Multiple Description Video Communications in Wireless Ad Hoc Networks

Xiaolin Cheng

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

Master of Science
in
Computer Engineering

Y. Thomas Hou, Chair
Shiwen Mao
Scott F. Midkiff
Jeffrey H. Reed

June 15, 2005
Blacksburg, Virginia

Keywords: Cross-layer design, genetic algorithms, multipath routing, multiple description video, server diversity, video multicast, wireless ad hoc networks

Multiple Description Video Communications in Wireless Ad Hoc Networks

Xiaolin Cheng

ABSTRACT

As developments in wireless ad hoc networks continue, there is an increasing expectation with regard to supporting content-rich multimedia communications (e.g., video) in such networks, in addition to simple data communications. The recent advances in *multiple description* (MD) video coding have made it highly suitable for multimedia applications in such networks. In this thesis, we study three important problems regarding multiple description video communications in wireless ad hoc networks. They are multipath routing for MD video, MD video multicast, and joint routing and server selection for MD video in wireless ad hoc networks. In the multipath routing for MD video problem, we follow an *application-centric* cross-layer approach and formulate an optimal routing problem that minimizes the application layer video distortion. We show that the optimization problem has a highly complex objective function and an exact analytic solution is not obtainable. However, we find that a metaheuristic approach such as *Genetic Algorithms* (GAs) is eminently effective in addressing this type of complex cross-layer optimization problem. We provide a detailed solution procedure for the GA-based approach, as well as a tight lower bound for video distortion. We use numerical results to compare this approach to several other approaches and demonstrate its superior performance. In the MD video multicast problem, we take the similar *application-centric*, cross-layer approach as in the multipath routing problem. We propose an MD video multicast scheme where multiple source trees are used. Furthermore, each video description is coded into multiple layers in order to cope with variation in wireless link bandwidths. Based on this multicast model, we formulate the multicast routing as a combinatorial optimization problem and apply a *Genetic Algorithm*-based metaheuristic procedure to solove this problem. Performance comparisons with existing approaches show significant gains for a wide range of network operating conditions. In the last problem, we study the important problem of joint routing and server selection for MD video in ad hoc networks. We formulate the task as a combinatorial optimization problem and

present tight lower and upper bounds for the achievable distortion. The upper bound also provides a feasible solution to the formulated problem. Our extensive numerical results show that the bounds are very close to each other for all cases studied, indicating the near-global optimality of the derived upper bounding solution. Moreover, we observe significant gains in video quality achieved by the proposed approach over existing server selection schemes. This justifies the importance of jointly considering routing and server selection for optimal MD video streaming in wireless ad hoc networks.

Acknowledgements

First, I would like to thank my advisor, Prof. Y. Thomas Hou, who led me into this exciting area of networking research in the last two years, for his invaluable support and guidance throughout the course of this thesis. His extensive knowledge, enlightening direction, and continuous encouragement made my thesis work positive and fruitful.

I want to thank Dr. Shiwen Mao specially for his great help in my thesis work. I have been working with him closely since I started my thesis work. I am deeply impressed by his ample knowledge, keen research insights and enumerous patience in his guidance to my research work. I experienced and gained much from him, both knowledge and methodology in networking research.

I also benefited very much from Prof. Hanif D. Sherali. I highly appreciate his great help in my thesis research. He reviewed our papers very carefully and offered many constructive comments and suggestions, which enhanced the quality of our papers significantly. His brilliant ideas have been impressed me and inspired me all the time.

I am also indebted to other members of my thesis committee, including Prof. Scott F. Midkiff and Prof. Jeffrey H. Reed. Their support for my thesis work was indispensable and helped me greatly. I want to thank them for spending time reviewing my work and giving valuable suggestions on my work. I learned much by following their excellent research work.

I want to acknowledge my fellow students and friends at Virginia Tech, including Sastry Kompella, Yi Shi, Anant Utgikar, Animesh Patcha, Haisang Wu and many others. I thank them for the discussions, cooperation, and assistance during past two years.

Finally, I would like to show my great appreciation to my parents, my sister and brother for their continuous support during all these years. Without their love, support, and encouragement, I would not have finished this research work.

The research in this thesis has been supported in part by the National Science Foundation (NSF) under Grants ANI-0312655, CNS-0347390, and DMI-0094462, and by the Office of Naval Research (ONR) under Grants N00014-03-1-0521 and N00014-05-1-0179.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Wireless Ad Hoc Networks

With the recent advances in wireless technologies, wireless networks are becoming a significant part of today's access networks. Ad hoc networks are wireless mobile networks without an infrastructure. Since no pre-installed base stations are required, ad hoc networks can be deployed quickly at conventions, disaster recovery areas, and battlefields. When deployed, mobile nodes cooperate with each other to find routes and relay packets for each other. In ad hoc networks, a wireless link usually has high transmission error rate because of shadowing, fading, path loss, and interference from other transmitting users. An end-to-end path found in ad hoc networks has an even higher error rate since it is the concatenation of multiple wireless links. Moreover, user mobility makes the network topology constantly change. Ad hoc networks also need to reconfigure themselves when users join or leave the network. The frequent link failures and topology changes are the key characteristics that distinguish ad hoc networks with wireline networks and traditional cellular-based wireless networks.

Wireless ad hoc networking is an active research topic in networking research communities. Communcations among peers in the network are achieved by multi-hop wireless communications without any prior infrastructure support. In wireless ad hoc networks, link or node failure, fading, shadowing and other dynamics associated with mobility make reliable com-

muncations among nodes difficult. To address those problems, many researchers have been working on routing protocols to support data communcations in ad hoc networks. However, most of the routing schemes proposed in wireless ad hoc networks so far are network-centric, which means the routing algorithms are based on the network layer metrics, such as number of hops, link failure probabilities, and link bandwidths. In multimedia communications, especially for video services, performance metrics in the application layer are more important. For example, in video applications, video quality at the receiver is a key measure of performance. Since link failures and frequent mobility cause packet losses and degrade the received video quality, these problems should be addressed to enable video services in such networks. Naturally, we need to consider application layer performance in this case and follow a cross-layer design principle when offering video services in wireless ad hoc networks.

1.2 Multiple Description Video Coding

It is a great challenge to provide video services in ad hoc networks. In ad hoc networks, an end-to-end route may only exist for a short period of time. The frequent link failures and route changes cause packet losses and reduce the received video quality. One common feature of wireline and wireless ad hoc networks is that both often have a mesh topology, which implies the likely existence of multiple paths between two nodes. If we use multiple paths for a video session, the video stream can be divided into multiple substreams and each substream is sent on one of the paths. Thus, the traffic is more evenly distributed in the network and congestion is less likely to occur. The packet loss due to congestion can, therefore, be greatly reduced. Furthermore, if these paths are disjoint, the losses experienced by the substreams would be relatively independent. Therefore, better error resilience can be achieved when traffic dispersion is performed appropriately and with effective error control for the substreams. Multipath transport (MPT) provides an extra degree of freedom in designing video coding and error control schemes.

For MPT to be helpful for 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, a new video coding technique called *multiple description coding* (MDC) has been successfully developed, which has this kind of nice feature. MDC 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. In the worst case, information received from *any* video description can be used to reconstruct video with acceptable quality. The quality of the reconstructed video improves in proportion to the information received from the number of video descriptions. This new video coding technique is drastically different from existing video coding schemes such as 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) and the decoding of the upper enhancement layer hinges upon the lower enhancement layers (in addition to the base layer). MDC is also completely different from traditional single stream video coding, whose perceived video quality at the receiver is highly susceptible to the dynamics of the single path.

It has been recognized that MD coding matches perfectly with the wireless ad hoc network environment for multimedia applications [38]. This is because the topology of such networks is intrinsically mesh, within which multiple paths exist between any source and destination pair. Although most of the paths in such networks are highly fragile (i.e., will not remain reliable for an extended period of time), as long as the link/node failure events on different paths are not entirely correlated, the probability of concurrent loss of all of the descriptions will be low. Therefore, MD coding will remain effective for most of the streaming period, while video quality improves as more descriptions are received.

For ad hoc networks operating under extreme conditions, the wireless link within such a network is 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. On the other hand, the new MDC technique does not require a reliable path. Indeed, within an MDC 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.

1.3 Metaheuristic and Genetic Algorithms

Since the problems associated with MD video communications in ad hoc networks are complex, and exact solutions are not obtainable in a reasonable amount of time, we suggest that the best strategy to address this type of problem is to view the problem as a "black-box" optimization problem and explore an effective *metaheuristic* approach. The term metaheuristic derives from the composition of two Greek words: *heuristic* means "to find" while the prefix *meta* means "beyond, in an upper level." It refers to the set of strategies that guide the iterative search process, in order to efficiently explore the search space to find (near-) optimal solutions [16]. In particular, we find the paradigm of the *genetic algorithm* (GA) to be emiently well-suited for addressing this type of a complex problem. GA is a *population-based* metaheuristic that is inspired by the principle of *survival of the fittest* in an analogous natural evolution context. It has 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 general, population-based metaheuristics can provide a natural, intrinsic way for the exploration of the search space. At each iteration, a number of operators are applied to the individuals of the current population to generate individuals of the population for the next generation. In particular, GA uses operators known as *recombination* (or *crossover*) to recombine two or more individuals to produce new individuals, and *mutation* (or *modification*) 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 having a higher degree of fitness will have a higher probability to be chosen as members of the population for the next iteration.

Genetic algorithms were originally proposed by John Holland in 1975 [29], which are the probabilistic optimization algorithms based on natural phenomena and biological mechanisms. The current set of feasible solutions represents the population and each of its elements is

viewed as an individual. In this perspective, the natural evolution of the population through generations is simulated according to genetic inheritance and survival rules. It is assumed that fit individuals are likely to get involved in the reproduction phase and, therefore, have enhanced probabilities of survival. Parents' characteristics are inherited in a probabilistic manner.

In the terminology of GA, individuals are called *genotypes,* whereas the solutions that are encoded by individuals are called *phenotypes.* The fundamental idea of GA is that it operates on a finite population of "chromosomes" (solutions). The chromosomes are fixed strings, typically, though not necessarily, with binary or integer value ("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 offsprings. Genetic operators are used to facilitate a breeding process that results in offsprings inheriting properties from their parents. The fitter a member of the population, the more likely it is to produce superior offsprings. The offsprings are evaluated and placed in the population, possibly replacing the weaker members of the last generation. Thus, the search mechanism consists of three phases: evaluation of the fitness of each chromosome, selection of the parent chromosomes, and applications of mutation and re-combination (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 ceases to improve. The "survival of the fittest principle" ensures that the overall quality of solutions increases as the algorithm progresses from one generation to the next.

We find that GA is a powerful approach to address the combinatorial optimization problems being considered in this thesis. At each iteration, both *self-adaptation* and *cooperation* take place. Self-adaptation refers to the individuals evolving independently, which is enabled by the mutation operator in the GA. On the other hand, cooperation implies the sharing of the best traits among the individuals, which is in the form of the crossover operator in the GA. Compared to trajectory (*single-solution based*) methods (e.g., Simulated Annealing and Tabu Search), which merely use local search to improve an initial solution (obtained via some constructive method), GAs have the additional advantage of being able to combine

good solutions in order to possibly derive improved solutions. The basic assumption within this paradigm is that good solutions often share parts with optimal solutions. Under GA, improved solutions are sought by combining the best parts of the currently-known promising solutions in the population.

The potential of GA in addressing networking problem has been recognized by some researchers in recent years. It has been explored to address many important networking problems. These efforts have made the important first step in exploring the potential of GA for network optimization. The research addressed in this thesis builds upon the early work on applying GA to address network-centric problem. We aim at exploring GA's potential to address complex cross-layer optimization problem with objective function at the application layer. Obviously, this problem is more complex than network-centric based GA research as it requires knowledge not only at the network layer, but also an understanding at the application layer (i.e., video coding capability) to fully exploit the optimization space across layers.

1.4 Major Contributions

In this thesis, we address the important problems of enabling video transport using multiple description video coding scheme in wireless ad hoc networks. The major contributions are as follows.

First, we studied the important problem of optimal multipath routing for MD video. We formulated the multipath routing problem following an application-centric, cross-layer approach. We designed a GA-based algorithm to address this multipath routing problem and found that this approach provides near-optimal results and is superior to the existing schemes. We also developed a tight lower bound for video distortion, which can be used to evaluate the performance of a GA-based solution as well as to set its termination criteria.

Second, we proposed a multicast scheme for MD video over ad hoc networks, within which multiple source trees are used. Furthermore, each video description is coded into multiple

layers to cope with diversity in wireless link bandwidths. A similar application-centric, cross-layer methodology was exploited. We found that GA was also highly suitable for solving the combinatorial optimization problem based on this MD multicast routing problem. Extensive simulations demonstrate significant gains in video quality achieved over existing approaches for a wide range of network operational conditions.

Third, we studied the important problem of jointly selecting servers and determining optimal routes for MD video streaming in wireless ad hoc networks. This task was formulated as a combinatorial optimization problem that minimizes the received video distortion. We derived a lower bound and an upper bound for the best achievable video distortion. Our extensive numerical results show that the bounds are very close to each other for all cases studied, indicating the near-global optimality of the derived upper bounding solution. Significant gains in video quality achieved by the proposed approach over existing server selection schemes were also observed.

1.5 Thesis Outline

The general background for the problems of enabling MD video communications in wireless ad hoc networks and their general solution approaches are presented in this chapter. The rest of the thesis is organized as follows.

We present the study on multipath routing for MD video in wireless ad hoc networks in Chapter 2. In Chapter 3, we present a multicast scheme for MD video in wireless ad hoc networks. These two problems are formulated into two combinatorial optimization problems from an application-centric, cross-layer perspective. To solve the problems, we developed efficient GA-based solution procedures and showed their excellent performance based on extensive simulation results. Next, a joint optimal routing and server selection problem in wireless ad hoc networks is presented in Chapter 4. We derived a lower bound and an upper bound for the client-end video distortion. We demonstrated the optimality of the upper bound and its superiority to the existing algorithms. Finally, we present our conclusions and future research directions in Chapter 5.

Chapter 2

Multipath Routing for MD Video over Wireless Ad Hoc Networks

2.1 Introduction

Several researchers have proposed to use MD coding with *multipath routing* for multimedia transport [7,8,17,26,38]. These interesting works have successfully demonstrated the efficacy of using MD with multipath routing, assuming that the set of paths is given *a priori*.. However, the difficult problem of finding the best paths for the descriptions has not been adequately addressed. In a recent work in [12], Begen *et al.* studied the problem of multipath routing for MD video in the context of Internet overlay networks. The optimal routing problem is, however, solved via exhaustive search which has an exponential complexity. To reduce the computational complexity, a heuristic algorithm was proposed in [13]. However, this heuristic relies on the special hierarchical structure of the Internet and overlay networks and, thus, may not be suitable for wireless ad hoc networks.

In this chapter, we study the important problem of multipath routing for MD video in wireless ad hoc networks. We follow a *cross-layer* approach in problem formulation by considering the application layer performance (i.e., average video distortion) as a function of network layer performance metrics (e.g., bandwidth, loss, and path correlation). We show that the

objective function is a complex ratio of high-order polynomials and is non-decomposable. Consequently, it would be hard to develop a tractable exact solution. However, we find that a metaheuristic technique, such as *Genetic Algorithms* (GAs) [9], is eminently suitable in addressing such type of complex cross-layer optimization problems. This is because GAs possess an intrinsic capability of handling a *population* of solutions rather than working with a single current solution during each iteration. Such a capability gives GAs the unique strength of identifying promising regions in the search space (not necessarily convex) and having less of a tendency to be trapped in a local optimum, as compared with other trajectory-based metaheuristics (e.g., *simulated annealing* (SA) and *tabu search* (TS) [16]). Using numerical results, we show that significant performance gains can be achieved by the GA-based approach over trajectory-based approaches. To examine the quality of GA solutions, as well as setting its termination conditions, we develop a simple but tight lower bound on video distortion, which has similar computational complexity as Dijkstra's algorithm. Finally, we show that the GA-based multipath routing can be incorporated into many existing distributed ad hoc network routing protocols (e.g., [19,32]), particularly the class of *proactive* protocols. As an example, we present a distributed implementation based on the Optimized Link State Routing Protocol (OLSR) [19].

The remainder of this chapter is organized as follows. In Section 2.2, we formulate a cross-layer optimization problem for MD video over multiple paths in ad hoc networks. Section 2.3 presents a lower bound for video distortion. In Section 2.4, we present our GA-based approach and Section 2.5 shows numerical results. Section 2.6 discusses a distributed implementation of the proposed approach. Section 2.7 discusses related work and Section 2.8 concludes this chapter.

