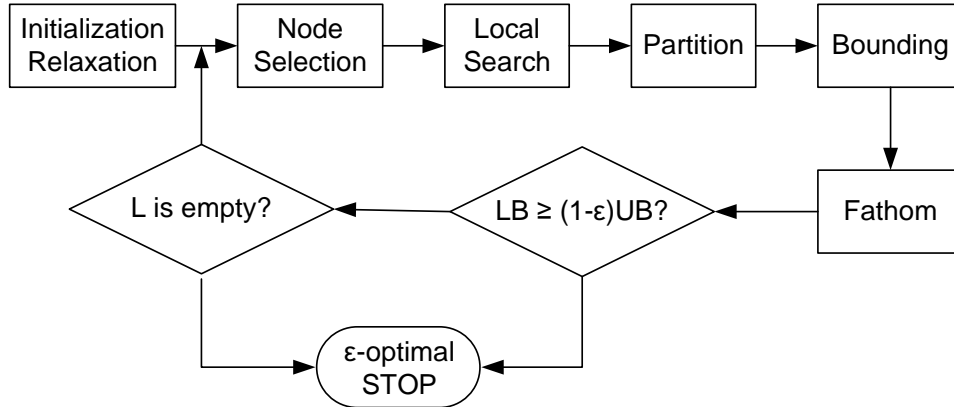


Figure 5.4: Flowchart for the ϵ -optimal solution procedure $\text{ALG}(\epsilon)$.

The constraints derived from reformulating the logarithmic terms (5.16) and (5.18) [in the form of (4.19)] can also be linearized by substituting $z_{ij} = I_{ij}^{(1)} \cdot I_{ij}^{(2)}$, and by introducing the corresponding linear RLT bound-factor constraints, for all $\{i, j\} \in \mathcal{E}$.

As a result, we obtain a *linear programming relaxation* Problem L-MR, which can be solved in polynomial-time.

5.6 Branch-and-Bound based Solution Procedure

We now describe the details of the specialized branch-and-bound algorithm based solution procedure, which we call $\text{ALG}(\epsilon)$, that produces ϵ -optimal solutions to the OPT-MR problem. Figure 5.4 shows the flowchart of this algorithm. The iterative branch-and-bound algorithm terminates with an ϵ -optimal solution when either the lower bound for the original problem is within ϵ of the upper bound, i.e., $LB \geq (1 - \epsilon) \cdot UB$, or the problem list L is empty. The operation of each step in $\text{ALG}(\epsilon)$ is described in the following.

Initialization and Relaxation

We start by initializing the current “best” solution, denoted as ψ^* , with the solution $\bar{\psi}$ obtained as described below, and the current “best” upper bound UB as the objective value obtained using this solution $\bar{\psi}$. We first solve the relaxed problem L-MR to obtain a possibly infeasible solution $\hat{\psi}$, for the original problem. For example, due to the RLT relaxation, the binary I -variables in $\hat{\psi}$ could actually be fractional. If $\hat{\psi}$ is already feasible to Problem OPT-MR, we set $\bar{\psi} = \hat{\psi}$ and then obtain a feasible solution in one iteration. Otherwise, we apply a local search algorithm (will be described in Section 5.6) to obtain a rounded feasible solution $\bar{\psi}$ to Problem OPT-MR. If necessary, we can also perform a restricted search by making a limited perturbation around the rounded feasible solution $\bar{\psi}$ in order to obtain an even better solution. This can be achieved by incorporating a new constraint to the LP relaxation,

$$\sum_{(i,j):\bar{\psi}_{ij}^{(h)}=0} \psi_{ij}^{(h)} + \sum_{(i,j):\bar{\psi}_{ij}^{(h)}=1} (1 - \psi_{ij}^{(h)}) \leq r_h, h = 1, 2, \quad (5.23)$$

for some integral deviation tolerance $r_h, h = 1, 2$.

That is, a feasible solution $\bar{\psi}$ is obtained by solving the root node of the branch-and-bound tree and applying an efficient local search to the resulting solution. Due to the properly designed RLT relaxations (see Section 5.5), the solution $\bar{\psi}$ is highly competitive in itself, and in many cases it achieves ϵ -optimality.

Once ψ^* is initialized, we then initialize the problem list L with the original problem, denoted as $PP(\Omega)$ (see Section 2.3.1). We denote the objective value obtained from the linear programming relaxation $LP(\Omega)$ as the lower bound LB_1 for $PP(\Omega)$. Also, since this is the only problem in the problem list, we initialize LB_1 as the current “best” lower bound LB for the original problem, i.e., set $LB = LB_1$.

Node Selection

At every iteration, problem k (or the corresponding node in the branch-and-bound tree) that has the minimum LB_k among all the problems $k \in L$ is selected. As discussed before, this problem is indicative of the lower bound for the original problem. Subsequent operations of local search, partitioning and bounding are performed on this problem k .

Local Search

As discussed in Section 2.3.2, the solution to the relaxation problem k that is selected in the node selection step, is usually infeasible to the original problem $PP(\Omega)$. This is especially true if the original problem involves binary variables (i.e., the I -variables could be fractions). A local search algorithm should be used to find a feasible solution to the original problem starting from the infeasible lower bounding solution.

Let $\hat{\psi}$ be the infeasible (or fractional) solution obtained by solving the LP relaxation of the original problem. Starting from this fractional solution, we solve for $h = 1, 2$ the following shortest path problem:

$$\text{Minimize} \quad \sum_{\{i,j\} \in \mathcal{E}} (-\hat{I}_{ij}^{(h)}) \cdot I_{ij}^{(h)} \quad (5.24)$$

subject to the flow constraints. Note that for an optimization variable y , \hat{y} denotes its value in the infeasible solution $\hat{\psi}$. Solving these shortest path problems provides us with a rounded heuristic solution $\bar{\psi}$ that has a tendency to round up relatively higher-valued components of $\hat{\psi}$ and round down relatively lower-valued components. The distortion value of the rounded solution $\bar{\psi}$ is an upper bound for this subproblem, i.e., UB_k .

Partitioning

The objective of the partitioning step is to find the branching variable that will enable us to split the feasible solution space Ω_k of problem k into two solution sub-spaces Ω_{k_1} and Ω_{k_2} . In

ALG(ϵ), we need to consider three classes of optimization variables for partitioning, i.e., the binary I -variables, the substitution variables (e.g., z_0), and the the logarithm substitution terms [e.g., ϕ in (5.18)].

When partitioning based on the I -variables, we need to select a variable that will offer the highest gain in terms of improving the objective value. For this purpose, we should choose the I -variable which is fractional and the closest to 0.5. A strategy that works well is to first find the index variable pair $\{I_{ij}^{(1)}, I_{ij}^{(2)}\}$, for all $\{i, j\} \in \mathcal{E}$ that gives the largest discrepancy between the RLT substitution variable \hat{z}_{ij} and the corresponding non-linear product $(\hat{I}_{ij}^{(1)} \cdot \hat{I}_{ij}^{(2)})$ (see Section 5.5.2). We then choose $I_{ij}^{(1)}$ or $I_{ij}^{(2)}$ to partition the problem (by fixing it to 0 or 1) depending on which variable is closer to 0.5. We break a tie arbitrarily.

In addition to the I -variables, we also need to examine branching decisions based on the substitution variables such as $z_0 = p_{dj}^{(1)} \cdot p_{dj}^{(2)}$. For such variables, we first find the maximum relaxation error between the substitution variable and the corresponding product term, say, $|\hat{p}_{dj}^{(1)} \cdot \hat{p}_{dj}^{(2)} - \hat{z}_0|$. We then verify whether the following condition is satisfied.

$$\begin{aligned} & \left[\left(p_{dj}^{(1)} \right)_U - \left(p_{dj}^{(1)} \right)_L \right] \cdot \min \{ \hat{p}_{dj}^{(1)} - \left(p_{dj}^{(1)} \right)_L, \left(p_{dj}^{(1)} \right)_U - \hat{p}_{dj}^{(1)} \} \geq \\ & \left[\left(p_{dj}^{(2)} \right)_U - \left(p_{dj}^{(2)} \right)_L \right] \cdot \min \{ \hat{p}_{dj}^{(2)} - \left(p_{dj}^{(2)} \right)_L, \left(p_{dj}^{(2)} \right)_U - \hat{p}_{dj}^{(2)} \}. \end{aligned}$$

If this condition holds true, we partition the solution space Ω_k of problem k into two new regions Ω_{k_1} and Ω_{k_2} , by dividing the range $\left[\left(p_{dj}^{(1)} \right)_L, \left(p_{dj}^{(1)} \right)_U \right]$ into two subregions $\left[\left(p_{dj}^{(1)} \right)_L, \hat{p}_{dj}^{(1)} \right]$ and $\left[\hat{p}_{dj}^{(1)}, \left(p_{dj}^{(1)} \right)_U \right]$. Otherwise, we partition Ω_k by dividing $\left[\left(p_{dj}^{(2)} \right)_L, \left(p_{dj}^{(2)} \right)_U \right]$ into $\left[\left(p_{dj}^{(2)} \right)_L, \hat{p}_{dj}^{(2)} \right]$ and $\left[\hat{p}_{dj}^{(2)}, \left(p_{dj}^{(2)} \right)_U \right]$.

Finally, the branching decisions also include the logarithm substitution terms, e.g., ϕ in (5.18). In such cases, we first find the variable that gives the greatest discrepancy between the logarithm value, say, $\log(\hat{\phi})$ and the RHS of the corresponding substitution [e.g., (5.18)] among all such terms, and then either bisect the interval of this variable (e.g., $[(\phi)_L, (\phi)_U]$) evenly, or divide this interval at the point $\hat{\phi}$.

Bounding

In the bounding step, we solve the RLT relaxation for the two sub-problems identified in the partitioning step, and obtain their corresponding lower bounds LB_{k_1} and LB_{k_2} , thereby updating the incumbent lower bounding solution. The corresponding upper bounds, i.e., UB_{k_1} and UB_{k_2} , are obtained by applying the local search algorithm starting from the relaxation solutions obtained, and the current LB and UB values for the original problem $PP(\Omega)$ are updated according to (2.4). If any of the following conditions

$$(1 - \epsilon) \cdot UB > LB_{k_1} \quad \text{and} \quad (1 - \epsilon) \cdot UB > LB_{k_2}$$

are satisfied, we add the corresponding problem into the problem list L , and remove problem k from the list.

Fathoming

For any problem k in the problem list L , if $LB_k \geq (1 - \epsilon) \cdot UB$, then the sub-space corresponding to this problem does not contain any solution that improves beyond the ϵ -tolerance of the incumbent solution. Therefore, we can prune this problem from the problem list, such that the computation time can be reduced.

5.7 Simulation Studies

In this section, we present simulation results for the proposed solution procedure to the cross-layer optimization problem. The simulation study consists of two parts. In the first part, we will examine the convergence behavior of the proposed solution procedure. In the second part, we will demonstrate the performance advantage of the proposed cross-layer approach over a non-cross-layer approach.