2.2 Problem Description

An ad hoc network can be modeled as a stochastic directed graph $\mathcal{G}\{V, E\}$, where V is the set of vertices and E is the set of edges. We assume that nodes are reliable during the video session, but links may fail with certain probabilities. Accurate and computationally

Table 2.1: Notation

Symbols	Definitions
$\mathcal{G}\{V, E\}$:	graph representation of the network
V :	set of vertices in the network
E :	set of edges in the network
s :	source node
t :	destination node
$\mathcal{P}$:	a path from s to t
g_i :	an intermediate node in a path
$\{i, j\}$:	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
d_0 :	distortion when both descriptions are received
d_h :	distortion when only Description h is received, h=1,2
D :	average distortion
T_{on} :	average "up" period of the joint links
P_{00} :	probability of receiving both descriptions
P_{01} :	probability of receiving description 1 only
P_{10} :	probability of receiving description 2 only
P_{11} :	probability of losing both descriptions
x_{ij}^h :	routing index variables, defined in (2.6)
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 $\mathcal{P}_h$
B_{jnt} :	minimum bandwidth of the shared links
θ :	crossover rate
μ :	mutation rate

efficient characterization of an end-to-end path in a wireless ad hoc network (or even a wireless link [33]) with consideration of mobility, interference, and the time-varying wireless channels is extremely difficult and remains an open problem. As an initial step, we focus on the network layer characteristics in this chapter, assuming that the physical and MAC layer dynamics of wireless links are translated into network layer parameters. For example, we could characterize a link $\{i, j\} \in E$ by:

- b_{ij}: the available bandwidth of link $\{i, j\}$;
- p_{ij}: the probability when link $\{i, j\}$ is "up";
- l_{ij}: average burst length for packet losses on link $\{i, j\}$.

In practice, these parameters can be measured by every node and distributed throughout the network using Link State Advertisements (LSA) [19]. We focus on the bandwidth and failure probabilities of a path, since these two are key characteristics for data transmission, as well as the most important factors that determine video distortion (see Eq. (2.2)). Other link characteristics, such as delay, jitter, congestion, and signal strength can be incorporated into this framework as well (e.g., see [37]). Table 2.1 lists the notation used in this chapter.

2.2.1 Rate-Distortion Regions for MD Coding

Throughout this chapter, we use double-description coding for MD video. We consider double-description video since it is most widely used in practice [7, 8, 12, 13, 17, 26, 38]. In general, using more descriptions and paths will increase the robustness to packet losses and path failures. However, more descriptions may increase the video bit rate for the same video quality. The study in [54] demonstrates that the most significant performance gain is achieved when the number of descriptions increases from 1 to 2, with only marginal improvement achieved for further increase in number of descriptions.

For video coding and communications, a rate distortion model describes the relationship between the bit rate and the achieved distortion. For two descriptions (each generated for

a sequence of video frames), denote d_h the achieved distortion when only Description h is received, $h = 1, 2$, and d_0 the distortion when both descriptions are received. Denote R_h the rate in bits/sample of Description h, $h = 1, 2$. The rate-distortion region for a memoryless *i.i.d.* Gaussian source with the square error distortion measure was first introduced in [44]. For computational efficiency, Alasti *et al.* in [4] introduce the following rate-distortion region (also employed in this chapter).

$$\begin{cases} d_0 = \frac{2^{-2(R_1+R_2)}}{2^{-2R_1}+2^{-2R_2}-2^{-2(R_1+R_2)}} \cdot \sigma^2 \\ d_1 = 2^{-2R_1} \cdot \sigma^2 \\ d_2 = 2^{-2R_2} \cdot \sigma^2, \end{cases} \tag{2.1}$$

where σ^2 is the variance of the source. Denote P_{00} the probability of receiving both descriptions, P_{01} the probability of receiving Description 1 only, P_{10} the probability of receiving Description 2 only, and P_{11} the probability of losing both descriptions. Then, the average distortion of the received video can be expressed as:

$$D = P_{00} \cdot d_0 + P_{01} \cdot d_1 + P_{10} \cdot d_2 + P_{11} \cdot \sigma^2. \tag{2.2}$$

2.2.2 Description Rates and Success Probabilities

As a first step to formulate the problem of optimal multipath routing, we need to know how to compute the average distortion D as a function of link statistics for a *given* pair of paths. That is, we need to compute the end-to-end bandwidth (or rate) for each stream and joint probabilities of receiving the descriptions (see Eqs. (2.1) and (2.2)).

For a source-destination pair $\{s, t\}$, suppose we have two given paths $[\mathcal{P}_1, \mathcal{P}_2]$ in $\mathcal{G}\{V, E\}$. Since we do not mandate "disjointedness" in path selection, $\mathcal{P}_1$ and $\mathcal{P}_2$ may share nodes and links in $\mathcal{G}\{V, E\}$. Similar to [12], we classify the links along the two paths 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 consist of disjoint links on the two paths, denoted as $\bar{\mathcal{J}}(\mathcal{P}_h)$, $h = 1, 2$, respectively. Then, the minimum bandwidth of $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$, B_{jnt}, is:

$$B_{jnt} = \begin{cases} \min_{\{i,j\} \in \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)}\{b_{ij}\}, & \text{if } \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) \neq \emptyset \\ \infty, & \text{otherwise.} \end{cases}$$

The rates of the two video streams, R_1 and R_2, can be computed as:

$$\begin{cases} R_h = \rho \cdot B(\mathcal{P}_h), & \text{if } \sum_{m=1}^{2} B(\mathcal{P}_m) \leq B_{jnt}, h = 1, 2 \\ R_1 + R_2 \leq \rho \cdot B_{jnt}, & \text{otherwise,} \end{cases} \tag{2.3}$$

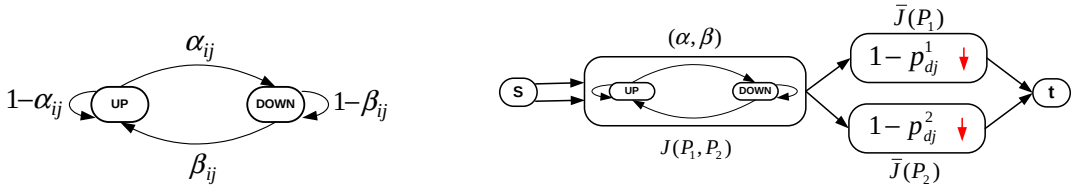
where $B(\mathcal{P}_h) = \min_{\{i,j\} \in \mathcal{P}_h} \{b_{ij}\}$, $h = 1, 2$, and ρ is a constant determined by the video format. The first line in (2.3) is for the case when the joint links are not the bottleneck of the paths. The second line of (2.3) is for the case where one of the joint links is the bottleneck of both paths. In the latter case, we assign the bandwidth to the paths by splitting the bandwidth of the shared bottleneck link in proportion to the mean success probabilities of the two paths.

We now focus on how to compute the end-to-end success probabilities. We model each link $\{i, j\}$ as an on-off process modulated by a discrete-time Markov chain, as shown in Figure 2.1(a). There is no packet loss when the link is in the "up" state; packet loss rate is 1 when the link is in the "down" state. Transition probabilities, $\{\alpha_{ij}, \beta_{ij}\}$, can be computed from the link statistics, as $\beta_{ij} = 1/l_{ij}$ and $\alpha_{ij} = (1 - p_{ij})/(p_{ij} l_{ij})$. For disjoint portion of the paths, it suffices to model the packet loss as a Bernoulli event, since losses on the two descriptions are assumed to be independent. Therefore, the success probabilities on the disjoint portions of the two paths are:

$$p_{dj}^{h} = \begin{cases} \prod_{\{i,j\} \in \bar{\mathcal{J}}(\mathcal{P}_h)} p_{ij}, & \text{if } \bar{\mathcal{J}}(\mathcal{P}_h) \neq \emptyset, h = 1, 2 \\ 1, & \text{otherwise}, h = 1, 2. \end{cases} \tag{2.4}$$

On the joint portion of the paths, losses on the two streams are correlated. If there are K shared links, the aggregate failure process of these links is a Markov process with 2^K states. To simplify the computation, we follow an approach similar to [12] 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. Denote T_{on} the average length of the "up' period. Then,

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



(a) The Gilbert two-state link model.

(b) A simplified path model for double-description video.

Figure 2.1: Link and path models.

The average success probability of the joint portion is:

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

Finally, the transition probabilities of the aggregate on-off process are:

$$\begin{cases} \alpha = \frac{1}{T_{on}} \\ \beta = \frac{p_{jnt}}{T_{on}(1-p_{jnt})}. \end{cases}$$

Note that $\alpha = 0$ and $\beta = 0$ if $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) = \emptyset$. The consolidated path model is illustrated in Figure 2.1(b), where $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$ is modeled as a two-state Markov process with parameters $\{\alpha, \beta\}$, and $\bar{\mathcal{J}}(\mathcal{P}_h)$ is modeled as a Bernoulli process with parameter p_{dj}^h, $h = 1, 2$.

With the above path model, the joint probabilities of receiving the descriptions can be computed as:

$$\begin{cases} P_{00} = p_{jnt} \cdot (1-\alpha) \cdot p_{dj}^1 \cdot p_{dj}^2 \\ P_{01} = p_{jnt} \cdot p_{dj}^1 \cdot \left[1 - (1-\alpha) \cdot p_{dj}^2\right] \\ P_{10} = p_{jnt} \cdot \left[1 - (1-\alpha) p_{dj}^1\right] \cdot p_{dj}^2 \\ P_{11} = 1 - p_{jnt} \cdot \left[p_{dj}^1 + p_{dj}^2 - (1-\alpha) \cdot p_{dj}^1 \cdot p_{dj}^2\right]. \end{cases} \tag{2.5}$$

2.2.3 The Optimal Multipath Routing Problem

With the above preliminaries, we now set out to formulate the multipath routing problem for MD video. To characterize any s–t path $\mathcal{P}_h$, we define the following binary variables:

$$x_{ij}^h = \begin{cases} 1, & \text{if } \{i,j\} \in \mathcal{P}_h \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

With these variables, an arbitrary path $\mathcal{P}_h$ can be represented by a vector $\mathbf{x}^h$ of $|E|$ elements, each of which corresponds to a link and has a binary value. We can formulate the problem of multipath routing for MD video (OPT-MM) as follows.

OPT-MM Given a wireless ad hoc network $\mathcal{G}\{V, E\}$ and a source destination pair s–t,

$$\textbf{Minimize}: \quad D = P_{00} \cdot d_0 + P_{01} \cdot d_1 + P_{10} \cdot d_2 + P_{11} \cdot \sigma^2 \tag{2.7}$$

subject to:

$$\sum_{j:\{i,j\}\in E} x_{ij}^h - \sum_{j:\{j,i\}\in E} x_{ji}^h = \begin{cases} 1, & \text{if } i = s, \quad \forall i \in V,\ h = 1,2 \\ -1, & \text{if } i = t, \quad \forall i \in V,\ h = 1,2 \\ 0, & \text{otherwise}, \quad \forall i \in V,\ h = 1,2 \end{cases} \tag{2.8}$$

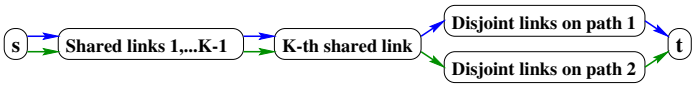
$$\sum_{j:\{i,j\}\in E} x_{ij}^h \begin{cases} \leq 1, & \text{if } i \neq t, \quad \forall i \in V,\ h = 1,2 \\ = 0, & \text{if } i = t, \quad \forall i \in V,\ h = 1,2 \end{cases} \tag{2.9}$$

$$x_{ij}^1 \cdot R_1 + x_{ij}^2 \cdot R_2 \leq \rho \cdot b_{ij}, \quad \forall \{i,j\} \in E \tag{2.10}$$

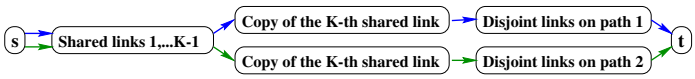
$$x_{ij}^h \in \{0,1\}, \quad \forall \{i,j\} \in E,\ h = 1,2. \tag{2.11}$$

In Problem OPT-MM, $\{x_{ij}^h\}$ are binary optimization variables. Constraints (2.8) and (2.9) guarantee that the paths are loop-free, while constraint (2.10) guarantees the links are stable. 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 (2.1), (2.3), and (2.5).

Clearly, the objective function (2.7) is a highly complex ratio of high-order polynomials of the x-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 (or can be conditioned on the exceedingly complex products of the binary factors x_{ij}^1 and $(1-x_{ij}^1)$ with x_{ij}^2 and $(1-x_{ij}^2)$). In [58], Sherali *et al.* considered a problem that seeks a pair of disjoint paths in a network such that the total travel time over the paths is minimized, where the travel time on a link might be either a constant, or a non-decreasing (or unstructured) function of the time spent on the previous links traversed. Even for a simple special case where all the links except one have a constant travel time (and hence linear objective terms), this problem is shown to be NP-hard. Our problem has much more complex relationships pertaining to the contribution of each individual link to the objective function, which depends



(a) Solution $\hat{x}$.



(b) Solution $\bar{x}$.

Figure 2.2: The two solutions have the same set of links. The only difference between them is that a link is shared in $\hat{x}$ (the K-th shared link), but not shared in $\bar{x}$ (a copy is appended to each of the disjoint portions).

in general on the other links that are included in a fashion that has no particular structural property such as convexity. Hence, it is likely to be NP-hard as well. However, we leave a rigorous proof of this NP-hardness to future research.

2.3 A Lower Bound for Distortion

Before describing our GA-based approach, we first construct a lower bound on the achievable video distortion. Such a bound will be very useful in measuring the performance of a heuristic algorithm, as well as serving as a reference for setting its termination conditions.

We find that the average video distortion D possesses the following *monotonicity* properties (the proofs are presented in Appendix A):

M1 : D is non-increasing with R_h, $h = 1, 2$.

M2 : For two completely disjoint paths, D is non-increasing with p_{dj}^h, $h = 1, 2$.

M3 : Consider the two solutions $\hat{x}$ and $\bar{x}$ shown in Figure 2.2. Assume (i) the two solutions provide the same description rates (i.e., the Kth shared link is not the bottleneck link of the two paths); and (ii) the on-off failure process of the Kth shared link is random or bursty, i.e., $\alpha_K + \beta_K \leq 1$. Then $D(\hat{x}) \geq D(\bar{x})$.

0. Find the maximum end-to-end bandwidth, b^*, among all s–t paths;
1. Find the maximum end-to-end success probability, p^*, among all s–t paths;
2. Construct a solution $x_l^* = \{\mathcal{P}_1^l, \mathcal{P}_2^l\}$, satisfying:
3. (1) $\mathcal{P}_1^l$ is disjoint with $\mathcal{P}_2^l$;
4. (2) $B(\mathcal{P}_h^l) = b^*$, $h = 1, 2$;
5. (3) $p_{dj}^h(\mathcal{P}_h^l) = p^*$, $h = 1, 2$.

Figure 2.3: ALG-LB: a procedure to construct a solution x_l^* that yields a lower bound for distortion D.

The assumption in Property M3 relates to the covariance of two consecutive failure events X_k and X_{k+1} on link $\{i, j\}$:

$$\text{Cov}\{X_k, X_{k+1}\} = \frac{\alpha_{ij}\beta_{ij}}{(\alpha_{ij} + \beta_{ij})^2}(1 - \alpha_{ij} - \beta_{ij}). \tag{2.12}$$

If $\alpha_{ij} + \beta_{ij} < 1$, two successive failures (or losing both descriptions sent back to back on this link) are positively correlated, i.e., the failure process is *bursty*, which, we argue, is not atypical in wireless ad hoc networks. When $\alpha_{ij} + \beta_{ij} = 1$, two successive failures are un-correlated, corresponding to *random* packet losses. When $\alpha_{ij} + \beta_{ij} > 1$, the successive failures are negatively correlated (called *sub-bursty*), which should be rare in wireless ad hoc networks. In Figure 2.2, if the Kth shared link has bursty or random losses, then $\bar{x}$ yields a distortion no higher than $\hat{x}$.

We are now ready to construct a simple, but, tight lower bound on the average video distortion. Algorithm ALG-LB in Figure 2.3 is such an algorithm for this purpose. In Figure 2.3, ALG-LB first determines the optimal end-to-end bandwidth b^* and the optimal end-to-end success probability p^*. It then constructs two *virtual* paths, which yield a distortion lower bound. Since we are interested in a lower bound, the corresponding physical paths are not necessarily feasible. In ALG-LB, b^* can be found using, e.g., the algorithm in [35] with time complexity $O(|E| \cdot \log^* |V|)$, where $\log^* n$ is the *iterated logarithm function*; p^* can be found by setting link costs to $\log(1/p_{ij})$, $\forall\{i, j\} \in E$, and then applying Dijkstra's algorithm to find the path having the minimum cost. The time complexity of finding p^* is $O(|E| \cdot \log |V|)$.

Proposition 1. *The distortion $D(x_l^*)$, where x_l^* is constructed by ALG-LB, is a lower bound for distortion D.*

A proof of Proposition 1 is available in the Appendix A.4. Note that although we show that x_l^* dominates all disjoint and joint feasible solutions, it does not necessarily imply that the optimal paths are always disjoint. The two optimal paths may share a "good" link to avoid the use of low quality links. Also, $D(x_l^*)$ becomes an *exact* bound, i.e., $D(x_l^*) = D(x^*)$, if x_l^* is realizable. We will illustrate the tightness of this lower bound in Section 2.5.

2.4 A GA-based Metaheuristic Approach

Although the lower bound offered by ALG-LB provides a good estimation of D, it may not yield a pair of feasible paths for the s–t video session. In this section, we present a solution procedure that always produces a pair of feasible and near-optimal paths.

We believe that the best strategy to address Problem OPT-MM is to explore an effective *metaheuristic* approach [16]. In particular, we find that *Genetic Algorithms* (GA) [9] are eminently suitable for addressing this type of complex combinatorial problems, most of which are multimodal and non-convex.[1] GAs are *population-based* metaheuristic inspired by the *survival-of-the-fittest* principle. 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. At each iteration, a number of genetic operators are applied to the individuals of the current population 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) for the next generation. The survival-of-the-fittest principle ensures that the overall quality of the population improves as the algorithm progresses from one generation to the next.

Figure 2.4 displays the flow chart for our GA-based approach to the MD multipath routing

[1]The other two *evolutionary computation* (EC) metaheuristics, i.e., *evolutionary programming* (EP) and *evolutionary strategy*, are usually used for continuous optimization problems, while GA is mainly applied to combinatorial optimization problems [16].

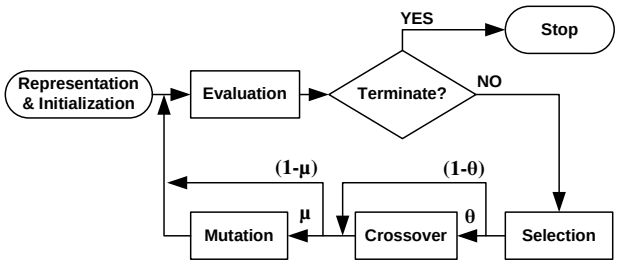


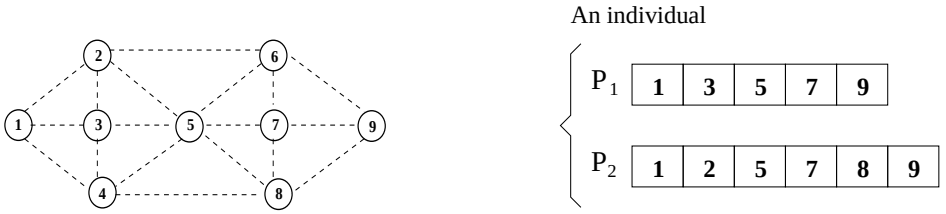
Figure 2.4: Flow chart of our GA-based approach.

problem, which includes the following components: *solution representation, initialization, evaluation, selection, crossover,* and *mutation.* The termination condition in Figure 2.4 could be based on the total number of iterations (generations), maximum computing time, a threshold of desired video distortion, or a threshold based on the lower bound obtained in Section 2.3. In what follows, we use an example ad hoc network shown in Figure 2.5(a) to illustrate the components in our GA-based approach.

Solution Representation and Initialization

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 (i.e., a chromosome). Such encoding should be suitable for fitness computation (which is determined by the objective function) and genetic operations. For a routing problem, a natural encoding scheme would be to define a node as a gene. Then, an end-to-end path, consisting of an ordered sequence of nodes (connected by the corresponding wireless links), can be represented as a chromosome [3]. For Problem OPT-MM, each feasible solution consists of a pair of paths (i.e., a pair of chromosomes), denoted as $[\mathcal{P}_1, \mathcal{P}_2]$. An individual in this case could be a pair of vectors containing the nodes on paths $\mathcal{P}_1$ and $\mathcal{P}_2$ (see, e.g., Figure 2.5(b)).

Before entering the main loop in Figure 2.4, we need to generate an initial population, i.e., a set of solutions. A simple approach would be to generate this set of solutions by randomly appending feasible elements (i.e., nodes with connectivity) to a partial solution. Under this approach, each construction process starts with source node s. Then, the process randomly chooses a link incident to the current end-node of the partial path and appends the link



(a) An example wireless ad hoc network.

(b) An example individual, $s = 1$, $t = 9$.

Figure 2.5: Example network and coding of an individual.

with its corresponding head-node to augment the path, until destination node t 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 s–t independently, a population of individuals can be constructed by pairing paths from this set. Our numerical results show that a properly-designed GA is not very sensitive to the quality of the individuals in the initial population.

Evaluation

The fitness function $f(\bar{x})$ of an individual, $\bar{x} = [\mathcal{P}_1, \mathcal{P}_2]$, is closely tied to the objective function (i.e., distortion D). Since the objective is to minimize the average distortion function D, we have adopted a fitness function defined as the inverse of the distortion value, i.e., $f(\bar{x}) = 1/D(\bar{x})$. This simple fitness definition appears to work very well, although we intend to explore other fitness definitions in our future effort.

Selection

During this operation, we select individuals that have a better chance or potential to produce "good" offsprings in terms of their fitness values. By virtue of the selection operation, "good" genes among the population are more likely to be passed to the future generations. Several selection schemes can be employed during this operation. For example, one possible scheme (known as *Roulette wheel* selection [9]) is to select an individual based on a probability in proportion to its normalized fitness value, i.e., $Pr\{\text{choosing individual } i\} = f(x_i)/\sum_j f(x_j)$. Another possible scheme (known as *Tournament* selection [9]) randomly chooses m individ-

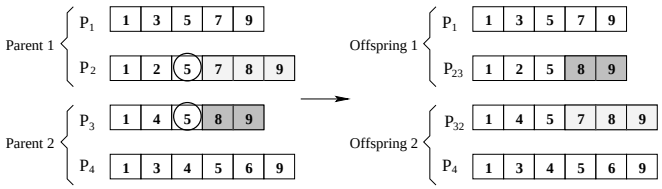


Figure 2.6: An example of the crossover operation.

uals from the population each time, and then selects the best of these m individuals in terms of their fitness values. By repeating either 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 offsprings. Crossover may create new individuals, thus exposing the search process to a new area of the fitness landscape. The decision of whether or not to perform a crossover operation is determined by the *crossover rate* θ.

Figure 2.6 illustrates one possible crossover implementation. Suppose that we have two parent individuals $x_1 = [\mathcal{P}_1, \mathcal{P}_2]$ and $x_2 = [\mathcal{P}_3, \mathcal{P}_4]$. We could randomly pick one path in x_1 and one in x_2, say $\mathcal{P}_2$ and $\mathcal{P}_3$. 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, where $g_r \notin \{s, t\}$, and we can then concatenate nodes $\{s, \cdots, g_r\}$ from $\mathcal{P}_2$ with nodes $\{g_{r+1}, \cdots, t\}$ in $\mathcal{P}_3$ (where g_{r+1} denotes the next downstream node of g_r in $\mathcal{P}_3$) to produce a new path $\mathcal{P}_{23}$. Likewise, using the first such node $g_{r'}$ in $\mathcal{P}_3$ that repeats in $\mathcal{P}_2$ (which may be different from g_r), we can concatenate the nodes $\{s, \cdots, g_{r'}\}$ from $\mathcal{P}_3$ with the nodes $\{g_{r'+1}, \cdots, t\}$ in $\mathcal{P}_2$ to produce a new path $\mathcal{P}_{32}$. It is important that we check the new paths to be sure that they are loop-free. The two offsprings generated in this manner are $[\mathcal{P}_1, \mathcal{P}_{23}]$ and $[\mathcal{P}_{32}, \mathcal{P}_4]$. On the other hand, if $\mathcal{P}_2$ and $\mathcal{P}_3$ are disjoint, we could swap $\mathcal{P}_2$ with $\mathcal{P}_3$ to produce two new offsprings $[\mathcal{P}_1, \mathcal{P}_3]$ and $[\mathcal{P}_2, \mathcal{P}_4]$.

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. 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.,

$$\begin{cases} \mu_0 = \mu_{max} \\ \mu_k = \mu_{max} - \frac{k \cdot (\mu_{max} - \mu_{min})}{T_{max}}, \end{cases} \tag{2.13}$$

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 2.7 illustrates a simple example of the mutation operation. In this example, we could implement mutation as follows. First, we choose a path $\mathcal{P}_h$, $h = 1$ or 2, with equal probabilities. Then, we can randomly pick an integer value k in the interval $[2, |\mathcal{P}_h| - 1]$, where $|\mathcal{P}_h|$ is the cardinality of $\mathcal{P}_h$, and let the partial path $\{s, \cdots, g_k\}$ be $\mathcal{P}_h^u$, where g_k is the k-th node along $\mathcal{P}_h$. Finally, we can use any constructive approach to build a partial path from g_k to t, denoted as $\mathcal{P}_h^d$, which does not repeat any node in $\mathcal{P}_h^u$ other than g_k. If no such alternative segment exists between g_k and t, we keep the path intact; otherwise, a new path can now be created by concatenating the two partial paths as $\mathcal{P}_h^u \cup \mathcal{P}_h^d$. For the example in Figure 2.7, we randomly choose node 5 (as g_k) on path $\mathcal{P}_1$, and reconstruct a new segment starting from node 5 to the destination node 9, i.e., $\{5, 6, 9\}$. The new path created by mutation, $\hat{\mathcal{P}}_1$, is the union of the first half of the original path $\{1, 3, 5\}$ and the

Figure 2.7: An example of the mutation operation.

newly-constructed segment $\{5, 6, 9\}$. The new individual is $\hat{x} = [\hat{\mathcal{P}}_1, \mathcal{P}_2]$.

2.5 Numerical Results

In this section, we present numerical results on Problem OPT-MM. In each experiment, we generate a wireless ad hoc network topology by placing a number of nodes at random locations in a rectangular region, where connectivity is determined by the distance coverage of each node's transmitter. The source and destination nodes, s and t, are randomly chosen. For every link, the failure probability is randomly chosen from [0.01, 0.3] according to a uniform distribution; the available bandwidth is randomly chosen from [100, 400] Kb/s according to a uniform distribution, with 50 Kb/s steps; the mean burst length is randomly chosen from [2,6] according to a uniform distribution. We set the GA's parameters as follows: the population size is 15; $\theta = 0.7$; μ is varied from 0.3 to 0.1 using the schedule described in Section 2.4; σ^2 is set to 1, since it only affects the absolute value of distortion, but does not affect path selection decisions. The GA program is terminated after a predefined number of generations or after a prespecified computation time has elapsed. The best individual found by the GA is prescribed as the suggested solution to Problem OPT-MM.

2.5.1 GA-based Algorithm versus Exhaustive Search

One important performance concern is the quality of the GA solutions. As discussed, due to the complex nature of Problem OPT-MM, a closed-form optimal solution is not obtainable. However, for small networks, an optimal solution may be numerically obtained via an exhaustive search and can be used to compare with the proposed GA-based solutions.

Table 2.2 shows the optimal distortion values found by GA (each is the average of 30 runs)

Table 2.2: Comparison of the average distortions obtained by the GA-based routing and exhaustive search

Topology	Topology 1	Topology 2	Topology 3	Topology 4
Network Size	10-node	10-node	15-node	15-node
Global Optimum	0.3308	0.2004	0.3863	0.2969
GA (average)	0.3330	0.2004	0.3937	0.2972
GA (std. dev.)	7.6e-6	0	2.8e-5	2.9e-6
Lower Bound	0.2810	0.1832	0.3527	0.2444

and by exhaustive search for two 10-node and two 15-node networks. We find that the solutions found by GA are very close to the global optimum in all cases. In addition, the deviation of the GA results is negligibly small, indicating that GA executions produce near-optimal or optimal solutions. The average computational time for GA is 0.29 s for the 10-node network (about 60 generations) and 0.39 s for the 15-node network (about 70 generations) on a Pentium 4 2.4 GHz computer (512 MB memory) with MATLAB 6.5. For exhaustive search, the average computational time is 58.7 s for the 10-node case and 1877 s for the 15-node case.

We also compute the lower bound using ALG-LB for each of the networks. The results are given in the last row of Table 2.2. We observe that the lower bounds are tight in all of the cases, i.e., within 8% to 16% of the global optimum.

2.5.2 Comparison with Trajectory Methods

To compare the GA-based approach to trajectory methods, we implemented simulated annealing (SA) and tabu search (TS), both of which have been used in solving certain networking problems. SA was initially motivated by an analogy between the way a piece of metal cools and freezes into a minimum energy crystalline structure (annealing process), and the

search for a minimum in a more general system [2]. When SA explores the solution space, it accepts a non-improving solution with a probability, which decreases with iterations. We use a probabilistic acceptance function:

$$Pr\{\bar{x} \leftarrow \hat{x}\} = \begin{cases} 1, & \text{if } D(\hat{x}) < D(\bar{x}) \\ \exp\{-\frac{|D(\hat{x}) - D(\bar{x})|}{c_k}\}, & \text{otherwise,} \end{cases} \tag{2.14}$$

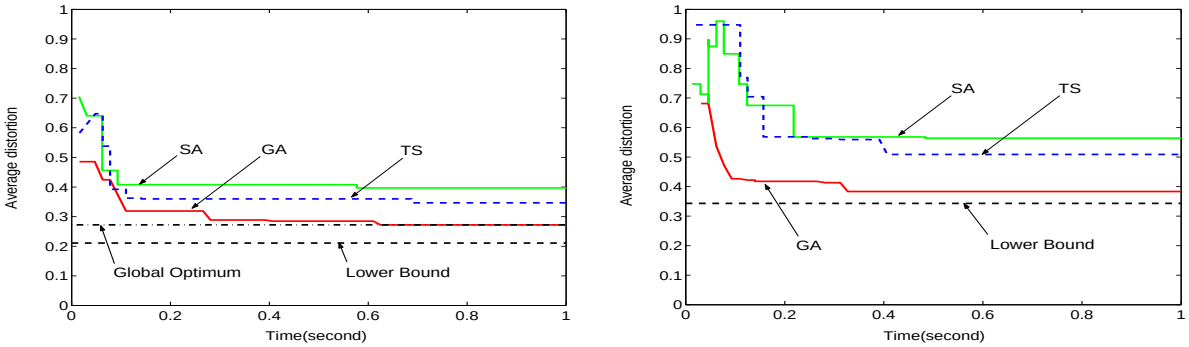
where c_k is a control parameter analogous to temperature in a physical system, $\bar{x}$ is the current solution, and $\hat{x}$ is a perturbation of $\bar{x}$. The fashion in which c_k is manipulated is called the *cooling schedule*. The following cooling schedule is used in our experiments [2]:

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. We set $\omega = 0.99$ for all experiments reported in this chapter.

Compared with SA, TS 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* to prevent from returning to recently visited solutions, therefore avoiding endless cycling and possibly forcing the search process to accept non-improving solutions [25]. In our experiments, we use a tabu list of 5 for small networks (e.g., 10-node networks) and 10 for large networks (e.g., 50-node networks). The tabu list is implemented using a First-In-First-Out queue. 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.

In Figure 2.8, we plot the evolution of distortion values obtained by GA, SA, and TS for a 10-node network and a 50-node network, respectively. All the three metaheuristics are terminated after running for 1 s. Upon termination, GA has evolved 210 generations in Figure 2.8(a) and 75 generations in Figure 2.8(b); SA ran for 1500 iterations in Figure 2.8(a) and 700 iterations in Figure 2.8(b); TS ran for 1050 iterations in Figure 2.8(a) and 550



(a) Distortion evolution for a 10-node network.

(b) Distortion evolution for a 50-node network.

Figure 2.8: Comparison of distortion evolution of three metaheuristic methods.

generations in Figure 2.8(b). GA has fewer iterations than SA and TS, due to its higher computational complexity (see Section 2.6.2 for more discussions). For both networks, the best distortion values found by GA are evidently much better than those by SA or TS. In Figure 2.8(a), GA converges to the global optimal very quickly, while both SA and TS are trapped at local optima (i.e., no further decrease in distortion value after hundreds of iterations). The same trend can be observed in the 50-node network case shown in Figure 2.8(b), although the global optimum is not obtainable here.

An interesting observation from Figure 2.8 is that for GA, the biggest improvement in distortion is achieved in the initial iterations, while the improvement gets smaller as GA evolves more generations. Also, note that the SA and TS curves increase at some time instances (e.g, the TS curve at 0.06 s in Figure 2.8(a) and the SA curve at 0.08 s in Figure 2.8(b)), which implies that a non-improving solution is accepted in order to escape from local minima. We also plot the lower bounds derived using ALG-LB in the figures, which are quite tight in both cases.

In addition to providing much better solutions, another strength of GA over trajectory methods is that multiple "good" solutions can be found after a single run. Such extra good paths can be used as alternative (or backup) paths if needed.

2.5.3 Comparison with Network-Centric Approaches

In this section, we compare our GA approach with network-centric routing approaches. In particular, we implement two popular network-centric multipath routing algorithms, namely k-shortest path (SP) routing (with $k = 2$ or 2-SP) [22] and disjoint path routing, Disjoint Pathset Selection Protocol (DPSP) [46]. Our 2-SP implementation uses hop count as routing metric such that two shortest paths are found. In our DPSP implementation, we set the link costs to $\log(1/p_{ij})$, for all $\{i, j\} \in E$, such that two disjoint paths having the highest end-to-end success probabilities are found. We compare the performance of our GA-based multipath routing with these two algorithms over a 50-node ad hoc network using a real video clip.