Throughout our simulation study, we consider a multi-hop wireless ad hoc network de-

Table 5.2: Performance of the proposed algorithm ($\epsilon = 0.01$)

Number of Nodes	Mean Computation Time(sec)	Variance
20	0.090	0.005
30	0.242	0.139
50	0.699	0.200
100	5.171	2.714

ployed over a rectangular region, where the connectivity between the nodes is determined by the distance between nodes' transmitter. The size of the area depends on the network size (total number of nodes) and will be described when we introduce the specific network. The source node s and destination node t are chosen randomly from the nodes in the network.

At the link level, we associate each link with a failure probability, available bandwidth, and mean burst length (for packet loss). Specifically, we associate each link with a failure probability, taken uniformly between $[0.01, 0.3]$; we associate each link with an available bandwidth, taken with equal probability from the set $[100, 150, 200, 250, 300, 350, 400]$ Kb/s; the mean burst length at each link is chosen uniformly between $[2, 6]$.

For our solution procedure, we set $\epsilon = 0.01$ (or 1%), which will be adequate for most application requirements in practice. We implement the proposed solution procedure in C program and use the LINDO API 3.0 for solving the LP relaxation problem.

5.7.1 Convergence Behavior

We first examine the convergence behavior of the proposed solution procedure for different network sizes and topologies. Table 5.2 shows the convergence time performance for 4 network sizes (20, 30, 50 and 100 nodes) with $\epsilon = 0.01$ and description rates $R_1 = R_2 = 320$ Kb/s. For each network size, we generate 100 topologies and run 100 computations to

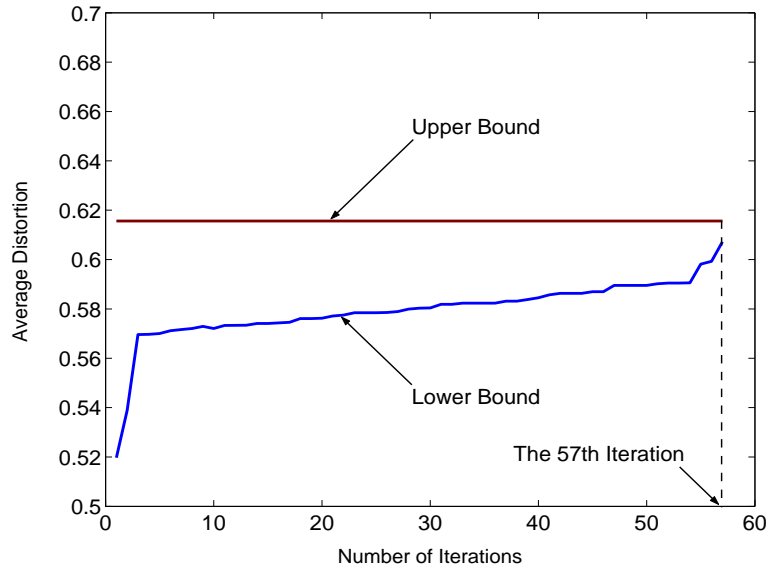
obtain the mean and variance of convergence time. The rectangular region for 20-, 30-, 50-, and 100-node networks are 300m x 300m, 400m x 400m, 500m x 500m, and 1000m x 1000m, respectively. The transmission range for each node is assumed to be 150m. The computation using our proposed solution procedure was run on a standard desktop PC with Pentium-4 2.4 GHz processor and 512 MB memory.

As shown in Table 5.2, the computational time for convergence to $\epsilon = 0.01$ is very fast (with average less than 1 second) for small to moderate sized network. Even for 100-node network, the average convergence time is only 5.171 second. With a high performance processor, we expect this time will be further reduced.

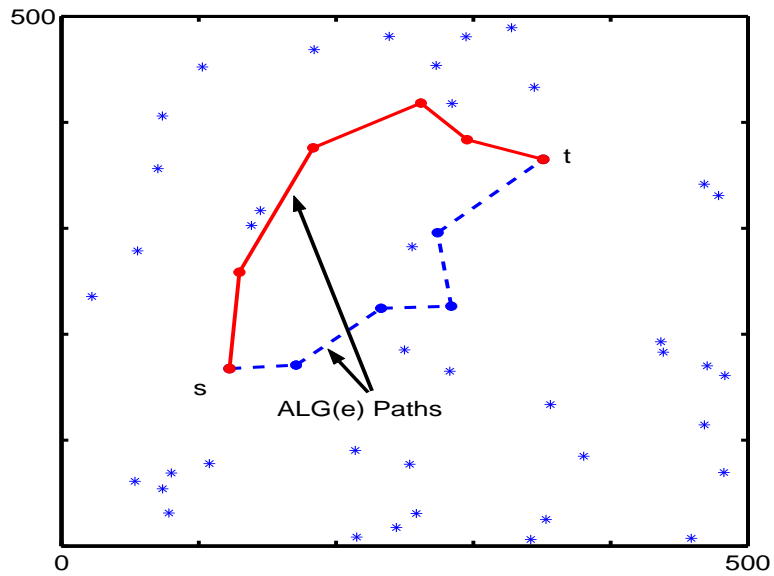
We notice that for most experiments, the algorithm converges very quickly and finds ϵ -optimal paths. To see the iterative convergence behavior of our solution procedure, we intentionally pick an experiment (from 100 experiments) with a convergence time longer than the average. Figures 5.5 and 5.6 show such instances for the 50-node network and 100-node network, respectively. For this particular 50-node network, the gap between upper bound UB and lower bound LB converges to $\epsilon = 1\%$ (i.e., $LB \geq 99\% \cdot UB$) after the 57th iteration within 1.87 second. As expected, the gap between LB and UB is strictly non-increasing with the number of iterations. Similarly, for this particular 100-node network, the solution procedure converges to $\epsilon = 1\%$ after the 39th iteration within 9.26 second. Figures 5.5(b) and 5.6(b) show the final optimal paths obtained by our solution procedure for the 50- and 100-node networks respectively.

5.7.2 Impact of Description Rates

In this section we study the performance of the proposed algorithm by varying the encoding rate of the video descriptions. Figure 5.7 presents the computation times of $ALG(\epsilon)$ for two different network topologies: a 50-node and a 30-node network. The video description rate is increased from 64 Kb/s to 320 Kb/s, in steps of 64K b/s. The computation time required by $ALG(\epsilon)$ to obtain the ϵ -optimal solutions depends on the number of subproblems examined

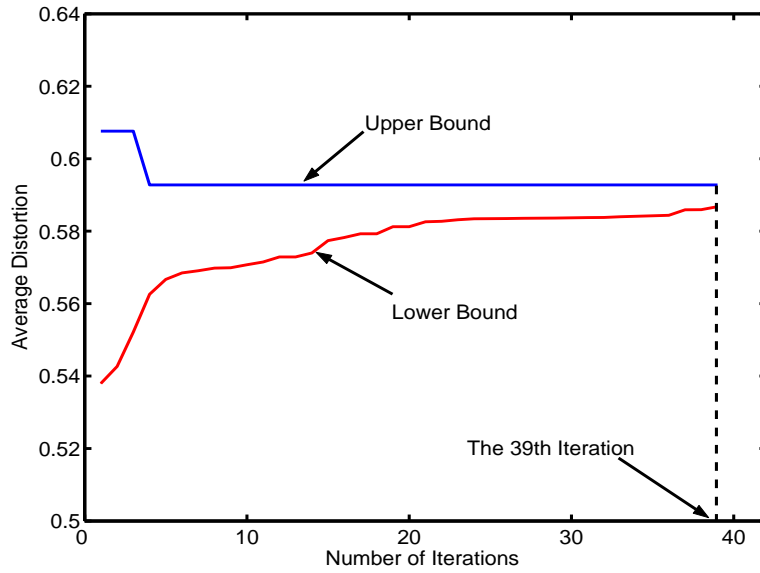


(a) Convergence behavior

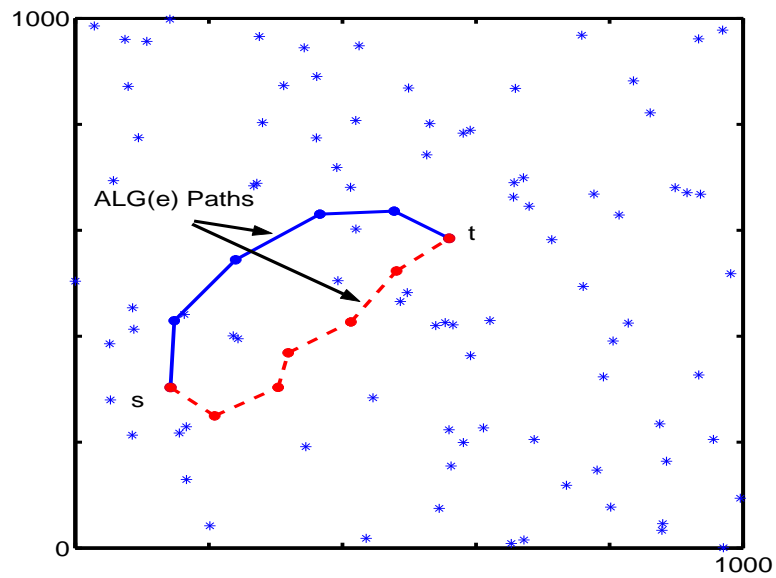


(b) Optimal paths for double description video

Figure 5.5: Convergence behavior and final optimal solution by the proposed solution procedure for a 50-node network.



(a) Convergence behavior



(b) Optimal paths for double description video

Figure 5.6: Convergence behavior and final optimal solution by the proposed solution procedure for a 100-node network.

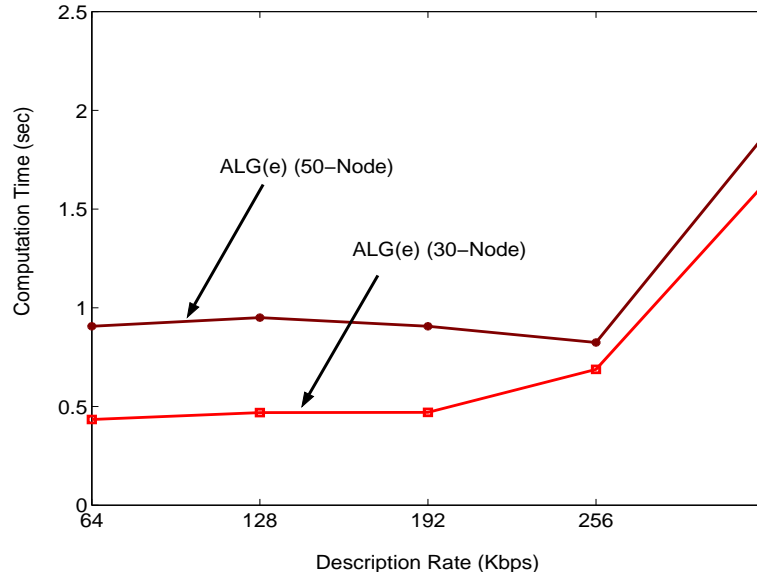


Figure 5.7: Impact of description rate R on $\text{ALG}(\epsilon)$ computation time.

(or nodes in the corresponding branch-and-bound tree), before the algorithm terminates. It can be seen that the number of nodes that are examined generally increases with the description rate.