There are many ways to generate MD video (see [27] for an excellent survey). We choose a time-domain partitioning coding scheme, where two descriptions are generated by separating the even and odd-numbered frames and coding them separately, as shown in Figure 2.9. 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 frames). A 10% macroblock level intra-refreshment is used, which has been found to be effective in suppressing error propagation for the range of loss rates considered. This simple time-domain partitioning method is widely used in many video streaming studies [7, 8, 12, 17, 38]. Compared with a traditional single description coder, this coder has a comparable computational complexity. Its coding efficiency is slightly lower than a single description coder, due to the fact that a longer motion prediction distance is used. However, this reduced coding efficiency is well justified by the resulting enhanced error resilience. The quarter common intermediate format (QCIF) [176×144 Y pixels/frame, 88×72 Cb/Cr pixels/frame] sequence "Foreman" (400 frames) is encoded at 15 fps for each description. Each Group of Blocks (GOB) is carried in a different packet. The received descriptions are decoded and PSNR values of the reconstructed frames computed. When a GOB is corrupted, the decoder applies a simple error concealment scheme by copying from the corresponding slice in the most recently received frame that has not beed corrupted.

The quality of the paths found by the algorithms are presented in Table 2.3. The 2-SP algo-

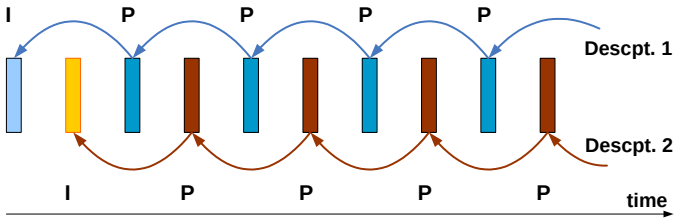


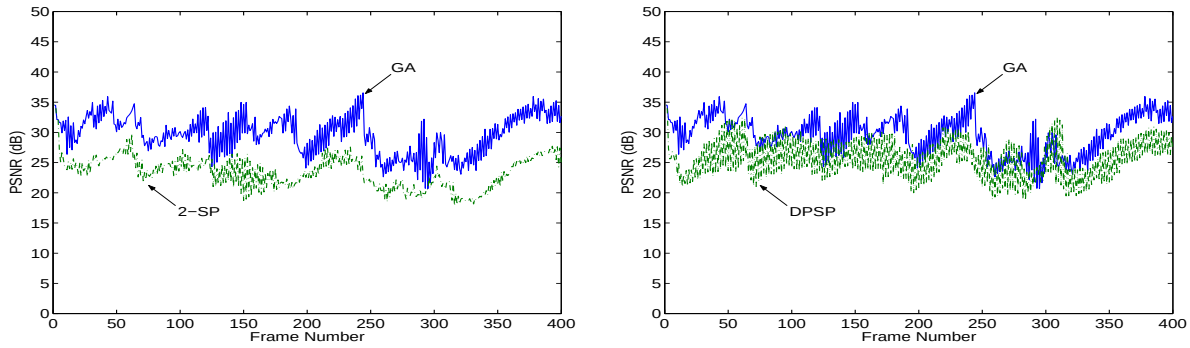
Figure 2.9: The MD coding scheme used in the numerical examples.

Table 2.3: Comparison of GA and Network-centric Routing

-	$\mathcal{P}_1$ Loss Ratio	$\mathcal{P}_2$ Loss Ratio	Desc. 1 Bandwidth	Desc. 2 Bandwidth	Average PSNR
GA	0.994	0.952	350 Kb/s	350 Kb/s	29.71 dB
2-SP	0.798	0.782	100 Kb/s	200 Kb/s	23.42 dB
DPSP	0.965	0.793	100 Kb/s	100 Kb/s	25.65 dB

rithm has the worst performance in terms of path success probabilities. The DPSP algorithm has an improved success probability performance since it uses link success probabilities in routing. However, it may sacrifice path bandwidth while pursuing low loss paths. As a result, it produces the lowest end-to-end bandwidths. We observe that our GA-based routing yields paths with much higher end-to-end success probabilities and end-to-end bandwidths, resulting in greatly improved video quality.

The PSNR curves of the received video frames are plotted in Figure 2.10. We observe that the PSNR curve obtained by GA is well above those obtained by the aforementioned network-centric routing approaches. Using GA, the improvements in average PSNR value over 2-SP and DPSP are 6.29 dB and 4.06 dB, respectively. We also experiment with an improved 2-SP algorithm where link success probabilities are used in routing (as in DPSP). Even in this case, our GA-based routing still achieves a 1.27 dB improvement over this enhanced 2-SP version, which is still significant in terms of visual video quality. To further illustrate the significance of the improvements, we plot reconstructed Frame 235 in Figure 2.11. We observe that the image delivered by GA has the best quality, while the video frames delivered



(a) GA-based algorithm versus 2-SP. (b) GA-based algorithm versus DPSP.

Figure 2.10: PSNR curves of received video sequences.

(a) Original. (b) GA-based routing. (c) 2-SP. (d) DPSP.

Figure 2.11: Frame 235 from the original and decoded video sequences.

by 2-SP and DPSP are barely recognizable. The relatively low quality in these latter two frames are caused by low encoding bit rates (which are determined by the bandwidths of the paths) and high packet loss rates (which are determined by the reliabilities of the paths), see (2.1) and (2.2). Obviously, our GA-based multipath routing offers significantly improved performance at the application layer over the two network-centric algorithms.

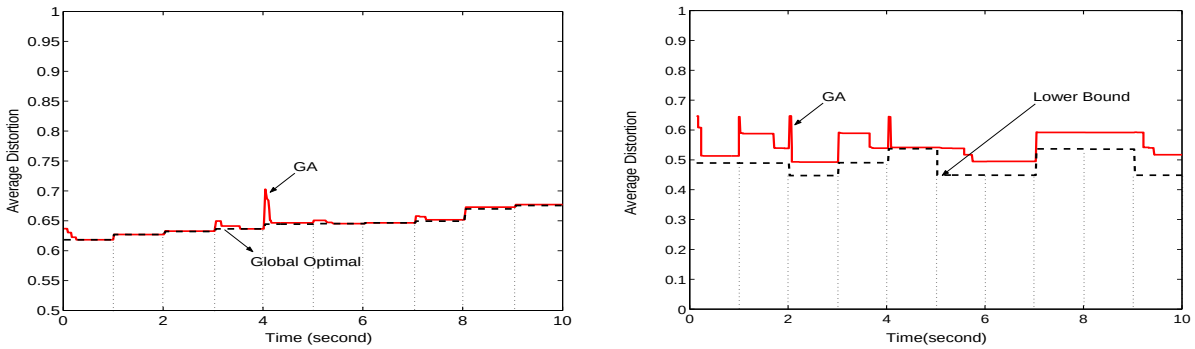
An inherent issue of transmitting MD video over multiple paths is that when the paths are unbalanced, e.g., either in bandwidth or in loss characteristics, the streams may have different qualities. When interlaced and displayed, such unbalanced streams may cause large variations in frame quality and yield low subjective quality (although a high objective quality, e.g., average PSNR, may always be achieved). In Problem OPT-MM, due to the symmetry in (2.1), our GA-based routing attempts to find a balanced pair of paths while minimizing D. For example, the two paths found by GA as in Table 2.3 have similar success probabilities and exactly the same bandwidth, resulting in relatively balanced descriptions. In the case

when the descriptions are highly unbalanced, the problem can be further alleviated by using an advanced MD coder that is capable of producing descriptions with unbalanced rates (but with relatively equal qualities) [7], or by striping packets of the descriptions across multiple paths to make losses of the descriptions relatively even.

2.5.4 Consideration of Mobility and Topology Changes

In addition to link/node failures, other characteristics associated with wireless ad hoc networks are node mobility and dynamic network topology changes. Figure 2.12 shows our investigations for a 10-node network in a 500 m $\times$ 500 m square region and for a 50-node network in a 1200 m $\times$ 1200 m square region. The transmission region is set to 250 m and every node moves at a speed of 10 m/s in both networks. We used the *Billiard* mobility model, where a node moves at a constant speed but at random directions. When it hits the boundary, it bounces back to another random direction. In all of the experiments, link state database at the source node is updated at 1 s intervals (marked by the vertical dashed lines in the figures).

In Figure 2.12(a), we plot the evolution of the average distortion obtained by GA for the 10-node network, as well the global optimal solution computed via an "offline" exhaustive search. GA is able to track closely the optimal solution during this experiment: it converges to the global optimum within a few iterations after each link state update. In Figure 2.12(b), we plot the GA average distortion evolution for the 50-node network, as well as the lower bound for each new topology as references (global optima are not obtainable, since an exhaustive search is impractical for such large-sized network). Again, we observe that GA can adapt and track the varying topology very well. After each update, a previous optimal solution may not be optimal anymore, resulting in a sudden rise in distortion. However, after some iterations, the GA distortion can quickly settle to a much lower value. This result is, again, due to the intrinsic self-adaptive nature of GA, as well as the continuous nature of node mobility. When nodes move, the fitness landscape varies gradually. It is very likely that an optimum solution in the updated network topology lies in the neighborhood of an optimal solution of the previous topology. GA is effective in exploring the neighborhoods of its current solutions



(a) 10-node network. (b) 50-node network.

Figure 2.12: Evolution of distortion values obtained by our GA-based algorithm. The dashed vertical lines mark time instances when link state updates were received.

to identify improved solutions.

2.6 Distributed Implementation

In addition to a centralized version of the GA-based multipath routing, we also investigate how to develop an effective distributed implementation for practical systems. Our approach is to implement our cross-layer routing algorithms by incorporating some proven ideas from existing network layer ad hoc routing algorithms.

2.6.1 A Distributed Implementation Architecture

Existing routing protocols can be roughly categorized as *proactive*, whereby a consistent and up-to-date view of the network is always maintained, and *reactive*, whereby route discovery is performed on-demand. We believe that the proposed GA-based routing is most suitable to be implemented within the proactive ad hoc routing paradigm. Our choice is motivated by the following two important and practical considerations. First, it is necessary to make quick routing decisions whenever a new MD 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

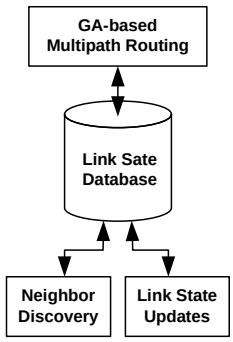


Figure 2.13: A distributed implementation architecture of the GA-based multipath routing.

(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.

Since an effective operation of cross-layer multipath routing 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. To this end, we find that the class of link state routing protocols, such as the *Optimized Link State Routing* protocol (OLSR) [19] and *Topology Dissemination Based on Reverse-Path Forwarding* (TBRPF) [43], are suitable for this purpose. Figure 2.13 depicts an implementation architecture of the proposed GA-based multipath routing at an ad hoc node. This implementation works in a completely distributed manner, and thus does not depend on any central entity in the network. We briefly describe the operations of its key modules.

Neighbor Discovery

Each node should detect its neighbor nodes to which a wireless link exists. This can be accomplished by periodically broadcasting HELLO messages containing information about neighbors and their link status to its one-hop neighbors. 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 [1]) can be used for this purpose.

Link State Updates

As with other link state routing protocols, our implementation also periodically broadcasts link state advertisements (LSA) to the entire network to distribute network topology information and link statistics. To reduce the control traffic overhead, we can use the *MultiPoint Relay* (MPR) technique [19] to minimize the flooding of control messages. With MPR, each node in the network selects a subset of its one-hop neighbors for retransmitting its packets, such that (1) all of its two-hop neighbors can be covered by the retransmissions, and (2) the number of retransmissions is minimized. It has been shown that the MPR technique can effectively suppress control traffic overhead. Furthermore, the denser and larger a network is, the more reduction can be achieved [19].

Link State Database

When a node receives LSAs from other nodes, it examines the message and extracts new link state information from the messages. 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. Each link item (along with its corresponding metrics) is associated with a sequence number, which is set to the sequence number of the LSA message from which this link item was learned. Therefore, a stale item will be overwritten by a fresher item, making the link state database consistent and up-to-date.

GA-based Multipath Routing

When the link state database is available, we can apply the GA-based multipath routing to compute a set of routes from this node to other nodes in the network, as discussed in the previous sections. When the paths are computed, the next question is how to establish the 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 [32]. 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 use routing tables in packet forwarding, and intermediate nodes

do not need to maintain flow information for the video session. 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* [30] 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.

2.6.2 Performance Issues

Link state routing protocols are quite successful in IETF standardization. In fact, two of the three existing RFCs for MANET routing, OLSR and TBRPF, 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 MPRs, as in OLSR [19], or by reporting partial topology information in the LSAs and using "differential" HELLO messages that report only changes in neighbor status, as in TBRPF [43]. Our initial distributed implementation experience shows that, with proper design extension and changes, proactive distributed protocols can be used to build and maintain a link state database for the GA-based distributed implementation.

Another practical consideration for the GA-based approach is computation time. Since GA evolves a population of solutions over a large number of generations, it may have higher computational complexity than trajectory methods or network-centric multipath routing schemes. Our numerical results show that a properly designed GA can compute very good routes for ad hoc networks with small and moderate sizes (e.g., 50 nodes as in Figure 2.8) in several hundred milliseconds, which are fast enough for practical uses. Incidentally, it has been shown in [3] that a GA-based algorithm is faster than Dijkstra's algorithm as the network size increases. Furthermore, our numerical results show that with GA, the great-

est improvement in fitness value is achieved after a small number of generations, and the improvement gets much smaller after these initial generations (see Figure 2.8). Therefore, there is a trade-off between solution optimality and computation time. For a delay-sensitive real-time application, GA can compute a set of "good" routes very quickly, and the application can use these "good" routes after a very small delay. As GA continues to evolve, the routes used by the application can be updated with newly computed better set of routes for enhanced performance.

2.7 Related Work

Multipath routing has been an active research area over the years. Various algorithms have been proposed to compute k-shortest paths [22], node- or link-disjoint paths [46, 58], or braided multiple paths [41]. Given multiple paths, traffic proportioning schemes are also designed to disperse traffic to the paths for an increased end-to-end throughput, load balancing, and fast failure recovery [28]. In the area of wireless ad hoc networks, many existing routing protocols are multipath-capable (e.g., Terminode Routing [15]). However, most of these multipath routing algorithms are *network-centric*: they do not explicitly address application layer performance issues from a cross-layer perspective, as illustrated in Section 2.5.

The problem of multipath routing for multimedia communications has recently been explored in [8,12]. In [12], Begen *et al.* studied the multipath routing problem in the context of overlay networks. Although path selection is formulated as an optimization problem that minimizes video distortion, it is actually solved by an *exhaustive search* over the exponential solution space. The fast heuristic algorithm proposed in [13] attempts to speed up the computation by resorting to infrastructure support from the underlying network, which is not available in ad hoc networks.

The potential of GA in addressing networking problems has been recognized in recent years. For example, GA has been explored to address various networking problems such as routing [3,10,60], admission control [62], channel assignment [24], network design [21], scheduling [42] and buffer management [23]. In particular, Ahn and Ramakrishna [3] applied GA to the

shortest path routing problem and compared its performance to Dijkstra's algorithm. The authors presented an elegant analysis on how to determine the optimal population size for the problem studied. These efforts have made the important step in exploring the potential of GA for network optimization. The research in this chapter builds upon these efforts and aims to explore GA's potential to address the more complex *cross-layer* optimization problem. This problem is more substantial than the network-centric GA problems, since it not only requires knowledge at the network layer, but also, an understanding at the application layer (i.e., video coding capability) in order to fully exploit the design and optimization space across the layers.

2.8 Summary

In this chapter, we studied the important problem of optimal multipath routing for MD video. We formulated the multipath routing problem following an application-centric cross-layer approach. We found that a GA-based approach is eminently suitable to address such optimal routing problems, which involve complex objective functions and exponential solution spaces. We consequently designed a GA-based algorithm to address this multipath routing problem and found that this approach provides near-optimal results. We also developed a tight lower bound for distortion, which can be used to evaluate the performance of a GA-based solution as well as to establish its termination criteria. Although a GA-based approach is a centralized algorithm in nature, we showed that it is amenable for distributed implementation in ad hoc routing protocols. This work provides an important methodology for addressing complex cross-layer optimization problems, particularly those involving the application and network layers.

Chapter 3

MD Video Multicast in Wireless Ad Hoc Networks

3.1 Introduction

In this chapter, we study the important and more difficult problem of *multicasting* MD video in wireless ad hoc networks. Although multicast routing for ad hoc networks has been explored in prior research (see, e.g., [51, 52]), previously proposed approaches are not quite suitable for MD video applications for several reasons. First, many existing algorithms use a single network-layer metric in routing. But video quality is usually determined by more than one metric, such as loss, delay, jitter, and available bandwidth. Optimizing one metric could lead to significant degradation in the other [34]. As a result, single network-layer metric based approaches do not necessarily offer optimal video quality. Second, many existing protocols are based on Dijkstra's algorithm, which requires additive routing metrics. For multimedia-centric routing, there exists a highly complex relationship pertaining to the contribution of any link to the video quality, which depends in general on the other links that are included in a fashion that has no particular structural property such as additivity and convexity. Consequently, there remain important open problems in multicast routing for MD video in wireless ad hoc networks.

We present an efficient MD video multicast scheme aiming to cope with the above issues. With this scheme, a number of multicast trees are used, with each multicast tree supporting one video description. Further, we propose each description be coded into a base layer and a number of enhancement layers. Packets belonging to the same description from both the base layer and enhancement layer are therefore transmitted on the same tree. We show that this MD video multicast approach can effectively deal with frequent link failures and diverse link qualities in wireless ad hoc networks. First, since a receiver is a member of multiple trees, it is connected to the source node via multiple paths. When a path is broken due to a failed link, the other path(s) may still be in a good condition, assuming link failures in the network are independent. As a result, such *path diversity* provides enhanced error resilience to an MD video session [6, 38]. Second, due to highly diverse wireless links, the end-to-end bandwidths from the sender to the receivers (called *path bandwidth* throughout this chapter) are also highly diverse. It is desirable to code each description into a number of layers, so that a receiver with a high path bandwidth can subscribe the maximum number of layers of the corresponding description that are allowed by its path bandwidth for best video quality. Figure 3.1 illustrates the proposed multicast scheme for MD video. Since the video session is coded into two descriptions, two multicast trees are needed. Note that each video description is further coded into a base layer and an enhancement layer. In the example, the path from the sender to Receiver 2 in Tree 1 has enough path bandwidth for both layers of Description 1. But in Tree 2, the path bandwidth between the sender and Receiver 2 only has enough bandwidth for the base layer of Description 2.

Within the proposed multi-tree multi-layer scheme for MD multicast, the most difficult problem lies in how to find multiple trees such that the overall video quality is optimized. To address this problem, we take an application-centric, cross-layer approach to formulate MD video multicast routing as a combinatorial optimization problem. In contrast to previous work [48, 51, 52, 63], our objective is to optimize the application layer performance metric, i.e., video distortion. It turns out that the optimization problem is highly complex and is expected to be NP-complete. As a result, it is desirable to develop efficient heuristic algorithms in practice. We find that *Genetic Algorithms* (GA) [9] are eminently suitable for addressing problems of this type, which have complex objective functions and large,

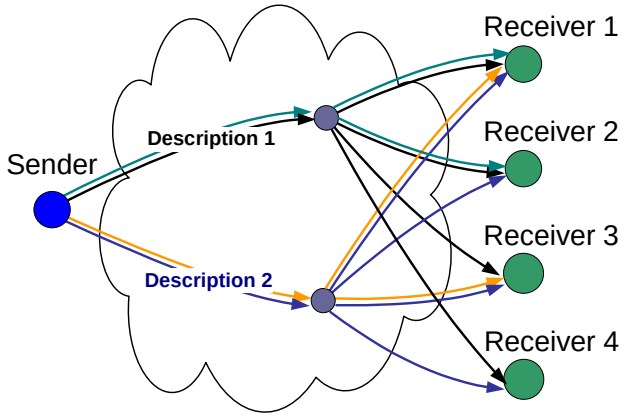


Figure 3.1: MD video multicast using two trees.

unstructured combinatorial solution spaces. We construct a GA-based solution procedure, and demonstrate its efficacy through extensive performance studies.

The remainder of this chapter is organized as follows. In Section 3.2, we formulate the problem of finding a pair of optimal multicast trees. In Section 3.3, we present a GA-based metaheuristic solution procedure. Section 3.4 presents our performance studies of the proposed approach. Related work is discussed in Section 3.5, and Section 3.6 concludes the chapter.

3.2 Problem Formulation

We model an ad hoc network with N nodes and L links as a directed graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$. As we discussed in Chapter 2, accurate and computationally efficient characterization of an end-to-end path is very difficult and still remains an open problem. So we still focus on the network layer characteristics in this chapter, assuming that the physical layer and MAC layer dynamics are translated into network layer parameters. For example, we could characterize a link $\{i, j\} \in \mathcal{L}$ by the available capacity of link $\{i, j\}$, c_{ij}; the probability with which link $\{i, j\}$ fails, p_{ij}; the fixed, or the minimum delay of link $\{i, j\}$, τ_{ij}; the mean delay of link $\{i, j\}$, t_{ij}; and the jitter of link $\{i, j\}$, δ_{ij}^2. We focus on bandwidth, failure probability, and delay because these are the key characteristics of wireless links, as well as key factors

Table 3.1: Notation

$\mathcal{G}\{\mathcal{N}, \mathcal{L}\}$:	graph representation of the network
$\mathcal{N}$:	the set of vertices in the network
$\mathcal{L}$:	the set of edges
s :	source node
$\mathcal{M}$:	the set of receivers
r :	a tagged receiver
$\{i, j\}$:	a wireless link from node i to j
c_{ij} :	available capacity of link $\{i, j\} \in \mathcal{L}$
p_{ij} :	probability with which link $\{i, j\}$ fails
τ_{ij} :	fixed, or minimum delay of link $\{i, j\}$
t_{ij} :	mean delay of link $\{i, j\}$
δ_{ij}:	jitter of link $\{i, j\}$
d_0 :	distortion when both descriptions are received
d_h :	distortion when only Description h is received
σ^2 :	variance of the video source
$\mathcal{T}_h$:	the h-th multicast tree
$\mathcal{P}_r^h$:	the path from s to r in Tree h
B_{base}^h :	bandwidth of the base layer of Description h
B_{tot}^h :	total bandwidth Description h
R_{base}^h :	bit rate of the base layer of Description h
R_{tot}^h :	total bit rate of Description h
P_{00}^b :	probability of receiving both base layers
P_{00}^e :	probability of receiving both enhancement layers
P_{01}^b :	probability of receiving Desc. 1's base layer only
P_{01}^e :	probability of receiving Desc. 1's enhancement layer only
P_{10}^b :	probability of receiving Desc. 2's base layer only
P_{10}^e :	probability of receiving Desc. 2's enhancement layer only
P_{11}^b :	probability that both base layers are lost
P_{11}^e :	probability that both enhancement layers are lost
D_r :	the average distortion of Receiver r
x_{ij}^h :	routing index variable associated with link $\{i, j\}$ in Tree h
u_i^h :	routing index variable associated with node i in Tree h
Δ_r :	the decoding deadline of Receiver r

that determine video distortion (see Section 3.2.1). Table 3.1 lists the notation used in the chapter.

3.2.1 Rate-Distortion Model for MD Video

Similar to the problem in last chapter, a rate distortion model describes the relationship between the bit rate and the achieved distortion. We consider a sender generating two descriptions, each being encoded into two layers. For a receiver, denote d_h the achieved distortion when only Description h is received, $h = 1, 2$, and d_0 the distortion when both descriptions are received. Note that when both descriptions are lost, the distortion is σ^2. Clearly, d_0, d_1, and d_2 are functions of the description rates. Let R^h_{base} be the base layer rate and R^h_{tot} the total rate (i.e., the aggregate rate of the base and enhancement layer) of Description h, $h = 1, 2$. For the two base layers (each from a description), denote P^b_{00} the probability of receiving both base layers; P^b_{01} the probability of receiving Description 1's base layer only; P^b_{10} the probability of receiving Description 2's base layer only; and P^b_{11} the probability of receiving neither of these base layers. For the two enhancement layers (each from a description), we define the probabilities P^e_{00}, P^e_{01}, P^e_{10}, and P^e_{11} in the same fashion.

As will be shown in Section 3.2.2, we choose the base layer rates for the two descriptions such that any receiver can have enough path bandwidths for both base layers. However, a receiver having a low path bandwidth from a tree may not be able to receive the enhancement layer from that tree (e.g., the enhancement layer is dropped at an upstream bottleneck link in the tree). According to a receiver's two path bandwidths and the rates of the layers, the layers it can receive can be classified into the following four cases (see the example in Figure 3.1):

- Case I: it can receive base layers and enhancement layers from both trees (e.g., Receiver 1 in Figure 3.1);
- Case II: it can receive base and enhancement layers from Description 1 but only base layer from Description 2 (e.g., Receiver 2 in Figure 3.1);
- Case III: it can receive base and enhancement layers from Description 2 but only base

layer from Description 1 (e.g., Receiver 3 in Figure 3.1);

- Case IV: it can only receive the base layers from the two descriptions (e.g., Receiver 4 in Figure 3.1).

Then, for Case I, the average distortion can be computed as:

$$\begin{aligned} D_r &= P_{00}^b \left\{ P_{00}^e d_0(R_{tot}^1, R_{tot}^2) + P_{01}^e d_0(R_{tot}^1, R_{base}^2) + P_{10}^e d_0(R_{base}^1, R_{tot}^2) + P_{11}^e d_0(R_{base}^1, R_{base}^2) \right\} + \\ & P_{01}^b \left\{ [P_{00}^e + P_{01}^e] d_1(R_{tot}^1) + [P_{10}^e + P_{11}^e] d_0(R_{base}^1) \right\} + \\ & P_{10}^b \left\{ [P_{00}^e + P_{10}^e] d_2(R_{tot}^2) + [P_{01}^e + P_{11}^e] d_2(R_{base}^2) \right\} + P_{11}^b \sigma^2, \end{aligned} \tag{3.1}$$

For Case II, the average distortion can be computed as:

$$\begin{aligned} D_r &= P_{00}^b \left\{ P_{01}^e d_0(R_{tot}^1, R_{base}^2) + P_{11}^e d_0(R_{base}^1, R_{base}^2) \right\} + P_{01}^b \left\{ P_{01}^e d_1(R_{tot}^1) + P_{11}^e d_1(R_{base}^1) \right\} + \\ & P_{10}^b d_2(R_{base}^2) + P_{11}^b \sigma^2. \end{aligned} \tag{3.2}$$

The average distortion for Case III can be computed similarly as (3.2) due to symmetry. For Case IV, the average distortion can be computed as:

$$D_r = P_{00}^b d_0(R_{base}^1, R_{base}^2) + P_{01}^b d_1(R_{base}^1) + P_{10}^b d_2(R_{base}^2) + P_{11}^b \sigma^2. \tag{3.3}$$

Similarly, we have the rate-distortion region for two-description video [4] as

$$\begin{cases} d_0(R_1, R_2) = \frac{2^{-2(R_1+R_2)}}{2^{-2R_1} + 2^{-2R_2} - 2^{-2(R_1+R_2)}} \cdot \sigma^2 \\ d_1(R_1) = 2^{-2R_1} \cdot \sigma^2 \\ d_2(R_2) = 2^{-2R_2} \cdot \sigma^2. \end{cases} \tag{3.4}$$

3.2.2 Computing End-to-End Statistics

Consider a multicast session with sender s and a set of receivers $\mathcal{M}$. The session uses two trees $\{\mathcal{T}_1, \mathcal{T}_2\}$, each rooted at sender s. Before formulating the optimal multicast routing problem, we need to compute the average distortion D_r of a receiver $r \in \mathcal{M}$ as a function of link statistics for a *given* pair of trees. Note that we do not mandate disjoint trees, which will unnecessarily shrink the solution space for optimization.

End-to-End Delay: For a tagged receiver $r \in \mathcal{M}$, let the path from the source s to r in tree $\mathcal{T}_h$ be $\mathcal{P}_r^h$, $h = 1, 2$. To model end-to-end delay, we follow the approach in [18]. For receiver r, the end-to-end delay on its path $\mathcal{P}_r^h$, denoted as t_r^h, could be modeled as a "shifted" Gamma distribution:

$$y(t_r^h) = \frac{\alpha_r^h}{\Gamma(n_r^h)} \left[\alpha_r^h \cdot (t_r^h - \tau_r^h)\right]^{n_r^h - 1} e^{-\alpha_r^h \cdot (t_r^h - \tau_r^h)}, \tag{3.5}$$

for $t_r^h \geq \tau_r^h$, $h = 1, 2$. The end-to-end delay from sender s to receiver r can be interpreted as the total delay of going through n_r^h nodes, each with a processing delay of τ_r^h/n_r^h and an exponentially distributed queueing delay (with mean α_r^h). The parameters of the shifted Gamma distribution can be estimated from the link statistics as:

$$\begin{cases} \tau_r^h = \sum_{\{i,j\} \in \mathcal{P}_r^h} \tau_{ij} & h = 1, 2, \forall r \in \mathcal{M}, \\ \alpha_r^h = \frac{\sum_{\{i,j\} \in \mathcal{P}_r^h} t_{ij} - \sum_{\{i,j\} \in \mathcal{P}_r^h} \tau_{ij}}{\sum_{\{i,j\} \in \mathcal{P}_r^h} \delta_{ij}^2} & h = 1, 2, \forall r \in \mathcal{M}, \\ n_r^h = \frac{(\sum_{\{i,j\} \in \mathcal{P}_r^h} t_{ij} - \sum_{\{i,j\} \in \mathcal{P}_r^h} \tau_{ij})^2}{\sum_{\{i,j\} \in \mathcal{P}_r^h} \delta_{ij}^2} & h = 1, 2, \forall r \in \mathcal{M}. \end{cases}$$

Success Probabilities: As indicated in (3.1) and (3.4), the video distortion is the highest when both descriptions are lost, since σ^2 is generally much larger than d_0, d_1, and d_2. In order to reduce the possibility of simultaneously losing both descriptions, the correlation of the loss processes of the two descriptions should be minimized at best [8]. It has been shown in previous work, e.g., [50], that packet interleaving can effectively reduce such a correlation and achieve a significantly improved video quality at the cost of an additional fixed interleaving delay. Consequently, we assume that video packets are *interleaved* with an appropriate interval (i.e., larger than the time-scale of link dynamics) before transmission, such that packet losses within the same frame are relatively independent.

Then, the probability of receiving a video packet before its decoding deadline Δ_r by receiver r (excluding the constant interleaving delay) from tree h is:

$$q_r^h = \left[\prod_{\{i,j\} \in \mathcal{P}_r^h} (1 - p_{ij})\right] \cdot \Pr(t_r^h \leq \Delta_r), h = 1, 2.$$

For Case I where the receiver can receive base and enhancement layers for both descriptions,

the joint probabilities of receiving the layers can be computed as:

$$\begin{cases} P^b_{00} = q^1_r \cdot q^2_r \\ P^b_{01} = q^1_r \cdot (1 - q^2_r) \\ P^b_{10} = (1 - q^1_r) \cdot q^2_r \\ P^b_{11} = (1 - q^1_r) \cdot (1 - q^2_r) \end{cases} \text{and} \begin{cases} P^e_{00} = q^1_r \cdot q^2_r \\ P^e_{01} = q^1_r \cdot (1 - q^2_r) \\ P^e_{10} = (1 - q^1_r) \cdot q^2_r \\ P^e_{11} = (1 - q^1_r) \cdot (1 - q^2_r). \end{cases} \tag{3.6}$$

For the remaining three cases, the probabilities of receiving the base layers are the same as in Case I. For the probabilities of receiving the enhancement layers, we have the following according to which enhancement layer can be received: (i) Case II: $P^e_{00} = 0$, $P^e_{01} = q^1_r$, $P^e_{10} = 0$, and $P_{11} = (1 - q^1_r)$; (ii) Case III: $P^e_{00} = 0$, $P^e_{01} = 0$, $P^e_{10} = q^2_r$, and $P_{11} = (1 - q^2_r)$; (iii) Case IV: $P^e_{00} = 0$, $P^e_{01} = 0$, $P^e_{10} = 0$, and $P^e_{11} = 1$.

Optimal Video Rates: Consider a receiver r and its two associated root paths $\{\mathcal{P}^1_r, \mathcal{P}^2_r\}$. We can classify the links within the two paths as the set of joint links, denoted as $\mathcal{J}(\mathcal{P}^1_r, \mathcal{P}^2_r)$, and the sets of disjoint links, denoted respectively as $\bar{\mathcal{J}}(\mathcal{P}^h_r)$, $h = 1, 2$. The minimum bandwidth of $\mathcal{J}(\mathcal{P}^1_r, \mathcal{P}^2_r)$ is defined as:

$$B^{jnt}_r = \begin{cases} B(\mathcal{J}(\mathcal{P}^1_r, \mathcal{P}^2_r)) & \text{if } \mathcal{J}(\mathcal{P}^1_r, \mathcal{P}^2_r) \neq \emptyset; \\ \infty & \text{otherwise,} \end{cases}$$

where $B(\mathcal{P}) \equiv \min_{\{i,j\} \in \mathcal{P}} \{c_{ij}\}$. Then, the path bandwidths of receiver r are:

$$\begin{cases} B^h_r = B(\mathcal{P}^h_r) & \text{if } \sum_{h=1}^{2} B(\mathcal{P}^h_r) \leq B^{jnt}_r, h = 1, 2; \\ B^1_r + B^2_r \leq B^{jnt}_r & \text{otherwise.} \end{cases} \tag{3.7}$$

The first case of (3.7) is for the situation when the joint links are not the bottleneck of the paths, while the second case of (3.7) is for the situation where one of the joint links is the bottleneck of both paths. In the latter case, we split the bandwidth of the shared bottleneck link in proportion to the mean success probabilities of the two root paths.