In Figure 5.8, we plot the distortion achieved by the proposed algorithm for different description rates. The same 50-node networks is used. We also vary the link failure probabilities [i.e., $(1 - p_{ij})$]: (i) in the LR1 case, the link failure probabilities are uniformly chosen between $[0.01, 0.1]$, and (ii) in the LR2 case, the link failure probabilities are uniformly chosen between $[0.01, 0.3]$.

Note that video distortion is computed against the original raw video. As a result, there will be two types of distortion in a reconstructed video: (i) distortion due to the lossy video coder, and (ii) distortion due to transmission errors. In Figure 5.8, as expected, distortion generally decreases with increased description rates in both cases. This is because the higher the description rate, the smaller the distortion due to the coder. We also observe that the reduction in distortion slows down, and the discrepancy between the two curves increases for further increases in R . This is because for a large R , there will be fewer links eligible

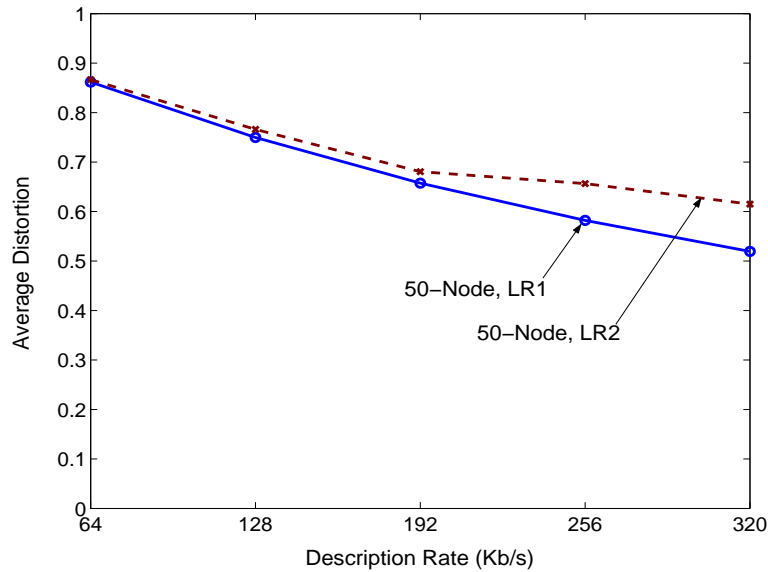


Figure 5.8: Impact of description rate R on average distortion obtained by $\text{ALG}(\epsilon)$.

for routing due to insufficient available bandwidth. $\text{ALG}(\epsilon)$ may be forced to choose high loss rate links (having a sufficiently large b_{ij}) in these cases, which diminishes the benefits gained at the coder due to the increased rate. Such diminishing effect is more obvious in a high loss network (LR2) than in a low loss network (LR1).

5.7.3 Comparison with Non-Cross-Layer Routing

In this section, we compare our cross-layer routing approach with a popular non-cross-layer approach. For the latter, we consider the k -shortest path (SP) routing algorithm [31], with $k = 2$ or 2-SP for double description video. We use hop count as the routing metric in the 2-SP algorithm.

Distortion Comparison

Table 5.3 compares the mean distortion achieved using our $\text{ALG}(\epsilon)$ and 2-SP for 4 network sizes (20, 30, 50 and 100 nodes) with $\epsilon = 0.01$ and $R = 320$ Kb/s. Again, for each network

Table 5.3: Average distortion values for different network sizes ($\epsilon = 0.01$)

Number of Nodes	ALG(ϵ)	2-SP
20	0.515	0.589
30	0.516	0.591
50	0.508	0.635
100	0.512	0.575

Table 5.4: Average PSNR values(dB) for different network sizes ($\epsilon = 0.01$)

Number of Nodes	ALG(ϵ)	2-SP
20	28.735	19.656
30	27.313	17.203
50	31.865	17.033
100	29.809	20.418

size, we generate 100 topologies and run 100 computations to obtain the mean distortion. As shown in Table 5.3, the distortion values under our cross-layer algorithm are consistently smaller than those under the 2-SP (non-cross-layer) approach.

Video Quality Comparison

We now encode a video sequence in order to transmit the double description video over the network, and compare the video quality at the receiver (measured using PSNR) under our cross-layer approach and the 2-SP approach. There are many ways to generate MD video (see [114]). We chose a time-domain partitioning coding scheme, where two descriptions are generated by separating the even- and odd-numbered frames and encoding them separately. This simple time-domain partitioning method is widely used in many video streaming studies

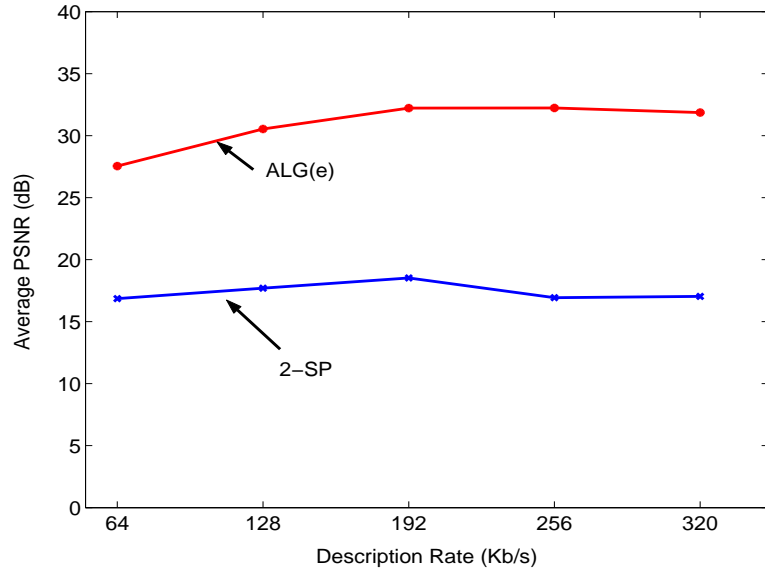


Figure 5.9: Average PSNR values for different description rates.

[6, 12, 16, 71]. An H.263+ like codec is implemented to generate the two descriptions. This codec encodes the video sequence into two balanced descriptions (i.e., $R_1 = R_2$). The QCIF sequence “Foreman” (400 frames) is encoded at 15 fps for each description. A 10% macroblock level intra-refreshment is used. Each Group of Blocks (GOB) is carried in a different packet. When a GOB is corrupted, the decoder applies a simple error concealment scheme by copying the corresponding slice from the most recent, correctly received frame.

Table 5.4 lists the average PSNR performance achieved using our ALG(ϵ) and 2-SP for 4 network sizes (20, 30, 50 and 100 nodes) with $\epsilon = 0.01$ and $R = 320$ Kb/s. We find that the average PSNR values obtained using ALG(ϵ) are much higher than those obtained using 2-SP algorithm. This is consistent with the results obtained for distortion comparison shown earlier.

We now vary the rate of each video description from 64 Kb/s to 320 Kb/s for the 50-node network and compare the PSNR performance under our cross-layer approach and 2-SP. This result is shown in Figure 5.9.

Note that as the description rate R increases, more links will become ineligible during

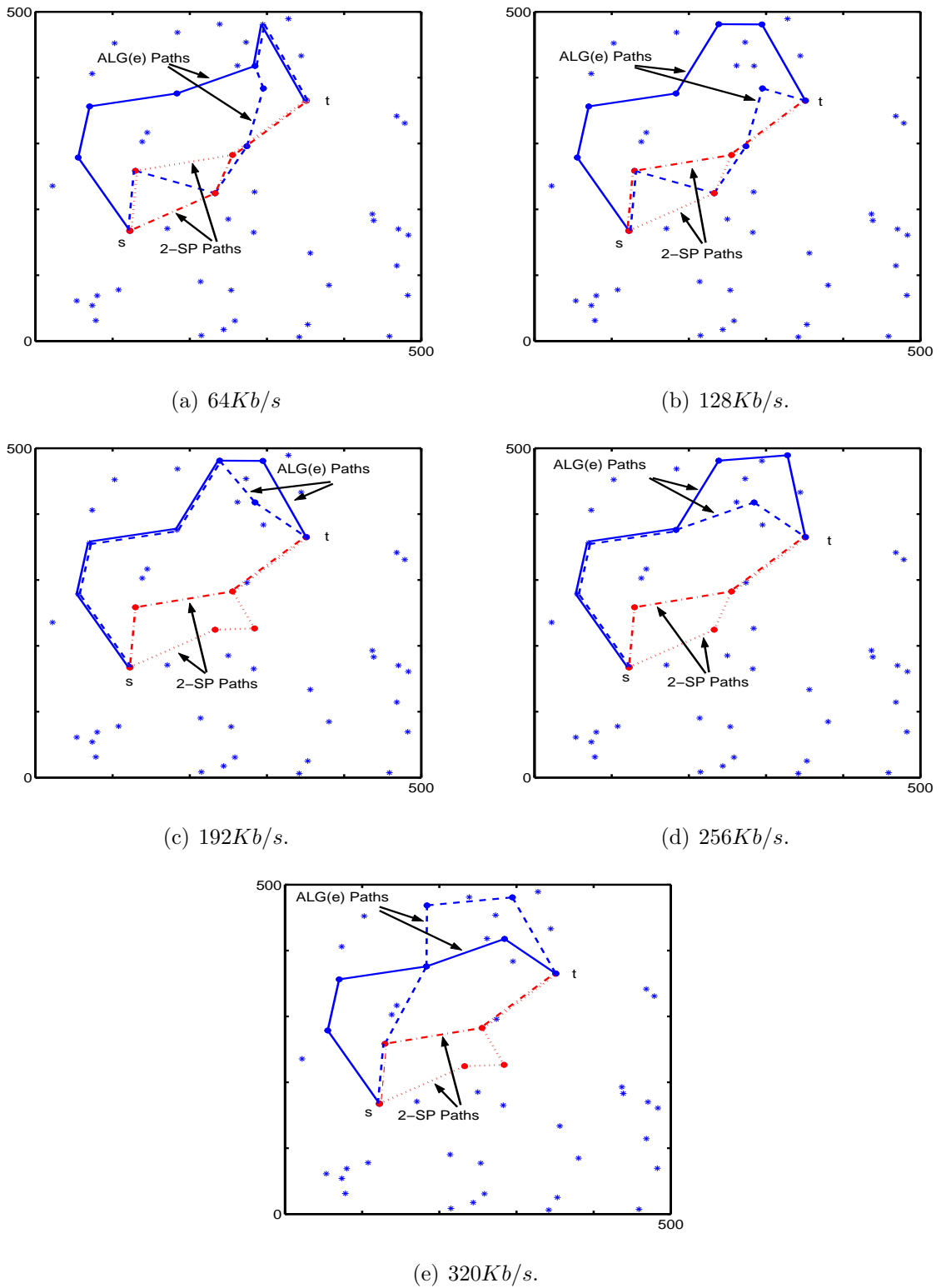


Figure 5.10: Examples of paths obtained by $ALG(\epsilon)$ and 2-SP for different description rates.

path selection process. Again, we find that our cross-layer approach provides higher PSNR over the 2-SP approach under all description rates. In Figure 5.10, we use specific example to illustrate the difference in paths chosen by our cross-layer approach and the 2-SP routing algorithm.