Once the path bandwidths are found, we need to determine the optimal bandwidths of the layers for each description. Clearly, all of the receivers should be able to receive the base layers in order to effectively decode the descriptions. Thus, we set the base layer bandwidth of Description h to:

$$B^h_{base} = \min_{r \in \mathcal{M}} \{B^h_r\}, \quad h = 1, 2. \tag{3.8}$$

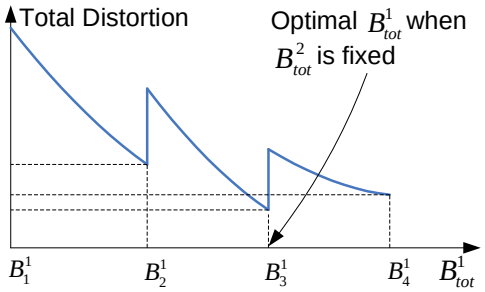


Figure 3.2: Optimal B^1_{tot} for a fixed B^2_{tot}.

The total bandwidth of Description h, B^h_{tot}, should be within the range $\left[B^h_{base}, \max_{r\in\mathcal{M}}\{B^h_r\}\right]$. For a chosen B^h_{tot}, receivers that satisfy $B^h_r \geq B^h_{tot}$ can receive both layers of Description h; other receivers with $B^h_r < B^h_{tot}$ can only receive the base layer of Description h.

It can be shown that the average distortion of a receiver, D_r, is a non-increasing function of the rate B^h_{tot}, $h = 1, 2$ [36]. Therefore, for a fixed B^h_{tot}, the total distortion of all receivers is a piece-wise non-increasing function of B^{3-h}_{tot} with discontinuous jumps at B^{3-h}_r, $r \in \mathcal{M}$. An example with four receivers is illustrated in Figure 3.2, where the total distortion is plotted as a function of B^1_{tot} for a fixed B^2_{tot}, assuming that $B^h_1 < B^h_2 < B^h_3 < B^h_4$, $h = 1, 2$. In this example, B^1_{base} is set to B^1_1, as given in (3.8). The total distortion is the highest when $B^1_{tot} = B^1_1$, since no enhancement layer for Description 1 can be received by any of the receivers. If we fix B^2_{tot} and increase B^1_{tot}, the total distortion keeps on decreasing, due to the monotonicity property of (3.1) [36]. When B^1_{tot} reaches B^1_2, there is a sudden increase in the total distortion, since Receiver 2 cannot receive the enhancement layer anymore, and so forth. We find that for a fixed B^2_{tot}, we only need to evaluate the total distortion at three points, i.e., $B^1_{tot} = B^1_r$, $r = 2, 3, 4$, in order to find the optimal B^1_{tot}, which is $B^1_{tot} = B^1_3$ in this example.

Figure 3.3 plots the total distortion for all feasible combinations of B^1_{tot} and B^2_{tot}. We find that the same monotonicity property holds true in this case. Due to the monotonicity property of (3.1), we only need to examine the total distortion at points $\{B^1_i, B^2_j\}$, $i, j \in \{2, 3, 4\}$, in order to find the optimal total rates that minimize the total distortion. In general, if there are K_h different path bandwidths in tree $\mathcal{T}_h$, $h = 1, 2$, we only need to evaluate the total distortion at $(K_1 - 1) \cdot (K_2 - 1)$ bandwidth combinations in order to find the optimal

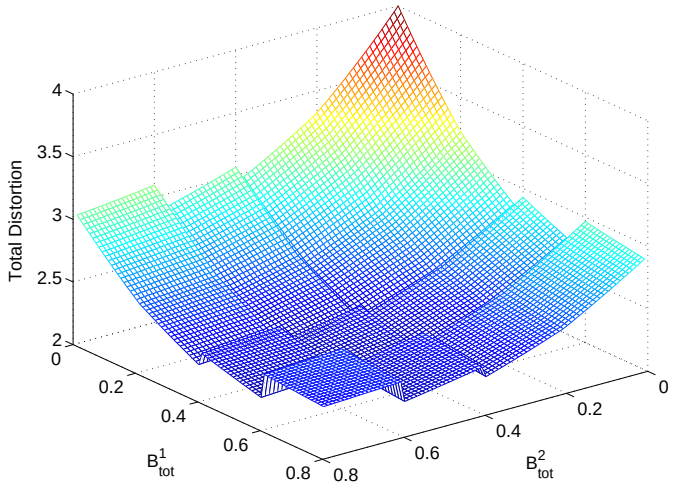


Figure 3.3: Total distortion for all combinations of B^1_{tot} and B^2_{tot}.

bandwidth for both descriptions. Note that the associated computational burden is low, since many wireless links operate at a small number of fixed bandwidths (e.g., $K_1 = K_2 = 4$ for a wireless LAN link).

After B^h_{tot} and B^h_{base}, $h = 1, 2$, are computed, the rates of the descriptions, in bits per pixel, can be determined as:

$$\begin{cases} R^h_{tot} = \rho \cdot B^h_{tot}, & h = 1, 2, \\ R^h_{base} = \rho \cdot B^h_{base}, & h = 1, 2, \end{cases} \tag{3.9}$$

where $\rho = \gamma \cdot W \cdot H \cdot R_f$ for a video with frame rate R_f and frame size $W \times H$; γ is a constant as determined by the chroma subsampling format (e.g., $\gamma = 1.5$ for the quarter common intermediate format (QCIF)).

3.2.3 The Multicast Routing Problem

To construct the problem formulation, we define the following two sets of variables in order to describe the choice of trees. For every link $\{i, j\} \in \mathcal{L}$, define

$$x^h_{ij} \stackrel{def}{=} \begin{cases} 1, & \text{if } \{i, j\} \in \mathcal{T}_h; \\ 0, & \text{otherwise}, h = 1, 2. \end{cases} \tag{3.10}$$

For every node $i \in \mathcal{N}$, define

$$u_i^h \stackrel{def}{=} \begin{cases} \text{number of hops from } s \text{ to } i,\ i \in \mathcal{T}_h \\ 0,\ i \notin \mathcal{T}_h, h = 1, 2. \end{cases} \tag{3.11}$$

Then, we formulate the optimal MD video multicast routing problem (OPT-MM) as follows.

OPT-MM Given a wireless ad hoc network $\mathcal{G}\{\mathcal{N}, \mathcal{L}\}$ and a multicast session $\{s, \mathcal{M}\}$, find a pair of trees $\{\mathcal{T}_1, \mathcal{T}_2\}$, such that the total video distortion of all the receivers in $\mathcal{M}$ is minimized. That is:

Minimize: $D = \sum_{r \in \mathcal{M}} D_r$ (3.12)

subject to:

$$x_{ij}^h \leq \sum_{k:\{k,i\} \in \mathcal{L}} x_{ki}^h,\ \forall i \neq s, \forall j \notin \{i, s\} : \{i, j\} \in \mathcal{L}, h = 1, 2 \tag{3.13}$$

$$\sum_{j:\{j,i\} \in \mathcal{L}} x_{ji}^h = 1,\ i \in \mathcal{M}, h = 1, 2 \tag{3.14}$$

$$\sum_{j:\{j,i\} \in \mathcal{L}} x_{ji}^h \leq 1,\ \forall i \in \mathcal{N} \backslash \mathcal{M}, h = 1, 2 \tag{3.15}$$

$$u_i^h - u_j^h + N \cdot x_{ij}^h \leq N - 1,\ \forall i, j \in \mathcal{N}, h = 1, 2 \tag{3.16}$$

$$x_{ij}^1 \cdot R_{tot}^1 + x_{ij}^2 \cdot R_{tot}^2 = (1 - \epsilon) \cdot \rho \cdot c_{ij}, \text{for some } \epsilon \in [0, 1], \forall \{i, j\} \in \mathcal{L} \tag{3.17}$$

$$x_{ij}^h \in \{0, 1\},\ \forall \{i, j\} \in \mathcal{L}, h = 1, 2 \tag{3.18}$$

$$u_i^h \in \{0, 1, \cdots, N - 1\},\ \forall i \neq s, h = 1, 2. \tag{3.19}$$

In Problem OPT-MM, constraints (3.13)–(3.16) represent the choice of a pair of trees. More specifically, Constraint (3.13) regulates the input-output relation of an arbitrary node, i.e., a node can forward a video stream only if it receives a video stream from its parent node; Constraint (3.14) guarantees that all member nodes are connected in the tree (with a single parent node); constraints (3.15) and (3.16) ensure the loop-free property of trees. Constraint (3.17) guarantees that the links are stable.

In its simplest form, i.e., when there is only one receiver in the group, Problem OPT-MM reduces to a QoS routing problem with two additive (delay and jitter), one multiplicative (loss), and one concave (bandwidth) metrics. Such a problem has been shown to be NP-complete in [61]. It is also worth noting that our problem is a far more complex variant of the

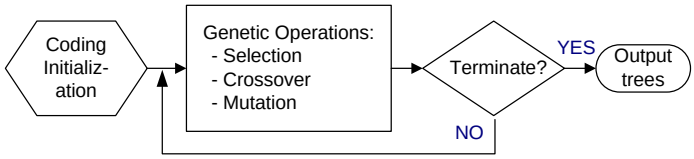


Figure 3.4: The GA-based multicast routing.

traditional Steiner tree design problem [31]. Therefore, we expect that Problem OPT-MM is also NP-complete. Furthermore, we note that for enhancing the model from the viewpoint of solvability, there exist several ways of tightening the representation of the constraints (3.13)–(3.19) with respect to the convex hull of feasible solutions as expounded by Sherali and Driscoll [57], for example. While this could be useful in a mathematical programming approach for solving Problem OPT-MM, or some relaxation thereof, we will be pursuing a metaheuristic procedure in the sequel.

3.3 A GA-based Metaheuristic Solution

Like our solution to the problem in Chapter 2, an effective strategy to address Problem OPT-MM is to view it as a "black-box" optimization problem and to explore an effective *metaheuristic* approach [16]. We find that *Genetic Algorithms* (GA) [9] are also quite suitable for addressing this complex problem. The basic framework for our GA-based multicast routing solution procedure is illustrated in Figure 3.4. We discuss each component in the sequel.

Coding and Initialization: Under GAs, it is critical to represent a solution in a proper form. In our approach, a solution for Problem OPT-MM is a pair of trees. In our implementation, we use an adjacency matrix A_h to describe the connectivity in tree h, $h = 1, 2$. That is, if $a_{ij}^h = 1$, link $\{i, j\}$ is in tree $\mathcal{T}_h$, $h = 1, 2$; if $a_{ij}^h = 0$, link $\{i, j\}$ is not in tree $\mathcal{T}_h$, $h = 1, 2$. Thus, we characterize a solution for our problem as a pair of such adjacency matrices.[1]

[1]The adjacency matrix data structure has an $O(N^2)$ storage requirement. The tree-list data structure as described in [11] can be used to reduce the storage to $O(3N)$.

With such encoding of solutions, we next generate an initial population. In order to make the individuals evenly distributed across the entire search space, we take a random construction approach. Starting with sender s, we randomly pick links emanating from s and include them (along with the "to-nodes" at which these links are incident) into the partial tree. Note that we only choose new links at each step for which exactly one end node is in the current partial tree in order to avoid loops, until all member nodes are included in the tree. After creating a number of trees in this manner, an individual can be created by randomly pairing the trees. An unbiased initial population is thus generated.

Genetic Operations: Genetic operations operate on the individuals according to their fitness values. The fitness of an individual $f(\bar{x}), \bar{x} \equiv \{\mathcal{T}_1, \mathcal{T}_2\}$, is closely related to its objective value. Since the objective is to minimize the total distortion, we define fitness as the inverse of the distortion value: $f(\bar{x}) = 1/D(\bar{x})$. This simple definition appears to work very well computationally in our simulations.

By the *selection* operation, we select the individuals that have more potential to produce better offsprings in terms of the fitness value. In our implementation, we experimented with both the *Tournament* scheme, where we randomly pick $k = 2$ individuals from the population and choose the one having a higher fitness value, and the *Roulette* scheme, where each individual is chosen with a probability in proportion to its fitness value. We found that the differences among the individual fitness values were generally small. With Roulette selection, the selection probabilities tend to be close to a uniform distribution, which decreases the *selection pressure* and slows down the convergence. However, using Tournament selection we are able to keep sufficient selection pressure as well as population diversity.

Crossover mimics the genetic mechanism of reproduction in the natural world, where genes from parents are recombined and passed to the offspring. For a pair of parent individuals, we first randomly choose a tree from each of them. Then, we find a common link in these two selected trees and exchange the corresponding subtrees connected by this link. After swapping the subtrees, we also check the two graphs obtained and make sure that they are feasible, i.e., they are loop free and include all the receiver nodes. If no such common link exists, we simply swap the two selected trees directly between the two parents. For two

parents, crossover is performed with a probability θ, called the crossover rate.

Mutation is the key ingredient of genetic algorithms. It is used to diversify the gene pool of the population, thus keeping the computation from being trapped at a local optimum. By randomly changing (mutating) one or more genes in an individual, the mutation produces a new individual with a random "jump" into a new area in the solution space. This operation therefore enables a wider exploration. In our algorithm, we randomly choose a link in a multicast tree (e.g., a link having a low bandwidth or a high failure probability). Removing this link results in two subtrees. Then, we add back a link or a branch that will connect the two subtrees with no loops. For an individual, the probability of being mutated is called the mutation rate μ.

Termination and Output Trees: As discussed, GA evolves a population of solutions toward the optimum. Generally, the more generations, the closer the GA solutions will be to the optimum. The termination condition in Figure 3.4 could be based on the total number of iterations, the maximum computational time, a threshold of desired video distortion, or a combination of these conditions.

Upon termination, the best individual (i.e., the one having the highest fitness value f^*) in the final population is taken as the solution to Problem OPT-MM. Note that in the final population, there may be more than one solution having this best-found fitness value f^*. It is also highly likely that there are other solutions having fitness values close to f^*. These solutions can be kept as back-ups for the multicast session, and can be used when the quality of the selected pair of trees deteriorates, thus reducing the need for executing the routing process for every tree interruption [51].

3.4 Performance Evaluation

In this section, we compare the performance of the GA-based multicast scheme with two representative network-centric algorithms, which are based on Dijkstra's shortest path routing algorithm. The first algorithm, the *Bounded Shortest Multicast Algorithm* (BSMA) algo-

rithm presented in [48], is designed to construct minimum-cost multicast trees with delay constraints and has been shown to achieve the lowest cost among several existing algorithms [52]. We extend BSMA to compute two trees, by running the algorithm twice, using a link cost metric $-\log(1-p_{ij})$ for the first run and a link cost metric $1/c_{ij}$ for the second run, in order to find a tree with the optimal loss characteristics and another tree with the largest bandwidths. The same decoding deadline Δ was used as a delay bound in BSMA.

In another work [51], Sajama and Haas presented the *Independent-Tree Ad Hoc Multicast Routing* (ITAMAR) procedure for efficient multicast routing in ad hoc networks. ITAMAR continuously maintains a set of multicast trees: one or more for the session to use and the rest as backups, so that the time between a tree failure and rerouting is minimized. Among the several algorithms proposed in [51], we implement the Shortest Path Heuristic (SPTH) algorithm for comparison, using the link cost metric $-\log(1-p_{ij})$. After a pair of trees are computed by these two algorithms, we apply the techniques in Section 3.2.2 to find the optimal rates for the descriptions and we compute the achieved distortion using (3.1).

3.4.1 Performance Comparison

For each experiment, we generate a wireless ad hoc network by placing a number of nodes at random locations in a square region. Connectivity is determined by the distance coverage of each node's transmitter. The source node s and the receivers $r \in \mathcal{M}$ are randomly chosen. For every link, the failure probability is randomly chosen from $[0, 1]$ with a truncated exponential distribution (having a mean of 0.01); and the available link bandwidth is randomly chosen from $[100\ Kb/s, 800\ Kb/s]$ according a uniform distribution, evenly spaced at 100 Kb/s intervals. The fixed delay τ_{ij} and mean delay t_{ij} of a link are set to 5 ms and 30 ms, respectively; the jitter δ_{ij} is randomly chosen from $[7ms, 17ms]$ according to a uniform distribution, $\forall\{i,j\} \in \mathcal{L}$. We set σ^2 to 1, since it only affects the absolute value of distortion, but does not affect path selection decisions.

The achieved average distortions by the various schemes are listed in Table 3.2, each being the average of 10 runs, for different network sizes and multicast groups. We find that the

Table 3.2: Average distortion achieved by GA and the two existing approaches

Ad Hoc Network	15-node (Δ=100 ms)		30-node (Δ=150 ms)		50-node (Δ=250 ms)	
Multicast Group	Dense (12)	Sparse (3)	Dense (25)	Sparse (5)	Dense (40)	Sparse (10)
BSMA	0.664	0.704	0.892	0.780	0.708	0.788
ITAMAR-SPTH	0.655	0.637	0.721	0.609	0.702	0.824
GA-based Routing	0.412	0.360	0.515	0.485	0.533	0.528

GA-based approach significantly outperformed the two network centric approaches in all of the cases studied. This is mainly due to the fact that both BSMA and ITAMAR-SPTH only optimize the network layer performance metrics, which does not necessarily achieve an optimal application layer performance. An interesting observation from Table 3.2 is that the improvement achieved by the GA-based multicast routing over other schemes is higher for sparse multicast sessions. This is because for sparse sessions, fewer receivers are involved and there is a greater freedom for the GA-based routing to select links, while the two network-centric algorithms do not explore this freedom very well since they are constrained by their shortest path routing objective.

3.4.2 GA versus ITAMAR-SPTH

Among the three schemes listed in Table 3.2, ITAMAR-SPTH has a performance closest to the GA-based routing in terms of distortion values. Therefore, we run ITAMAR-SPTH and the GA-based routing for a five-member group in a 15-node network, and compare the PSNRs of the reconstructed video frames, in order to demonstrate the efficiency of the proposed scheme. We use an H.263+ codec (originally from the University of British Columbia (UBC)) and the 400-frame "Foreman" trace in the QCIF format. We employ the same time-domain partitioning coding scheme as we do in Chapter 2. The video sequence is encoded with a frame rate of 30 fps and an intra MB refresh rate of 1/10. When necessary, the SNR scalable coding is used to code each description into two layers. We implement the off-line rate control for the enhancement layer, which is not available in the original UBC distribution. Each group of blocks (GOB) is transmitted in a packet to make them independently decodable.

The qualities of the trees found by both schemes are presented in Table 3.3. In general, GA is comparable to ITAMAR-SPTH in terms of the success delivery ratio. This is due to the fact that ITAMAR-SPTH uses link loss rates as the routing metric when determining the trees. However, ITAMAR-SPTH does not consider bandwidths and delays in the algorithmic design, making the delay and the bandwidth of the resulting trees unpredictable. For example, a receiver may have an extremely high delay such that almost all of the video packets are overdue (e.g., Receiver 3, Description 2). On the other hand, the GA-based routing optimizes video distortion directly, which is a compound function of the link statistics. Consequently, the GA-based approach achieves much higher video rates than does ITAMAR-SPTH. Such higher video rates greatly reduce the distortion caused by the lossy video coder. On average, the GA-based multicast routing achieves a 3.72 dB improvement in PSNR over the ITAMAR-SPTH algorithm in this experiment.

Table 3.3: GA-based routing versus ITAMAR-SPTH

	ITAMAR-SPTH (Δ=100 ms)					GA (Δ=100 ms)				
Receiver	1	2	3	4	5	1	2	3	4	5
Desc. 1 succ. ratio	95.6%	47.1%	86.2%	84.2%	99.5%	98.8%	56.6%	98.3%	98.9%	99.6%
Desc. 2 succ. ratio	99.3%	98.5%	0.2%	98.6%	99.0%	98.1%	99.6%	99.3%	97.9%	98.8%
Desc. 1 BL rate (Kb/s)	100	100	100	100	100	200	200	200	200	200
Desc. 1 EL rate (Kb/s)	0	0	0	0	300	0	300	0	300	300
Desc. 2 BL rate (Kb/s)	100	100	100	100	100	100	100	100	100	100
Desc. 2 EL rate (Kb/s)	0	0	0	0	0	300	0	300	300	300
Mean PSNR (dB)	29.53	24.16	24.38	27.55	30.95	31.64	28.63	31.43	31.75	31.70

We also plot the PSNR curves for three representative receivers in Figures 3.5, 3.6, and 3.7, respectively. The GA PSNR curves are well above the ITAMAR-SPTH curves for most of the frames. It can be seen that the GA-based routing attempts to achieve a balanced quality for the two descriptions, yielding a better subjective video quality. This is due to the symmetry of the description rates and loss probabilities in the objective function (see (3.1)). Minimizing such an objective function will drive GA to find balanced trees. It is possible to further reduce the quality difference between the two descriptions by using advanced MD coders. In order to illustrate the decoded video quality, we present Frame 226 obtained by

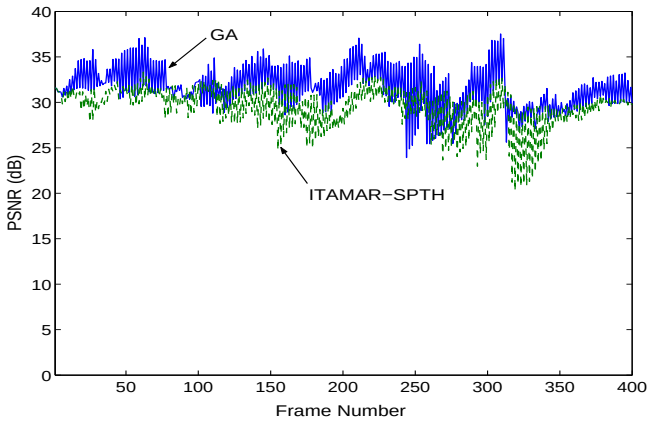


Figure 3.5: PSNRs of received frames by Receiver 1.

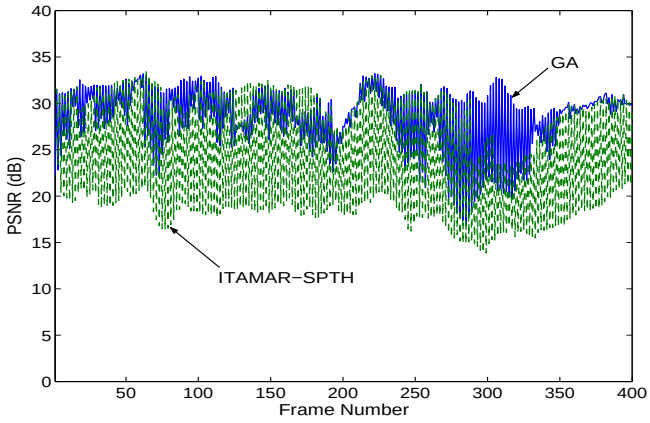


Figure 3.6: PSNRs of received frames by Receiver 2.

receivers 1, 2, and 3 in Figure 3.8. For all the receivers, the perceived quality obtained by the GA-based routing is much better than those obtained by ITAMAR-SPTH. Specifically, the pictures obtained by ITAMAR-SPTH for receivers 2 and 3 are barely recognizable.

One advantage of the network-centric algorithms, such as BSMA and ITAMAR, is that they have lower computational complexity than GA-based approaches. However, the efficiency of the GA algorithm can be improved since it is well suited for parallel computation. In addition, our numerical results show that with GA, the greatest improvement in fitness value is achieved after a few initial number of generations, and subsequent improvements are much smaller after these early generations. Therefore, for a delay-sensitive real-time application, GA can compute a set of "good" trees for the application to use after a very

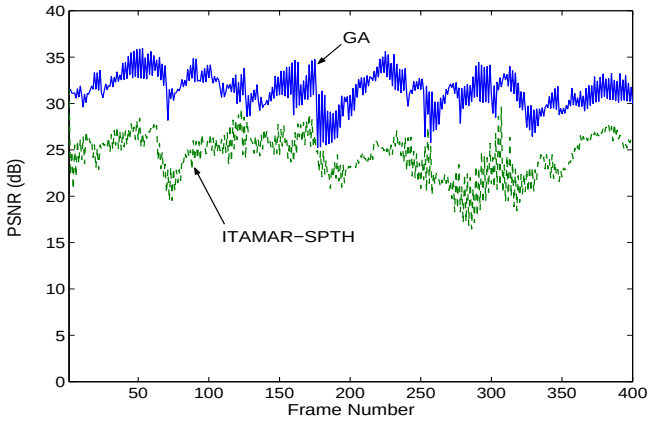


Figure 3.7: PSNRs of received frames by Receiver 3.

(a) Receiver 1, GA-based routing.

(b) Receiver 2, GA-based routing.

(c) Receiver 3, GA-based routing.

(d) Receiver 1, ITAMAR-SPTH.

(e) Receiver 2, ITAMAR-SPTH.

(f) Receiver 3, ITAMAR-SPTH.

Figure 3.8: Reconstructed Frame 226 at the receivers.

small delay. As GA continues to evolve, the trees can be dynamically updated with newly computed (better) trees for enhanced performance.

3.5 Related Work

The most relevant work to this research is ITAMAR by Sajama and Haas [51], which has been discussed in detail in Section 3.4. In this section, we discuss some other related work that contribute to further background knowledge of this investigation.

Multicasting MD video was first discussed in CoopNet [45] in the context of *application-level multicast* as a means to prevent web servers from being overwhelmed by a large "flash crowd." In CoopNet, clients form one or more distribution trees rooted at the server for live media streaming. Video is coded into multiple descriptions, each sent on a different tree, in order to reduce the disruption caused by node departures. CoopNet is quite effective for large-scale media multicasting, since it complements the client-server architecture (thus achieving the efficiency of centralized schemes) and exploits the unique strength in scalability of peer-to-peer networks. However, the CoopNet approach is not suitable for MD video multicast in ad hoc networks for the following reasons. First, the main design objective of CoopNet is to make servers robust to "flash crowds," with video quality as a secondary consideration. As a result, routing is performed by packing the clients into short and largely balanced trees, in which each tree edge is actually a (possibly large) number of network links. Such a logical link level routing approach cannot be easily translated into a physical-level link routing, which is the primary interest in this research. Second, CoopNet routing is not optimized in terms of video quality. As a result, such an approach cannot be optimal in ad hoc networks. Finally, the MD coding scheme used in CoopNet generates balanced descriptions. It does not consider the possibility that different trees may have different bandwidths. Since link qualities in an ad hoc network are highly diverse, such an approach may make the overall performance dependent on the quality of the worst tree. Moreover, in CoopNet, each description is not scalable (i.e., with a fixed rate). Thus, the performance of a tree is dependent on the quality of its worst link.

MD video streaming has been an active research area due to MD video's unique error resilience and open-loop operation capabilities [6, 8, 14, 17, 36, 38]. An empirical MD rate-distortion model has been presented in [8] for computing average video distortions from loss probabilities of path links. The scheme in [14] also shows how to compute the average video distortion from link statistics in the context of overlay networks for unicast MD video streaming. These models can be easily incorporated into the framework presented in this chapter. Some other works focus on end-system based schemes for supporting MD video unicast streaming for a set of given paths [17, 18, 38]. The important problem of finding multiple paths is not addressed.

QoS multicast routing has been an active research area for many years. Most of these problems belong to the class of minimum or constrained minimum Steiner tree problems [31], which are well-known to be NP-complete. Various efficient heuristic algorithms have been proposed (e.g., BSMA [48]). We refer readers to [52] for a comparison study and [40] for a survey of multicast routing protocols in ad hoc networks. Most of the proposed algorithms aim to find a single tree using network layer performance metrics. As our numerical results in the previous section show, such network-centric approaches do not necessarily deliver good performance at the application layer.

In [63], Zhang and Leung propose an orthogonal genetic algorithm for multimedia multicast routing, which is essentially a delay constrained Steiner tree problem. An interesting experimental design method, called orthogonal design, is incorporated into the crossover operation and is shown to greatly improve the convergence speed of the GA. In our earlier work [36], a GA-based multicast routing scheme for unicast MD video streaming is presented. In this chapter, we study a much more difficult problem of finding a pair of trees, while optimizing the application layer performance (i.e., MD video distortion).

3.6 Summary

In this chapter, we proposed a multicast scheme for MD video over ad hoc networks, within which multiple source trees are used. Furthermore, each video description is coded into

multiple layers in order to cope with diversity in wireless link bandwidths. We followed an application-centric, cross-layer approach and formulated the MD multicast routing into a combinatorial optimization problem. Due to the complex nature of problem structure, we pursued to develop efficient metaheuristic algorithm instead of exact analytic solution. We found that Genetic Algorithms are highly suitable for such problems and consequently developed a GA-based solution procedure for the MD multicast routing problem. Extensive simulations demonstrate significant gains in video quality achieved over existing approaches for a wide range of network operational conditions.

Chapter 4

Joint Routing and Server Selection for MD Video Streaming

4.1 Introduction

With the rapid advances in wireless ad hoc networking, it is expected that techniques such as caching and server replication have great potential in providing scalable multimedia services in wireless ad hoc networks [47], due to their great success for content delivery in the Internet [20].

Recent advances in Multiple Description (MD) coding have made it highly suitable for providing multimedia communications in wireless ad hoc networks [5, 27, 36, 38, 39], especially for distributed media delivery. MD for distributed storage has been suggested in [27], where a typical user would have fast access to the local video descriptions. For higher quality, one or more remote descriptions could be retrieved and combined with the local ones. In wireless ad hoc networks, links are much more diverse in terms of quality (e.g., available bandwidth and loss) than links in wireline networks: any link in an ad hoc network could be highly fragile with dynamic state conditions. In such an environment, pure server selection-based algorithms, although effective in the Internet, may produce low video quality if the default route links happen to have low available bandwidth or high loss rates.

In this chapter, we study the important problem of joint routing and server selection for MD video streaming in wireless ad hoc networks. In addition to selecting a pair of servers, we also explore optimal routing strategies to find good paths to the servers. Such a joint routing and server selection scheme opens a new dimension of freedom for further improving the MD video quality, since it explores a much larger solution space than existing server selection schemes.

Specifically, we first take an application-centric, cross-layer approach to formulate the joint routing and server selection task as a combinatorial optimization problem that minimizes the received video distortion. This problem is highly complicated, so the exact solutions are hard to obtain. Instead, we present schemes to compute an upper bound and a lower bound on the best achievable video distortion based on the monotonicity properties of the objective function. In addition, the upper bound produces a near-optimal pair of servers and a pair of corresponding paths for the client. The proposed approach is computationally efficient and can be easily incorporated into existing routing protocols for ad hoc networks. Our extensive numerical results show that the upper and lower bounds are very close to each other for all the cases studied, indicating that they are very close to the global optimum. We also observe significant gains in video quality achieved by the proposed approach over existing server selection schemes, which justify the importance of jointly considering routing and server selection for optimal MD video streaming in wireless ad hoc networks.

The remainder of this chapter is organized as follows. In Section 4.2, we formulate the joint routing and server selection task as a combinatorial optimization problem. We then examine the properties of the objective function and present algorithms for computing an upper bound and a lower bound for the achievable optimal distortion in Section 4.3. Our extensive experimental studies are presented in Section 4.4. We discuss practical issues in Section 4.5 and related work in Section 4.6. Section 4.7 concludes the chapter.

4.2 Problem Statement

In this section, we formulate the problem of joint routing and server selection for MD video streaming in wireless ad hoc networks. The notation used in the following presentation is given in Table 4.1.

Following the similar network definitions as we did in the previous chapters, we model a wireless mobile ad hoc network as a probabilistic directed graph $\mathcal{G}\{V, E\}$, where V is the set of vertices and E the set of edges. We assume that nodes are reliable during the video session, but a link may fail with certain probabilities. Similarly, we focus on network layer characteristics in this chapter, assuming that physical and MAC layer dynamics of wireless links are translated into network layer parameters.

4.2.1 Rate Distortion Model of MD Coding

Similar to the problems of multipath routing and multicast in previous chapters, we use the double-description (DD) coding to illustrate the problem formulation, since it is most widely used in MD video streaming in practice [4, 5, 12, 36, 38, 39]. For two descriptions, each generated for a sequence of video frames, let d_h denote the achieved distortion when only description h is received, $h = 1, 2$, and d_0 as the distortion when both descriptions are received. Also, let P_{00} be the probability of receiving both descriptions, P_{01} the probability of receiving Description 1 only, P_{10} the probability of receiving Description 2 only, and P_{11} the probability of losing both descriptions. Then, the average distortion of a received DD video can be expressed as [4]:

$$D = P_{00}d_0 + P_{01}d_1 + P_{10}d_2 + P_{11}\sigma^2, \tag{4.1}$$

where σ^2 is the variance of the source.