To illustrate the quality of video frames, we plot a sample video frame from the the original video in Figure 5.11(a) and compare it to the reconstructed video frames under our $\text{ALG}(\epsilon)$ and 2-SP for $R = 128$ Kb/s and $R = 256$ Kb/s, respectively in Figure 5.11(b)-(e). The frames under $\text{ALG}(\epsilon)$ have a visual quality very close to the loss-free frame, while the frames under 2-SP are barely recognizable.

5.8 Related Work

Multipath routing has been a topic of active research over the years. For example, various polynomial time algorithms have been proposed to compute k -shortest paths [31]. Other important works include, node- or link-disjoint paths [82, 96], and braided multiple paths [74]. There exists many routing protocols, even in the wireless domain that are capable of multipath routing [118]. However, most of these algorithms do not explicitly consider optimizing performance at the application layer. It is not hard to see that these routing algorithm may not yield optimal performance for video applications.

Cross-layer design for wireless networks has also received much attention in the context of data and video applications [25, 28, 63]. Typically, such designs consider jointly optimizing two or more of the following: power control, MAC, routing, scheduling and source coding, in order to make use of the network resources efficiently. Theoretical optimization formulations tend to benefit from some of the commonly used assumptions that the objective function is strictly concave, non-decreasing and continuously differentiable [63]. However, in this chapter, we show that when modeling the user perceived video quality as a function of network parameters, the objective function is considerably more complex and does not follow



(a) Original.

(b) ALG(ϵ), 128Kb/s.

(c) 2-SP, 128Kb/s.

(d) ALG(ϵ), 256Kb/s.

(e) 2-SP, 256Kb/s.

Figure 5.11: Frame 278 from the reconstructed video sequences ($R = 128Kb/s, 256Kb/s$).

these earlier assumptions.

The problem of path selection for MD video has recently been explored in [6, 12, 66]. In [6], Apostolopoulos *et al.* presented several MD surrogate selection algorithms to provide guidelines on selecting double description (DD) video servers in a content delivery network. In this work, although DD servers are selected such that video distortion is minimized, the problem of finding optimal paths to the servers has not been explicitly addressed. In [12], Begen *et al.* studied the problem of path selection for DD video in the context of overlay networks, where path selection is formulated as an optimization problem that minimizes video distortion. The problem is solved by an exhaustive search over the exponential solution space.

5.9 Summary

In this chapter, we studied the problem of how to route MD video over multi-hop wireless networks with the objective of optimizing the application layer performance. We formulated this problem into an cross-layer optimization problem with an application performance metric as the objective function and routing and link layer considerations as constraints. We developed a formal RLT-based branch-and-bound solution procedure and showed that this solution procedure is able to produce a set of routes whose objective is within ϵ from the optimal objective function value. Simulation results demonstrated the efficacy of the proposed solution procedure. The results in this chapter contribute to the theoretical foundation of complex cross-layer problems associated with video communications in wireless networks.

Chapter 6

Implementation Considerations

In this chapter, we discuss how our proposed cross-layer routing algorithms can be implemented in a multihop wireless networks. At the highest level, each of the proposed algorithms is a link-state algorithm (like Dijkstra’s algorithm) that requires topology and link statistics information. Computation to find a set of paths (one or more for each source-destination pair) can then be performed at a source node in order to transport the video sessions to the destination. Once the set of paths are computed, source routing (i.e., specification of nodes along the path in the packet header) can be used to forward the video packets for each video description.

6.1 General Approach

Existing routing protocols can be roughly categorized as *proactive*, whereby a consistent and up-to-date view of the network topology is always maintained, and *reactive*, whereby route discovery is performed on-demand. We believe that our proposed cross-layer routing algorithms are most suitable to be implemented within the proactive routing paradigm. Our choice is motivated by the following considerations. First, it is necessary to make

quick routing decisions whenever a new video request arrives. The readily available route information under a proactive paradigm is well suited for this purpose, which can reduce session initiation delay for real-time multimedia applications. Second, for many applications (e.g., search and rescue), it is highly desirable to maintain an accurate network topology and link state information at an ad hoc node for administrative purposes. In this sense, a link state protocol can readily provide such information.

Proactive link state routing protocols are quite successful in IETF standardization. In fact, two of the three existing RFCs for MANET routing, i.e., OLSR [23] and TBRPF [79], follow the link state paradigm. It has been shown that the overhead associated with maintaining a link state database can be effectively minimized by either using Multipoint Relays (MPR), as in OLSR, or by reporting partial topology information in the LSAs and using “differential” HELLO messages that report only changes in neighbor status, as in TBRPF. Our initial distributed implementation experience [110] shows that, with proper design extension and changes, proactive protocols can be used to build and maintain a link state database for developing a distributed implementation for the various cross-layer routing algorithms proposed in this dissertation.

Since an effective cross-layer routing operation requires the knowledge of a set of end-to-end paths, at the core of distributed implementation are efficient means to build and maintain network topology and link statistics databases at each node. We find that we can develop such a distributed implementation by building it on top of an existing MANET routing protocol framework such as OLSR. Once the topology information is available, we can simply replace the link state routing engine (shortest path routing) in OLSR with our proposed cross-layer algorithms, either using the GA-based approach as discussed in Chapter 3 or the specialized branch-and-bound based algorithms discussed in Chapters 4 and 5.

Once the paths are computed, the next question is how to establish these paths for a video session. This could be done with *source routing*, in which each data packet carries the entire path in the packet header [52]. Each intermediate node forwards a packet to the next

hop node based on the path information carried in the header. With source routing, there is no need to maintain routing tables at intermediate nodes. This approach is particularly suitable for small and medium-sized networks. When network size increases, the overhead incurred by carrying paths in the packet could be high. Consequently, a technique such as *soft flow states* [48] can be used to minimize such overhead in the packet header. With soft flow states, only the first packet contains the full path information. As the first packet travels from source to destination, the flow state mechanism allows each intermediate node to record the address of the next hop along this source route. Subsequent packets from the same flow may then be forwarded along the same route without the need to carry the same source routing information in the packet. Such per-flow state will be refreshed by a new packet belonging to the same flow, and will expire after a timeout period.

6.2 Wireless Node Architecture

Before we delve into the specifics of distributed routing protocol, we present a node architecture for possible implementation, as shown in Figure 6.1. We briefly describe the key modules in the node architecture.

6.2.1 Topology Database

As with any link state routing protocol such as OLSR (see Section 6.3.1), our implementation distributes network topology information by periodically broadcasting link state advertisements (LSA) to the entire network. In order to reduce the control traffic overhead, we shall use a multipoint relaying technique [23], which is known to minimize the flooding of control messages. An LSA is periodically sent by each node to declare its MPR selector set, i.e., the list of neighbors who have selected the sender as their multipoint relay, along with any relevant addressing information. This information will help each node in the MANET to build its topology database.

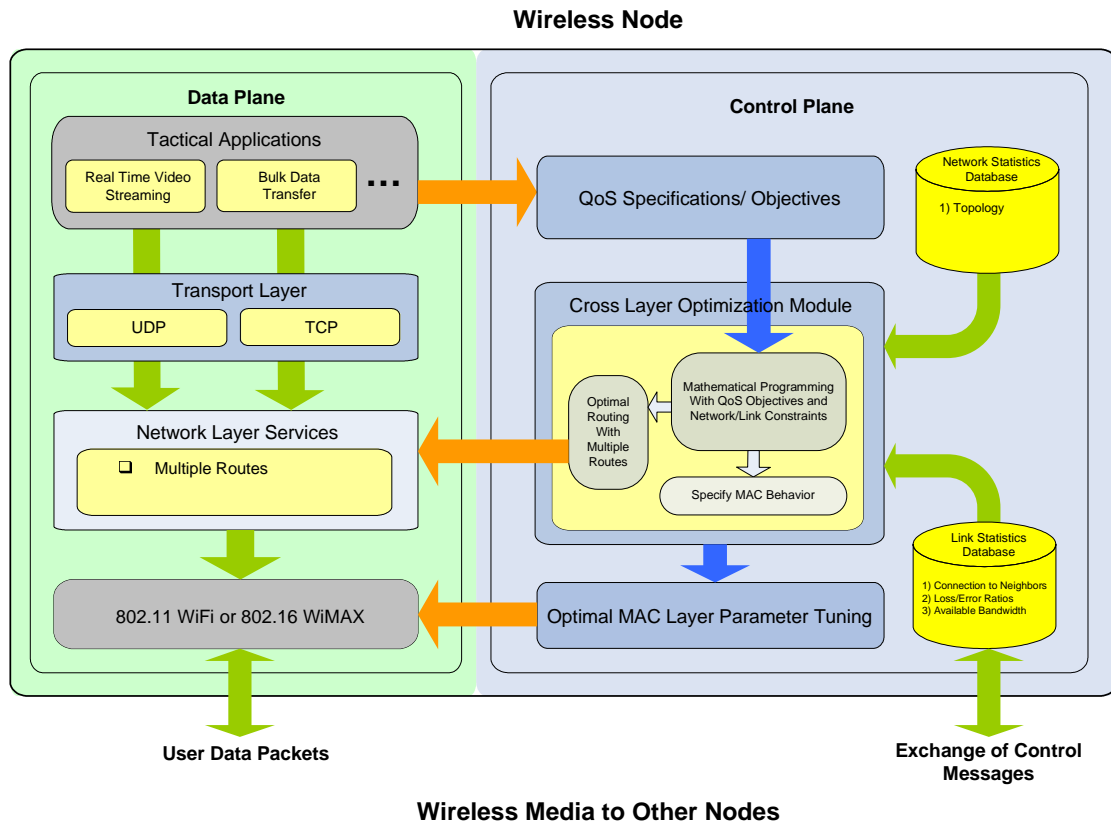


Figure 6.1: Wireless node architecture.

6.2.2 Link Statistics Database

Each node detects its neighbor nodes by periodically broadcasting HELLO messages containing information about its neighbors and their link status. Each node continuously measures the link metrics, such as bandwidth, loss rate, and delay. Several effective algorithms for such measurements, e.g., those proposed in QOLSR [8], can be used for this purpose. When a node receives an LSA, it examines the message and extracts new link state information. Then, newly-learned links, as well as their metrics, are pooled in a link state database, which contains the learned topology of the wireless ad hoc network and the metrics of every learned link. Using a sequence number, we make sure that a stale item is always overwritten by a newer item, making the link state database up-to-date.

6.2.3 Cross Layer Optimization Module

The applications in the data plane are associated with specific QoS requirements. When an application, say, video streaming, is initiated by the user at a particular node, these QoS requirements are fed into the cross-layer optimization module in the control plane. The routing engine in the control plane utilizes the topological and statistical information that is maintained by the node in the various databases, and computes optimal routes for the video application. This routing information is then fed back to the network layer services in order to establish the routes. Once the set of paths are found, source routing is employed to establish the routes and forward video packets related to the specific application.

When a node is neither a source nor a destination in a MANET, it simply acts as a relay node by forwarding the video/data packets to the next hop node in the path.

6.3 Routing Protocol Implementation

We propose to build our cross-layer routing algorithms on top of the so-called *proactive* routing protocols, such as *Optimized Link State Routing* (OLSR) protocol [23]. Due to its proactive nature, OLSR maintains an up-to-date global network topology information in its link-state database. For QoS routing, each node locally measures the link statistics and distributes them by controlled flooding of LSAs. In OLSR, such messages are called Topology Control (TC) messages [8]. In this section we present a brief introduction to OLSR, and explain how we can take advantage of some of the proven ideas in OLSR, and develop a distributed routing protocol based on the cross-layer routing concepts proposed in this dissertation.

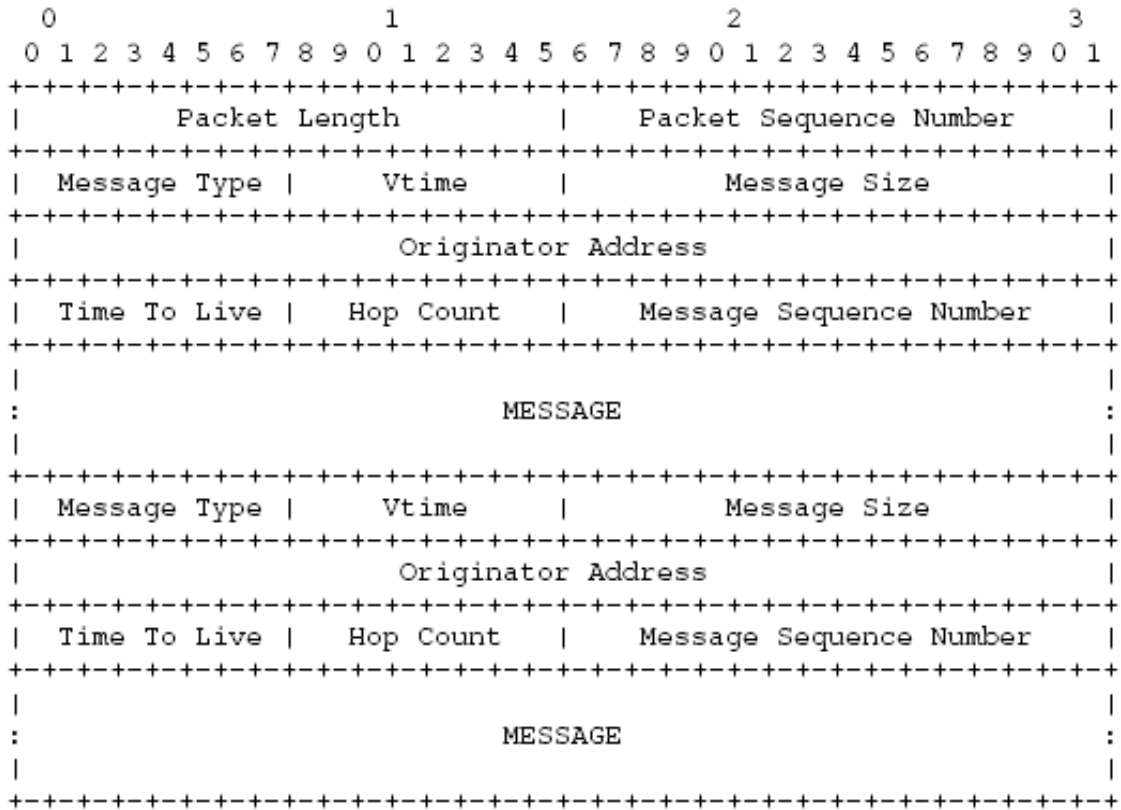


Figure 6.2: OLSR packet header.

6.3.1 OLSR

OLSR is a table-driven proactive link state routing protocol developed for MANET. It employs periodic exchange of messages to maintain topology information of the network in various databases at each node. Specifically, OLSR uses the following databases: Multiple Interface Association Information Base, Link Set, Neighbor Set, 2-hop Neighbor Set, MPR Set, MPR Selector Set, Topology Information Base and Duplicate set. These databases are registered with timeouts to determine the validity of the information. Figure 6.2 shows the various fields in an OLSR packet along with their field lengths.

OLSR uses an optimized technique called multipoint relaying to preserve bandwidth and to disseminate network topology information, by reducing duplicate retransmissions while forwarding a broadcast packet. Only node(s) selected as MPR forwards control traffic,

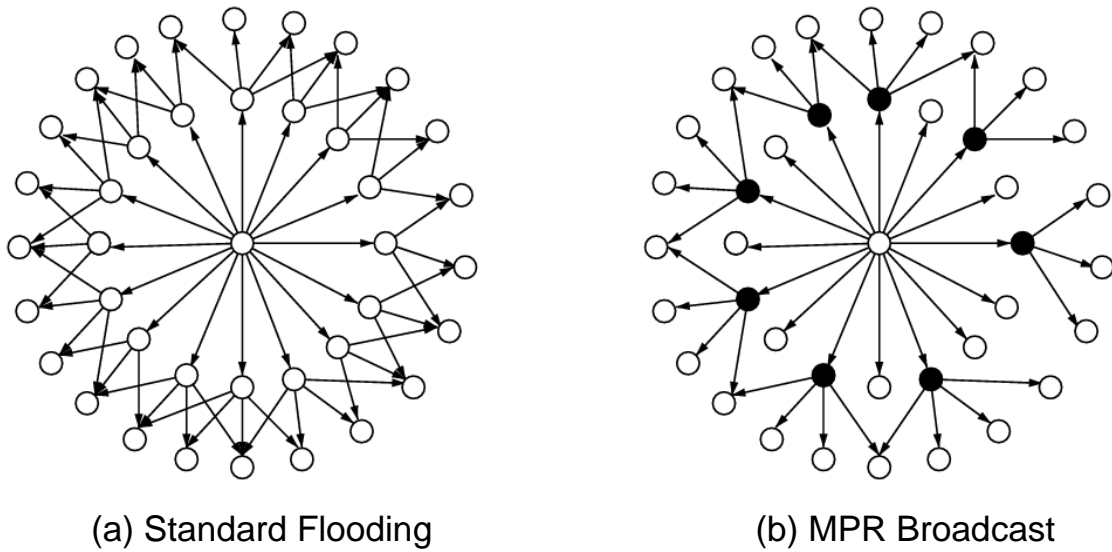


Figure 6.3: Comparison of standard flooding and MPR broadcast.

thereby reducing the number of transmissions required. This way, control messages are flooded efficiently by the network of MPRs. Multipoint relays are selected among the one hop neighbors with a bi-directional link. Therefore, selecting a route from a source to destination through multipoint relays automatically avoids the problems associated with packet transfer on uni-directional links (such as receiving acknowledgments).

Every node selects its own set of MPRs from a subset of its symmetric neighbors so that all two-hop neighbors can be reached through an MPR. Nodes announce their willingness to act as MPRs for its neighbor. Through the use of HELLO messages, such willingness is spread and considered during the MPR selection phase. Figure 6.3(b) shows how such MPRs are selected around a node. When compared with standard flooding, as shown in Figure 6.3(a), it can be seen that by selecting MPRs efficiently, the flooding of the control messages can be greatly reduced. A jitter is introduced in message forwarding to avoid radio collisions due to synchronized forwarding. Once selected, MPRs declare their status periodically in control messages along with the link state information of the neighbors. This way, the current network topology is maintained at each node in the MANET.

HELLO, TC (Topology Control), and MID (Multiple Interface Declaration) are the three

basic types of control messages defined by OLSR. HELLO messages are sent on regular intervals to detect all one hop neighbors to achieve link sensing, neighbor sensing, two-hop neighbor sensing and MPR selector sensing. These control messages are transmitted to all one hop neighbors in broadcast mode, but are not relayed any further. Topology messages (TC) are flooded on regular intervals using MPR optimization to the network with information about a node's local links. A TC message is also generated when changes are detected in the MPR selection set. A sequence number is associated with each packet in the packet header in order to avoid loops. That is, a packet is not retransmitted if its sequence number is less than or equal to the last registered packet from the sender.

6.4 Summary

In this chapter, we presented an implementation architecture for our cross-layer routing algorithms, by leveraging some of the proven ideas of an existing proactive MANET routing protocol framework like OLSR. Since the topology information is readily available at each node, we can simply replace the link state routing engine (shortest path routing) in OLSR with our proposed cross-layer algorithms. We also presented a wireless node architecture that facilitates our cross-layer optimized routing implementation.

Chapter 7

Summary and Future Work

7.1 Summary

Multihop wireless networks offer great potential in situations where traditional communication infrastructure is either expensive or time consuming to deploy. The demand for such networks has been recognized not only by the military, but also for civilian applications such as wireless emergency networks, voice and video communications for first responders in disaster relief, environmental and biometric sensor networking for homeland security, to name a few.

The primary focus of this dissertation has been to identify and address key challenges that are facing video communications over multihop wireless networks. Following the review of two types of global optimization techniques, we address three important problems: concurrent routing, path selection and rate allocation, and multipath routing for multiple description video. In concurrent routing, we focused on modeling the inter-session interactions that couple the performance of an individual flow with that of other concurrent video sessions. Path selection and rate allocation refers to the problem of not only selecting the best set of paths for video communication, but also, computing the optimal rate and partition-

ing it among the chosen set of paths. In studying multipath routing for multiple description video, we show that MD coding when used in combination with multipath routing in wireless networks has tremendous advantages over traditional layered video coding techniques.

In each of the three problems, our research effort demonstrated that substantial improvement in video quality can be achieved by employing a cross-layer design approach that factors in video quality (at the application layer) into routing algorithms at the network layer. In each case, we formulated an application centric cross-layer optimization problem that minimizes video distortion as a function of network and link layer behavior.

Since video distortion is a highly complex function of *multiple* layer metrics, we recognized early on that specialized algorithms would have to be developed in order to solve such complex nonconvex problems. Therefore, we resorted to two types of global optimization techniques. We developed a GA-based solution approach for the concurrent routing problem and showed it to be highly suitable for such complex nonconvex optimization problems. The strengths of the GA-based approach were highlighted when compared with other trajectory based metaheuristics such as simulated annealing and tabu search, as well as in comparison with traditional network centric routing schemes such as shortest path and disjoint shortest path routing. However, the quest for finding an optimal solution led us to develop a specialized branch-and-bound algorithm for solving the complex cross-layer problems to global optimality. This was achieved by reformulating the nonconvex problems, constructing tight linear programming relaxations by using logarithmic convexification and RLT-based inequalities, and then embedding this RLT construct into a branch-and-bound algorithm. The nonlinear optimization techniques developed in this dissertation can be used to solve a broad class of cross-layer problems, and we hope that the investigations presented in this dissertation opens doors for future cross-layer research in multihop wireless networks

Finally we presented a distributed implementation architecture for our proposed cross-layer routing algorithms. We described the key modules and discussed efficient means to build and maintain network topology and link statistics information at every node in the

network. Our strategy is to take advantage of some of the proven insights of proactive routing protocols such as OLSR, and show that with proper design extension such as source routing and other changes, we can adapt the existing proactive routing paradigm into incorporating our cross-layer designs.

In conclusion, the unique and demanding characteristics of multihop wireless networks call for design changes that look beyond the conventional layering philosophy. As described in this dissertation, large gains could be achieved when cross-layer strategies are incorporated. This cross-layer design principle shows a great deal of promise, and our research will have an impact on video communications over multihop wireless networks.

7.2 Future Research Direction

As it stands, the problem of multipath routing for multiple description video is formulated based on only a single source-destination pair. Since the importance of concurrent routing has already been established, we plan to extend the analysis as well as the solution algorithm to include video communication for multiple source-destination pairs. This would add significant complexity to the problem formulation, and increase the number of binary variables by an order of magnitude. Therefore a specialized solution procedure might have to be developed in order to solve this extended problem. This specialized algorithm would need to take advantage of any special structure that the new problem might reveal.

Another research direction is to extend the cross-layer routing algorithms for video communication presented in this dissertation to incorporate the idea of performing routing decisions in the presence of inaccurate or uncertain (stale) topology or link statistics information. For example, uncertainties may arise in the parameters advertised by the links, such as bandwidth, path loss probability etc., This is even more true in highly dynamic networks where changes are frequent. Our goal in this situation would be to identify paths that are *most likely* to minimize the application layer performance objective and satisfy the QoS requirements.

With the advent of software-defined radio technology [87] and the inadequacy of FCC's current fixed bandwidth policy, future multihop wireless networks would have to be spectrum agile. Recent FCC studies show that most of the fixed allocated spectrum blocks are used sparsely in the spatio-temporal domain, while the remaining useful spectrum available for newer wireless services is being depleted. These observations have prompted a *dynamic spectrum access* approach, where wireless devices are allowed to sense and identify currently unused spectrum blocks for communication, without causing interference to the primary allocated user. Since there is no statically allocated fixed spectrum for use in shared spectrum networks, each node must individually detect the *white bands* (allocated spectrum blocks that are not currently in use) for communication, and there may not exist a common frequency band shared by all the nodes in the network. Video communication over such shared spectrum networks brings with it a completely new set of challenges. For example, when interference power exceeds certain threshold (as defined based on the specific spectrum block as well as the specific application), the video application degrades in performance. When this degradation becomes significant (referred to as *outage*), one of two corrective measures can be taken. Either the network layer reacts by identifying an alternate path for video, or the physical layer can try to restore communication by identifying an alternate spectrum block to transmit over. One of our future research directions is to focus on theory, algorithmic development and implementation to understand and mitigate such challenges associated with enabling video communications over multihop wireless networks with programmable radios.

Appendix A

Other Metaheuristic Algorithms

Trajectory methods refer to the class of metaheuristics where a single solution is maintained throughout the iterative search process. They are called trajectory methods because the search process performed by these methods is characterized by a trajectory in the search space. Representative trajectory methods include *Basic Local Search* (LS), *Simulated Annealing* (SA), *Tabu Search* (TS), *Greedy Randomized Adaptive Search Procedure* (GRASP) and *Iterated Local Search*. All of these trajectory schemes are variations of local search (LS) methods, but they incorporate certain forms of diversification or randomization mechanisms to overcome the weakness of LS. For example, SA adopts a strategy to accept a non-improving solution with a decreasing probability, while TS uses the memory of search history (a tabu list) to guide the search process to avoid being trapped in local minima.

An important issue for trajectory methods is how to define a neighborhood structure, i.e., how to perturb the current solution to create a new solution in the neighborhood. We use the following neighborhood structure in the implementations of SA and TS. First, we randomly choose a path from the current solution $\bar{x} = [\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3]$, say, \mathcal{P}_1 . Then, we randomly pick a link $\{g_k, g_{k+1}\} \in \mathcal{P}_1$, and use a random construction method to find an alternative path (more than one hop) from g_k to g_{k+1} , say, g_k, g_r, g_{k+1} . Thus, a new path \mathcal{P}'_1 is created by replacing link $\{g_k, g_{k+1}\}$ in the original \mathcal{P}_1 , and the new solution is $[\mathcal{P}'_1, \mathcal{P}_2, \mathcal{P}_3]$. This process

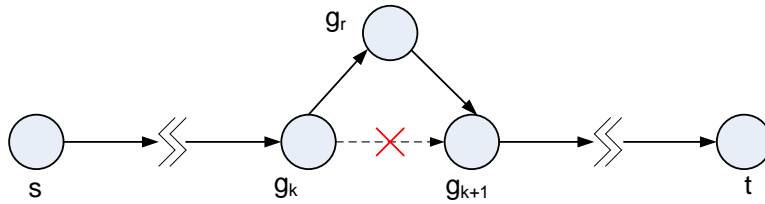


Figure A.1: Neighborhood structure used in SA and TS implementation.

is illustrated in figure A.1.

A.1 Simulated Annealing

Simulated Annealing was developed in the 1980's to deal with highly nonlinear problems. It is motivated by the way in which a metal cools and freezes into a minimum energy crystalline structure (annealing process) [1]. When SA explores the solution space, it accepts a non-improving solution with a probability, which decreases with iterations. This probabilistic acceptance function is inspired by the Boltzmann distribution from statistical mechanics and is shown here:

$$Pr\{\bar{x} \leftarrow \hat{x}\} = \begin{cases} 1, & \text{if } D(\hat{x}) < D(\bar{x}) \\ \exp\left\{-\frac{|D(\hat{x})-D(\bar{x})|}{c_k}\right\}, & \text{otherwise,} \end{cases} \quad (\text{A.1})$$

where c_k is a control parameter analogous to temperature in a physical system, \bar{x} is the current solution, and \hat{x} is a perturbation of \bar{x} (using the neighborhood structure defined before). The fashion in which c_k is manipulated is called the *cooling schedule*. The following cooling schedule is used in our experiments [1]:

1. $c_0 = 1$: i.e., nearly all transitions will be accepted at the beginning of the search process;
2. $c_{k+1} = \omega \cdot c_k$: i.e., the control parameter is decremented every time when a non-improving solution is accepted, and remains at each value for a sufficient time for the system to “return to an equilibrium.” ω is the decay coefficient.

Algorithm SA

```

 $\bar{x} \leftarrow \text{GenerateInitialSolution}()$ 
 $c \leftarrow c_0$ 
while termination conditions not met do
   $\hat{x} \leftarrow \text{PickAtRandom}(\mathcal{N}(\bar{x}))$ 
  if  $D(\hat{x}) < D(\bar{x})$  then  $\bar{x} \leftarrow \hat{x}$ 
  else Accept  $\hat{x}$  with probability given in A.1
  endif
  Update  $c$  according to the cooling schedule
endwhile

```

Figure A.2: Simulated Annealing algorithm

Algorithm TS

```

 $\bar{x} \leftarrow \text{GenerateInitialSolution}()$ 
 $TabuList \leftarrow \emptyset$ 
while termination conditions not met do
   $\hat{x} \leftarrow \text{ChooseBestFrom}(\mathcal{N}(\bar{x}) \setminus TabuList)$ 
  Update  $TabuList$ 
endwhile

```

Figure A.3: Tabu Search algorithm

The algorithm for SA is given in Figure A.2, where $\mathcal{N}(\bar{x})$ denotes the neighborhood of solution \bar{x} .

A.2 Tabu Search

Compared with SA, Tabu Search (see Figure A.3) explicitly uses the history of the search, both to escape from local minima and to implement an exploratory strategy. Specifically, TS uses a *tabu list*, implemented using a First-In-First-Out queue, to prevent from returning to recently visited solutions, therefore avoiding endless cycling and possibly forcing the search process to accept even non-improving solutions [40]. An explored solution is always inserted at the tail of the queue, while if the queue is full, the head of the queue is removed.

Bibliography

- [1] E. Aarts and J. Korst, *Simulated Annealing and Boltzman Machines*. New York, NY: John Wiley & Sons, 1989.
- [2] C. W. Ahn and R. S. Ramakrishna, “A genetic algorithm for shortest path routing problem and the sizing of populations,” *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 6, pp. 566 – 579, Dec. 2002.
- [3] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin, “Multiple description coding in networks with congestion problem,” *IEEE Trans. on Information Theory*, vol. 47, no. 3, pp. 891 – 902, Mar. 2001.
- [4] J. G. Apostolopoulos, “Reliable video communication over lossy packet networks using multiple state encoding and path diversity,” in *Proc. SPIE Visual Communications and Image Processing*, Seattle, WA, Jan. 2001, pp. 399 – 409.
- [5] J. G. Apostolopoulos and S. J. Wee, “Unbalanced multiple description video communication using path diversity,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, Thessaloniki, Greece, Oct. 2001, pp. 966 – 969.
- [6] J. G. Apostolopoulos, T. Wong, W. Tan, and S. Wee, “On multiple description streaming in content delivery networks,” in *Proc. IEEE INFOCOM*, New York, NY, June 2002, pp. 1736 – 1745.

- [7] T. Back, D. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. New York, NY: Oxford University Press, 1997.
- [8] H. Badis and K. A. Agha, "Quality of service for ad hoc optimized link state routing protocol (QOLSR)," Mar. 2006, IETF Internet Draft,. [Online]. Available: <http://www.ietf.org/internet-drafts/draft-badis-manet-qolsr-03.txt>
- [9] H. Badis and K. A. Agha, "QOLSR, QoS routing for ad hoc wireless networks using OLSR," *European Trans. on Telecommunications*, vol. 15, no. 4, 2005.
- [10] N. Banerjee and S. K. Das, "Fast determination of QoS-based multicast routes in wireless networks using genetic algorithm," in *Proc. IEEE International Conference on Communications (ICC)*, Helsinki, Finland, June 2001, pp. 2588 – 2596.
- [11] A. C. Begen, Y. Altunbasak, and O. Ergun, "Fast heuristics for multi-path selection for multiple description encoded video streaming," in *Proc. IEEE International Conference on Multimedia & Expo (ICME)*, Baltimore, MD, July 2003, pp. 517 – 520.
- [12] A. C. Begen, Y. Altunbasak, and O. Ergun, "Multi-path selection for multiple description encoded video streaming," *EURASIP Signal Processing: Image Communications*, vol. 20, no. 1, pp. 39 – 60, Jan. 2005.
- [13] D. Bertsekas and R. Gallager, *Data Networks*. Upper Saddle River, NJ: Prentice Hall, 1992.
- [14] S. Blake, D. Black, M. Carlson, E. Davies, and Z. Wang, "An architecture for differentiated services," Dec. 1998, IETF RFC 2475. [Online]. Available: <http://www.ietf.org/rfc/rfc2475.txt>
- [15] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. Data Compression Conference (DCC)*, Snowbird, Utah, Mar. 2003, pp. 203 – 212.

- [16] J. Chakareski, S. Han, and B. Giro, “Layered coding vs. multiple descriptions for video streaming over multiple paths,” in *Proc. ACM Multimedia*, Berkeley, CA, Nov. 2003, pp. 422 – 431.
- [17] C. T. Chang and C. C. Chang, “A linearization method for mixed 0-1 polynomial programs,” *Computers and Operations Research*, vol. 27, no. 10, pp. 1005 – 1016, Sep. 2000.
- [18] S. Chen and K. Nahrstedt, “Distributed quality-of-service routing in ad-hoc networks,” *IEEE J. on Selected Areas in Communications*, vol. 17, no. 8, pp. 1488 – 1505, Aug. 1999.
- [19] H. Chernoff, “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations,” *Annals Math. Statist.*, vol. 23, pp. 493 – 507, 1952.
- [20] P. A. Chou and Z. Miao, “Rate-distortion optimized sender-driven streaming over best-effort networks,” in *Proc. Workshop on Multimedia Signal Processing (MMSP)*, Cannes, France, Oct. 2001, pp. 587 – 592.
- [21] P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” *IEEE Trans. on Multimedia*, vol. 8, no. 2, pp. 390 – 404, Apr. 2006.
- [22] D. Chung and Y. Wang, “Multiple description coding using pairwise correlating transforms,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, pp. 895 – 908, Sep. 1999.
- [23] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol,” Oct. 2003, IETF RFC 3626. [Online]. Available: <http://www.ietf.org/rfc/rfc3626.txt>
- [24] T. H. Cormen, C. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, Massachusetts; London, England: The MIT Press, 1990.

- [25] R. L. Cruz and A. V. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar 2003, pp. 702 – 711.
- [26] P. de Cuetos and K. W. Ross, "Optimal streaming of layered video: Joint scheduling and error concealment," in *Proc. ACM Multimedia*, Berkeley, CA, Nov. 2003, pp. 55 – 64.
- [27] K. DeJong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan, 1975.
- [28] T. ElBattand and A. Ephremides, "Joint scheduling and power control for wireless ad hoc networks," *IEEE Trans. on Wireless Communications*, vol. 3, no. 1, pp. 74 – 85, Jan. 2004.
- [29] R. Elbaum and M. Sidi, "Topological design of local-area networks using genetic algorithms," *IEEE/ACM Trans. on Networking*, vol. 4, no. 5, pp. 766 – 778, Oct. 1996.
- [30] A. Elwalid, D. Heyman, T. V. Lakshman, D. Mitra, and A. Weiss, "Fundamental bounds and approximations for atm multiplexers with applications to video teleconferencing," *IEEE J. on Selected Areas in Communications*, vol. 13, no. 6, pp. 953 – 962, Aug. 1995.
- [31] D. Eppstein, "Finding the k shortest paths," *SIAM J. on Computing*, vol. 28, no. 3, pp. 652 – 673, Aug. 1999.
- [32] G. D. Fatta, F. Hoffmann, G. L. Re, and A. Urso, "A genetic algorithm for the design of a fuzzy controller for active queue management," *IEEE Trans. Systems, Man and Cybernetics*, vol. 33, no. 3, pp. 313 – 324, Aug. 2003.
- [33] M. Fleming and M. Effros, "Generalized multiple description vector quantization," in *Proc. Data Compression Conference (DCC)*, Snowbird, UT, Mar. 1999, pp. 3 – 12.

- [34] D. B. Fritchman, "A binary channel characterization using partitioned markov chains," *IEEE Trans. on Information Theory*, vol. 13, no. 2, pp. 221 – 227, Apr. 1967.
- [35] M. R. Garey and D. S. Johnson, *Computers and Intractability – A Guide to the Theory of NP-Completeness*. New York, NY: W.H. Freeman, 1979.
- [36] M. Gen and R. Cheng, *Genetic Algorithms & Engineering Optimization*. New York, NY: John Wiley & Sons, Inc., 2000.
- [37] M. Ghanbari, "Two-layer coding of video signals for vbr networks," *IEEE J. on Selected Areas in Communications*, vol. 7, pp. 771 – 781, June 1989.
- [38] H. Gharavi and K. Ban, "Dynamic adjustment packet control for video communications over ad-hoc networks," in *Proc. IEEE International Conference on Communications (ICC)*, Paris, France, June 2004, pp. 3086 – 3090.
- [39] S. C. Ghosh, B. P. Sinha, and N. Das, "Channel assignment using genetic algorithm based on geometric symmetry," *IEEE Trans. on Vehicular Technology*, vol. 52, no. 4, pp. 860 – 875, July 2003.
- [40] F. Glover and M. Laguna, *Tabu Search*. Boston, MA: Kluwer-Academic, 1997.
- [41] N. Gogate, D. Chung, S. Panwar, and Y. Wang, "Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 9, pp. 777 – 792, Sep. 2002.
- [42] D. E. Goldberg, Ed., *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison - Wesley Publishing Co., 1989.
- [43] C. Goldie and C. Klüppelberg, *Subexponential Distributions*. Basel, Switzerland: Birkhäuser Publishing Ltd., 1998, pp. 435 – 459.

- [44] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, pp. 74 – 93, Sep. 2001.
- [45] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, "Multi-path TCP: A joint congestion control and routing scheme to exploit path diversity on the internet," *IEEE/ACM Trans. on Networking*, 2007, to appear.
- [46] J. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1975.
- [47] U. Horn, K. Stuhlmuller, M. Link, , and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communication*, vol. 15, no. 1-2, pp. 77 – 94, Sep. 1999.
- [48] Y.-C. Hu and D. B. Johnson, "Design and demonstration of live audio and video over multihop wireless ad hoc networks," in *Proc. IEEE MILCOM*, Anaheim, CA, Oct. 2002, pp. 7 – 10.
- [49] "Information technology-coding of audio-visual objects: Visual(MPEG-4)," ISO/IEC, Mar. 1998, Final Committee Draft 14496-2, JTC1/SC29/WG11.
- [50] H. Jafarkhani and V. Tarokh, "Multiple description trellis coded quantization," *IEEE Trans. on Communications*, vol. 47, pp. 799 – 803, June 1999.
- [51] W. Jiang and A. Ortega, "Multiple description speech coding for robust communication over lossy packet networks," in *Proc. IEEE International Conference on Multimedia & Expo (ICME)*, New York, NY, Aug. 2000, pp. 444 – 447.
- [52] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," Apr. 2003, IETF Internet Draft. [Online]. Available: <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt>
- [53] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross layer design," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 3 – 11, Feb. 2005.

- [54] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, “Rate control in communication networks: Shadow prices, proportional fairness and stability,” *J. Operational Research Society*, vol. 49, no. 3, pp. 237 – 252, Mar. 1998.
- [55] M. Khansari, A. Zakauddin, W.-Y. Chan, E. Dubois, and P. Mermelstein, “Approaches to layered coding for dual-rate wireless video transmission,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 1, Austin, TX, Oct. 1994, pp. 258 – 262.
- [56] B.-J. Kim, Z. Xiong, and W. A. Pearlman, “Low bit-rate scalable video coding with 3d set partitioning in hierarchical trees (3-D SPIHT),” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374 – 1387, Dec. 2000.
- [57] C. Kim and S. Lee, “Multiple description motion coding algorithm for robust video transmission,” in *Proc. IEEE International Symposium on Circuits and Systems (IS-CAS)*, vol. 4, Geneva, Switzerland, May 2000, pp. 717 – 720.
- [58] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, “Optimal rate control for video transport over multi-hop wireless networks,” in *Proc. IEEE Wireless Communications & Networking Conference (WCNC)*, Las Vegas, NV, Apr. 2006.
- [59] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, “Cross-layer optimized routing for md video in multi-hop wireless networks,” *IEEE J. on Selected Areas in Communications*, submitted for publication.
- [60] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, “On path selection and rate allocation for video in wireless mesh networks,” *IEEE Trans. on Networking*, submitted for publication.
- [61] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, “Optimal multipath routing for application performance guarantees in multi-hop wireless networks,” in *Proc. IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS)*, submitted for publication.

- [62] C. R. Lin and J.-S. Liu, "QoS routing in ad hoc wireless networks," *IEEE J. on Selected Areas in Communications*, vol. 17, no. 8, pp. 1426 – 1438, Aug. 1999.
- [63] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multi-path routing," Purdue University, Tech. Rep., 2004.
- [64] W. Liu and Y. Fang, "Courtesy piggybacking: Supporting differentiated services in multihop mobile ad hoc networks," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, pp. 1273 – 1283.
- [65] N. Malpani and J. Chen, "A note on practical construction of maximum bandwidth paths," *Information Processing Letters*, vol. 83, pp. 175 – 180, Aug. 2002.
- [66] S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar, "MRTP: a multiflow real-time transport protocol for ad hoc networks," *IEEE Trans. on Multimedia*, vol. 8, no. 2, pp. 356 – 369, Apr. 2006.
- [67] S. Mao, Y. T. Hou, X. Cheng, H. D. Sherali, and S. F. Midkiff, "Multi-path routing for multiple description video over wireless ad hoc networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 740 – 750.
- [68] S. Mao, S. Kompella, Y. T. Hou, and S. F. Midkiff, "A fast greedy algorithm for routing concurrent video flows," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Kobe, Japan, May 2005, pp. 3535 – 3538.
- [69] S. Mao, S. Kompella, Y. T. Hou, and H. D. Sherali, "Routing for multiple concurrent video sessions in wireless ad hoc networks," in *Proc. IEEE International Conference on Communications (ICC)*, vol. 2, Seoul, Korea, May 2005, pp. 1229 – 1235.
- [70] S. Mao, S. Kompella, Y. T. Hou, H. D. Sherali, and S. F. Midkiff, "Routing for concurrent video sessions in ad hoc networks," *IEEE Trans. on Vehicular Technology*, vol. 55, no. 1, pp. 317 – 327, Jan. 2006.

- [71] S. Mao, S. Lin, S. S. Panwar, Y. Wang, and E. Celebi, "Video transport over ad hoc networks: Multistream coding with multipath transport," *IEEE J. on Selected Areas in Communications*, vol. 21, no. 10, pp. 1721 – 1737, Dec. 2003.
- [72] S. Mao, S. Lin, Y. Wang, S. S. Panwar, and Y. Li, "Multipath video transport over wireless ad hoc networks," *IEEE Wireless Communications Magazine*, vol. 12, no. 4, pp. 42 – 49, Aug. 2005.
- [73] S. Mao, S. S. Panwar, and Y. T. Hou, "On optimal partitioning of realtime traffic over multiple paths," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 2325 – 2336.
- [74] S. Murthy and J. J. Garcia-Luna-Aceves, "Congestion-oriented shortest multipath routing," in *Proc. IEEE INFOCOM*, vol. 3, San Francisco, CA, May 1996, pp. 1028 – 1036.
- [75] S. Nelakuditi, Z.-L. Zhang, R. P. Tsang, and D. H. C. Du, "Adaptive proportional routing: A localized QoS routing approach," *IEEE/ACM Trans. on Networking*, vol. 10, no. 6, pp. 790 – 804, Dec. 2002.
- [76] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY: John Wiley & Sons, 1999.
- [77] C. Y. Ngo and V. O. K. Li, "Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms," *IEEE Trans. on Communications*, vol. 51, no. 9, pp. 1439 – 1441, Sep. 2003.
- [78] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE J. on Selected Areas in Communications*, vol. 13, no. 6, pp. 953 – 962, Aug. 1995.
- [79] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," Feb. 2004, IETF RFC 3684. [Online]. Available: <http://www.ietf.org/rfc/rfc3684.txt>

- [80] L. Ozarow, "On a source coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, vol. 59, no. 10, pp. 84 – 91, Dec. 1980.
- [81] E. R. Panier and A. L. Tits, "A superlinearly convergent feasible method for the solution of inequality constrained optimization problems," *SIAM J. on Control and Optimization*, vol. 25, no. 4, pp. 934 – 950, July 1987.
- [82] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path set selection in mobile ad hoc networks," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Lausanne, Switzerland, June 2002, pp. 1 – 11.
- [83] C. E. Perkins, E. M. Royer, and S. R. Das, "Quality of Service in Ad hoc On-Demand Distance Vector routing," July 2000, IETF Internet Draft, draft-ietf-manet-qos-00.txt.
- [84] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," July 2003, IETF RFC 3561. [Online]. Available: <http://www.ietf.org/rfc/rfc3561.txt>
- [85] D. Quaglia and J. C. de Martin, "Delivery of mpeg video streams with constant perceptual quality of service," in *Proc. IEEE International Conference on Multimedia & Expo (ICME)*, vol. 2, Lausanne, Switzerland, Aug. 2002, pp. 85 – 88.
- [86] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen, "Scalable internet video using MPEG-4," *Signal Processing: Image Communication*, vol. 15, no. 1-2, pp. 95 – 126, Sep. 1999.
- [87] J. H. Reed, *Software Radio: A Modern Approach to Radio Engineering*. New Jersey: Prentice Hall, May 2002.
- [88] A. Reibman, H. Jafarkhani, Y. Wang, and M. Orchard, "Multiple description video using rate-distortion splitting," in *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 1, Thessaloniki, Greece, Oct. 2001, pp. 979 – 981.

- [89] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, and R. Puri, “Multiple description coding for video using motion compensated prediction,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 3, Kobe, Japan, Oct. 1999, pp. 837 – 841.
- [90] K. W. Ross, *Multiservice loss models for broadband telecommunication networks*. New York, NY: Springer, 1995.
- [91] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, “Cross-layer design of ad hoc networks for real-time video streaming,” *Wireless Communications Magazine*, vol. 12, no. 4, pp. 59 – 65, Aug. 2005.
- [92] E. Setton, X. Zhu, and B. Girod, “Congestion based multipath routing of multimedia data over ad hoc networks,” in *Proc. IEEE International Conference on Multimedia & Expo (ICME)*, Taipei, Taiwan, June 2004, pp. 1619 – 1622.
- [93] H. D. Sherali and W. P. Adams, “A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems,” *Discrete Applied Mathematics*, vol. 52, pp. 83 – 106, 1994.
- [94] H. D. Sherali and W. P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Boston, MA: Kluwer Academic Publisher, 1999.
- [95] H. D. Sherali and W. Adams, “A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems,” *SIAM J. on Discrete Mathematics*, vol. 3, no. 3, pp. 411 – 430, 1990.
- [96] H. D. Sherali, K. Ozbay, and S. Subramanian, “The time-dependent shortest pair of disjoint paths problem: Complexity, models, and algorithms,” *Networks*, vol. 31, no. 4, pp. 259 – 272, Dec. 1998.

- [97] H. D. Sherali and C. H. Tuncbilek, "A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique," *J. of Global Optimization*, vol. 2, no. 1, pp. 101 – 112, 1992.
- [98] H. D. Sherali and C. H. Tuncbilek, "A reformulation-convexification approach for solving nonconvex quadratic programming problems," *J. of Global Optimization*, vol. 7, pp. 1 – 31, 1995.
- [99] J. Shin, J. Kim, and C.-C. J. Kuo, "Quality-of-service mapping mechanism for packet video in differentiated services network," *IEEE Trans. on Multimedia*, vol. 3, no. 2, pp. 219 – 231, June 2001.
- [100] M. C. Sinclair, "Minimum cost wavelength-path routing and wavelength allocation using a genetic-algorithm/heuristic hybrid approach," *IEE Proc. Communications*, vol. 46, no. 1, pp. 1 – 7, Feb. 1999.
- [101] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A core-extraction distributed ad hoc routing algorithm," *IEEE J. on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454 – 1465, Aug. 1999.
- [102] K. Stuhlmuller, N. Farber, , M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012 – 1032, June 2000.
- [103] W.-T. Tan and A. Zakhor, "Internet video using error resilient scalable compression and cooperative transport protocol," in *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 3, Chicago, IL, Oct. 1998, pp. 458 – 462.
- [104] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," in *Proc. IEEE INFOCOM*, New York, NY, June 2002, pp. 763 – 772.
- [105] "Video coding for low bitrate communication," Telecom. Standardization Sector of ITU, Feb. 1998, ITU-T Recommendation H.263 Version 2.

- [106] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. New York, NY: Prentice Hall, 2001.
- [107] V. Vaishampayan and S. John, “Balanced interframe multiple description video compression,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 3, Kobe, Japan, Oct. 1999, pp. 812 – 816.
- [108] G. Venter and R. T. Haftka, “Two species genetic algorithm for design under worst case conditions,” in *Proc. World Congress on Structural and Multidisciplinary Optimization*, Buffalo, NY, 1999, pp. 183 – 195.
- [109] W.-H. Wang, M. Palaniswami, and S. H. Low, “Optimal flow control and routing in multi-path networks,” *Performance Evaluation*, vol. 52, no. 2-3, pp. 119 – 132, 2003.
- [110] X. Wang, “Design and implementation of an emulation testbed for video communications in ad hoc networks,” Master’s thesis, Virginia Polytechnic Institute and State University, Jan. 2006.
- [111] Y. Wang and S. Lin, “Error resilient video coding using multiple description motion compensation,” in *Proc. IEEE Workshop on Multimedia Signal Processing (MMSP)*, vol. 2, Cannes, France, Oct. 2001, pp. 441 – 447.
- [112] Y. Wang, M. Orchard, and A. Reibman, “Multiple description image coding for noisy channels by pairing transform coefficients,” in *Proc. IEEE Workshop on Multimedia Signal Processing (MMSP)*, Princeton, NJ, June 1997, pp. 419 – 424.
- [113] Y. Wang, M. Orchard, V. Vaishampayan, and A. Reibman, “Multiple description coding using pairwise correlating transforms,” *IEEE Trans. on Image Processing*, vol. 10, no. 3, pp. 351 – 366, Mar. 2001.
- [114] Y. Wang, R. R. Reibman, and S. Lin, “Multiple description coding for video delivery,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 57 – 70, Jan. 2005.

- [115] Z. Wang and J. Crowcroft, "Quality-of-Service routing for supporting multimedia applications," *IEEE J. on Selected Areas in Communications*, vol. 17, no. 8, pp. 1488 – 1505, Aug. 1999.
- [116] T. Wiegand, N. Frber, K. Stuhlmller, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE J. on Selected Areas in Communications*, vol. 18, no. 6, pp. 1050 – 1062, June 2000.
- [117] A. Yener and C. Rose, "Genetic algorithms applied to cellular call admission: Local policies," *IEEE Trans. on Vehicular Technology*, vol. 46, no. 1, pp. 72 – 79, Feb. 1997.
- [118] B. Zhang and H. T. Mouftah, "QoS routing for wireless ad hoc networks: Problems, algorithms, and protocols," *IEEE Communications Magazine*, vol. 43, no. 10, pp. 110 – 117, Oct. 2005.

Vita

Sastry Kompella obtained his B.E. degree in Electronics and Communication Engineering from Andhra University, Visakhapatnam India in 1996 and the M.S. degree in Electrical Engineering from Texas Tech University in 1998. His Master's research focused on developing vector quantization algorithms for image compression. From 1998 to 2003, Sastry worked as a member of technical staff at Symtx Inc., Austin, TX. His work experience included building Automated Test Equipment (ATE) for network router switch modules, DSL modems, cellular telephony, and Synchronous Optical Network (SONET) devices.

From 2003 to 2006, Sastry was a Ph.D. student in the Bradley Department of Electrical and Computer Engineering at Virginia Polytechnic Institute and State University (Virginia Tech), and worked as a Graduate Research Assistant in the Complex Network Systems Research (CNSR) Lab headed by his advisor Prof. Y. Thomas Hou. His Ph.D. was funded by the National Science Foundation (NSF) through the Integrated Research and Education in Advanced Networking (IREAN) Fellowship. His research interests are in the algorithmic design and optimization for wireless network systems, protocol design, and cross-layer routing for multimedia streaming systems.

Publications

1. **S. Kompella**, S. Mao, Y. T. Hou, and H. D. Sherali, "Cross-layer optimized routing for md video in multi-hop wireless networks," submitted to *IEEE J. Selected Areas in Communications*, under review.

2. **S. Kompella**, S. Mao, Y. T. Hou, and H. D. Sherali, “On path selection and rate allocation for video in wireless mesh networks,” submitted to *IEEE Trans. on Networking*, under review.
3. S. Mao, **S. Kompella**, Y. T. Hou, H. D. Sherali, and S. F. Midkiff, “Routing for concurrent video sessions in ad hoc networks,” *IEEE Trans. on Vehicular Technology*, vol. 55, no. 1, pp. 317 – 327, Jan. 2006.
4. **S. Kompella**, S. Mao, Y. T. Hou, and H. D. Sherali, “Optimal rate control for video transport over multi-hop wireless networks,” in *Proc. IEEE IEEE Wireless Communications & Networking Conference (WCNC)*, Las Vegas, USA, Apr. 2006.
5. S. Mao, **S. Kompella**, Y. T. Hou, and S. F. Midkiff, “A fast greedy algorithm for routing concurrent video flows,” in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3535 – 3538, Kobe, Japan, May 2005.
6. S. Mao, **S. Kompella**, Y. T. Hou, and H. D. Sherali, “Routing for multiple concurrent video sessions in wireless ad hoc networks,” in *Proc. IEEE International Conference on Communications (ICC)*, vol. 2, pp. 1229 – 1235, Seoul, Korea, May 2005.
7. **S. Kompella**, *An Optimized Vector Quantization for Color Image Compression*, Master’s Thesis, Texas Tech University, Feb. 1998. BIB Call Number: AC805 .T3 1998 No.52
8. S. Mitra, M. Wilson, and **S. Kompella**, “Performance of multi-resolution pattern classifiers in medical image encoding from wavelet coefficient distributions,” in *SPIE Proc. Medical Imaging*, vol. 3338, pp. 256 – 263, Feb. 1998.
9. S. Mitra, S. Pemmaraju, **S. Kompella**, and S. Meadows, “Efficient color image compression using integrated fuzzy neural networks for vector quantization,” in *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, pp. 184 – 188, Oct. 1997.
10. S. Mitra, S. Pemmaraju, and **S. Kompella**, “Low bit rate with excellent image reconstruction capability by adaptive multiresolution code books in vector quantization,” in *SPIE Proc. Medical Imaging*, vol. 3031, pp. 277 – 285, May 1997.