Let R_h be the rate in bits per sample of description h, $h = 1, 2$. We use the same rate-

Table 4.1: Notation

$\mathcal{G}\{V, E\}$:	graph representation of the network.
V :	set of vertices.
E :	set of edges.
$\mathcal{S}_h$:	server set that hosts description h, $h = 1, 2$.
s_h :	a chosen server, $h = 1, 2$.
u :	a client node.
$\mathcal{P}_h$:	a path from s_h to u, $h = 1, 2$.
$\{i, j\}$:	a link from node i to node j.
b_{ij} :	available bandwidth of link $\{i, j\}$.
p_{ij} :	success probability of link $\{i, j\}$.
l_{ij} :	average loss burst length of link $\{i, j\}$.
R_h :	rate of description h in bits per sample,$h = 1, 2$.
d_0 :	distortion when both descriptions are received.
d_h :	distortion when only description h is received, $h = 1, 2$.
D :	average video distortion.
T_{on} :	average "up" period of the joint links.
P_{00} :	probability that both descriptions are received.
P_{01} :	probability of receiving description 1 only.
P_{10} :	probability of receiving description 2 only.
P_{11} :	probability that both descriptions are lost.
x_{ij}^h :	routing index variables, defined in (4.3).
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 $\mathcal{P}_h$.
x_u^* :	constructed upper bounding solution.
x_l^* :	constructed lower bounding solution.

distortion region as we did in the previous chapters [4]:

$$\begin{cases} d_0 = \frac{2^{-2(R_1+R_2)}}{2^{-2R_1}+2^{-2R_2}-2^{-2(R_1+R_2)}} \cdot \sigma^2 \\ d_1 = 2^{-2R_1} \cdot \sigma^2 \\ d_2 = 2^{-2R_2} \cdot \sigma^2. \end{cases} \tag{4.2}$$

4.2.2 Computing Distortion for Two Given Paths

Within the network, let there be two sets of server nodes, denoted as $\mathcal{S}_h$, each hosting description h of a video in their cache or public directory, $h = 1, 2$. Note that these two sets do not have to be disjoint. If $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$, then nodes in $\mathcal{S}_1 \cap \mathcal{S}_2$ can offer both descriptions of the video clip. For video streaming, usually the server nodes do not perform on-line coding. Therefore, we assume that the descriptions have fixed and balanced rates, i.e., $R_1 = R_2 = R$.

Before we mathematically formulate the problem of joint routing and server selection, we need to compute the average distortion D as a function of link statistics for a *given* pair of servers and paths. We first define the following indices for describing the choice of a pair of paths:

$$x_{ij}^h = \begin{cases} 1, & \text{if } \{i,j\} \in \mathcal{P}_h, \ \ \forall \{i,j\} \in E, h = 1,2 \\ 0, & \text{otherwise}, \ \ \forall \{i,j\} \in E, h = 1,2. \end{cases} \tag{4.3}$$

With these index variables, an arbitrary path $\mathcal{P}_h$ can be represented by a vector $\mathbf{x}^h$ of $|E|$ elements, each of which corresponds to a link and has a binary value. Then, the bandwidth constraints of the links can be expressed as:

$$x_{ij}^1 \cdot R_1 + x_{ij}^2 \cdot R_2 \leq \rho \cdot b_{ij}, \ \ \forall \{i,j\} \in E, \tag{4.4}$$

where $\rho = \gamma \cdot W \cdot H \cdot R_f$ for a video with frame rate R_f and frame size $W \times H$; γ is a constant determined by the chroma subsampling format (e.g., $\gamma = 1.5$ for the quarter common intermediate format (QCIF) videos).

We now consider how to compute the end-to-end success probabilities. Since we do not mandate "disjointedness" in routing, $\mathcal{P}_1$ and $\mathcal{P}_2$ may share nodes and links. We classify the links along the two paths into three sets: set one consisting of joint 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 respectively as $\bar{\mathcal{J}}(\mathcal{P}_i)$, $i = 1, 2$. For disjoint portions of the paths, it suffices to model the packet losses as Bernoulli events, since the losses of the two descriptions are independent. Therefore, the success probabilities on the disjoint portions are:

$$p_{dj}^h = \begin{cases} \prod_{\{i,j\}\in\bar{\mathcal{J}}(\mathcal{P}_h)} p_{ij}, & \text{if } \bar{\mathcal{J}}(\mathcal{P}_h) \neq \emptyset,\ h = 1, 2 \\ 1, & \text{otherwise},\ h = 1, 2. \end{cases} \tag{4.5}$$

On the joint portion of the paths, the losses of the two streams are correlated. In order to capture such correlation, similar to the approach in Chapter 2, we model each link $\{i, j\}$ as an on-off process modulated by a discrete-time Markov chain, as shown in Figure 4.1(a). There is no packet loss when the link is in the "up" state, and the packet loss rate is 1 when the link is in the "down" state. The transition probabilities, $\{\alpha_{ij}, \beta_{ij}\}$, can be computed from the measured link statistics, as $\beta_{ij} = 1/l_{ij}$ and $\alpha_{ij} = (1 - p_{ij})/(p_{ij}l_{ij})$. 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 a similar approach in [5] and [12] to model the aggregate process as an on-off process. Observe that a packet is successfully delivered on the joint portion if and only if all joint links are in the "up" state. Therefore, we can lump all the states having at least one link failure into a single "down" state, while using the remaining state where all the links are in the good condition as the "up" state. Letting T_{on} be the average length of the "up' period, we have,

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

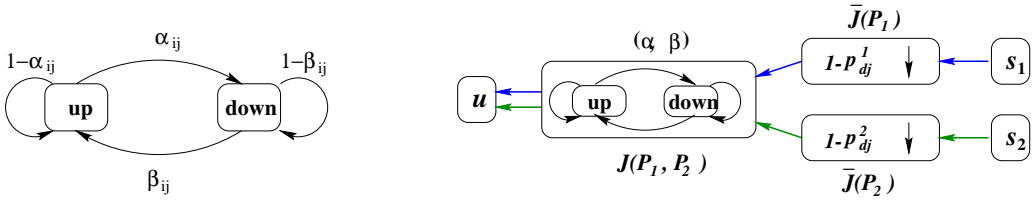
Furthermore, the average success probability of the joint portion is:

$$p_{jnt} = \begin{cases} \prod_{\{i,j\}\in\mathcal{J}(\mathcal{P}_1,\mathcal{P}_2)} p_{ij}, & \text{if } \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) \neq \emptyset \\ 1, & \text{otherwise.} \end{cases} \tag{4.6}$$

Finally, the transition probabilities of the aggregate on-off process are:

$$\begin{cases} \alpha = \frac{1}{T_{on}} \\ \beta = \frac{p_{jnt}}{T_{on}(1-p_{jnt})}. \end{cases}$$

Note that $\alpha = 0$ and $\beta = 0$ if $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) = \emptyset$. The consolidated path model is illustrated in Figure 4.1(b), where $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$ is modeled as a two-state Markov process with parameters $\{\alpha, \beta\}$, and $\bar{\mathcal{J}}(\mathcal{P}_h)$ is modeled as a Bernoulli process with parameter p_{dj}^h, $h = 1, 2$.



(a) The two-state Markov link model.

(b) A consolidated path model for double-description video.

Figure 4.1: Link and path models.

With the above path model, the joint probabilities of receiving the descriptions can be computed as:

$$\begin{cases} P_{00} = p_{jnt} \cdot (1-\alpha) \cdot p_{dj}^1 \cdot p_{dj}^2 \\ P_{01} = p_{jnt} \cdot p_{dj}^1 \cdot [1-(1-\alpha) \cdot p_{dj}^2] \\ P_{10} = p_{jnt} \cdot p_{dj}^2 \cdot [1-(1-\alpha) \cdot p_{dj}^1] \\ P_{11} = 1 - p_{jnt} \cdot [p_{dj}^1 + p_{dj}^2 - (1-\alpha) \cdot p_{dj}^1 \cdot p_{dj}^2]. \end{cases} \tag{4.7}$$

Let $a = 2^{-2R_1}$ and $b = 2^{-2R_2}$. For balanced descriptions, we have that $a = b$. The average video distortion can be computed by substituting (4.2) and (4.7) into (4.1):

$$\begin{aligned} \frac{D}{\sigma^2} \;=\; & 1 + p_{jnt} \cdot \Big[(a-1) \cdot p_{dj}^1 + (b-1) \cdot p_{dj}^2 + \\ & (1-\alpha)\frac{(a+b)(a-1)(b-1)}{a+b(1-a)} \cdot p_{dj}^1 \cdot p_{dj}^2 \Big]. \end{aligned} \tag{4.8}$$

4.2.3 The Optimal Routing Problem

With the above preliminaries, we can mathematically formulate the joint routing and server selection problem for MD video (OPT-JRSS) as follows:

OPT-JRSS For two given server sets $\{\mathcal{S}_1, \mathcal{S}_2\}$ and a given client u, find an optimal solution

$x^* = \{s_1^*, s_2^*, \mathcal{P}_1^*, \mathcal{P}_2^*\}$ that minimizes the average distortion D defined in (4.1). That is,

$$\textbf{Minimize}: \ D \tag{4.9}$$

subject to:

$$\sum_{j:\{i,j\}\in E} x_{ij}^h = \begin{cases} \leq 1, & \text{if } i \neq u \\ = 0, & \text{if } i = u \end{cases}, \forall i \in V, h = 1,2 \tag{4.10}$$

$$\sum_{j:\{i,j\}\in E} x_{ij}^h - \sum_{j:\{j,i\}\in E} x_{ji}^h = \begin{cases} 1, & \text{if } i = s_h \\ -1, & \text{if } i = u \\ 0, & \text{otherwise} \end{cases}, \ \forall i \in V, h = 1,2 \tag{4.11}$$

$$x_{ij}^1 \cdot R_1 + x_{ij}^2 \cdot R_2 \leq \rho \cdot b_{ij}, \ \ \forall \{i,j\} \in E \tag{4.12}$$

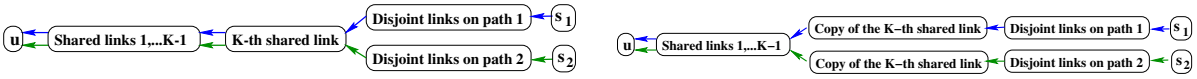
$$x_{ij}^h \in \{0,1\}, \ \ \forall \{i,j\} \in E, h = 1,2 \tag{4.13}$$

$$s_h \in \mathcal{S}_h, \ \ h = 1,2. \tag{4.14}$$

In Problem OPT-JRSS, $\{x_{ij}^h\}_{\{i,j\}\in E, h=1,2}$ and $\{s_h\}_{h=1,2}$ are optimization variables, representing the choice of a pair of servers and the links on a pair of paths from the chosen servers to the client. Constraints (4.10) and (4.11) guarantee that the paths are loop-free, while constraint (4.12) guarantees that the links are stable. 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 (4.1), (4.2), and (4.7). If multiple solutions are found having the minimum distortion, we can break ties by choosing the solution that has the largest bandwidth along the two chosen paths (see (4.12)).

4.3 Lower and Upper Distortion Bounds

In the following, we first introduce several monotonicity properties of the objective function (4.9). Then, we construct a lower bound and an upper bound on the achievable video distortion.



(a) Solution $\hat{x}$. (b) Solution $\bar{x}$.

Figure 4.2: The two solutions have the same set of links. The only difference between them is that a link is shared in $\hat{x}$ (the K-th shared link), but not shared in $\bar{x}$ (appended to each of the disjoint portions).

4.3.1 Properties of the Objective Function

The objective function of Problem OPT-JRSS, (4.9), has the following *monotonicity* properties (which are similar to the monotonicity properties in Chapter 2).

Property 1. D is non-increasing with R_h, $h = 1, 2$.

Proof. Recall that $a = 2^{-2R_1} \leq 1$ and $b = 2^{-2R_2} \leq 1$. From (4.1) and (4.2), we have

$$\frac{1}{\sigma^2}\frac{\partial D}{\partial R_1} = -P_{00}\frac{2\ln 2 \cdot ab^2}{(a+b-ab)^2} - 2\ln 2 \cdot P_{01}a \leq 0$$

Similarly, we have $\frac{\partial D}{\partial R_2} \leq 0$ due to the symmetry in (4.8). □

Property 2. For two completely disjoint paths, D is non-increasing with p_{dj}^h, $h = 1, 2$.

Proof. For a disjoint path set $\{\mathcal{P}_1, \mathcal{P}_2\}$, we have that $p_{jnt} = 1$ and $\alpha = 0$. Then, we have

$$\begin{aligned}\frac{1}{\sigma^2}\frac{\partial D}{\partial p_{dj}^1} &= (a-1)\left[\frac{(1-p_{dj}^2)(a+b(1-a)) + b^2 p_{dj}^2}{a+b(1-a)}\right] \\ &\leq 0.\end{aligned}$$

Similarly, we have $\frac{\partial D}{\partial p_{dj}^2} \leq 0$ due to the symmetry in (4.8). □

Property 3. Consider the two solutions $\hat{x}$ and $\bar{x}$ shown in Figure 4.2. If the the on-off failure process of the K-th shared link is random or bursty, i.e., $\alpha_{ij} + \beta_{ij} \leq 1$, then $D(\hat{x}) \geq D(\bar{x})$.

1. Remove link(s) $\{i,j\}$ having $\rho \cdot b_{ij} < R$, $\forall\{i,j\} \in E$ to obtain a reduced graph $G(V,E')$;
2. Set the cost of link $\{i,j\}$ to $\log(1/p_{ij})$, $\forall\{i,j\} \in E'$;
3. Find the path $\mathcal{P}_h^l$ from a server $s_h^l \in \mathcal{S}_h$ to u in $G(V,E')$ that has the minimum cost among all paths to all $s_h \in \mathcal{S}_h$, $h = 1,2$;
4. Assuming $\mathcal{P}_1^l \cap \mathcal{P}_2^l = \emptyset$, compute $D(x_l^*)$, where $x_l^* = \{s_1^l, s_2^l, \mathcal{P}_1^l, \mathcal{P}_2^l\}$.

Figure 4.3: ALG-LB: Construct a lower bounding solution x_l^*.

Proof. For solution $\hat{x} = \{s_1, s_2, \mathcal{P}_1, \mathcal{P}_2\}$ in Figure 4.2(a), let there be K joint links with parameters $\{\alpha_k, \beta_k\}$, $k = 1, \cdots, K$. We have:

$$\frac{1}{\sigma^2}\left[D(\hat{x}) - D(\bar{x})\right] = p_{jnt} \cdot p_{dj}^1 \cdot p_{dj}^2 \cdot \prod_{k=1}^{K-1}(1-\alpha_k) \cdot (1-\alpha_K-\beta_K)(1-p_K)\frac{(a+b)(1-a)(1-b)}{a+b(1-a)} \geq 0,$$

according to the "bursty" assumption. □

4.3.2 A Distortion Lower Bound

We are now ready to construct a tight lower bound on the average video distortion. From the monotonicity properties of D, we need to find a path pair having the best loss characteristics.

Algorithm ALG-LB in Figure 4.3 can be used to construct a solution x_l^* that yields a lower bound for D. In ALG-LB, we set the cost of a link $\{i,j\}$ to $\log(1/p_{ij})$, $\forall\{i,j\} \in E$. Then, the total cost of a path $\mathcal{P}$ is:

$$\sum_{\{i,j\}\in\mathcal{P}} \log\left(\frac{1}{p_{ij}}\right) = \log\left(\frac{1}{\prod_{\{i,j\}\in\mathcal{P}} p_{ij}}\right).$$

Applying a shortest path routing algorithm, we can find a path having the minimum cost, so that then, $\prod_{\{i,j\}\in\mathcal{P}} p_{ij}$ is maximized. According to (4.8) and Property 2, if the paths $\{\mathcal{P}_1^l, \mathcal{P}_2^l\}$ found in ALG-LB are disjoint, then they are the optimal solution to Problem OPT-JRSS; otherwise, the computed distortion assuming that $\{\mathcal{P}_1^l, \mathcal{P}_2^l\}$ are disjoint will be a lower bound of the distortion achieved by the optimal solution (according to Property 3). For the constructed solution x_l^*, we have the following proposition holding true.

Proposition 2. The distortion, $D(x_l^*)$, of x_l^* constructed by ALG-LB is a lower bound for the average distortion D defined in (4.8).

Proof. The formation of the optimal solution $x^* = \{s_1^*, s_2^*, \mathcal{P}_1^*, \mathcal{P}_2^*\}$ could conform with one of the following two cases:

Case I: If x^* is comprised of a pair of disjoint paths, then from the construction procedure, we have that $p_{dj}^1(\mathcal{P}_1^l, \mathcal{P}_2^l) \geq p_{dj}^1(\mathcal{P}_1^*, \mathcal{P}_2^*)$ and $p_{dj}^2(\mathcal{P}_1^l, \mathcal{P}_2^l) \geq p_{dj}^2(\mathcal{P}_1^*, \mathcal{P}_2^*)$, where $p_{dj}^h(\mathcal{P}_1, \mathcal{P}_2) = \prod_{\{i,j\} \in \bar{\mathcal{J}}(\mathcal{P}_h)} p_{ij}$ for disjoint paths $\{\mathcal{P}_1, \mathcal{P}_2\}$, $h = 1, 2$ (see (4.5)). From Property 2, we have that $D(x_l^*) \leq D(x^*)$.

Case II: If $\mathcal{P}_1^*$ and $\mathcal{P}_2^*$ share K links, we can construct a virtual solution $\bar{x}^* = [\bar{\mathcal{P}}_1^*, \bar{\mathcal{P}}_2^*]$, by (i) appending a copy of the shared link k to the disjoint portions of the two paths; (ii) removing the shared link k from the shared portion, $k = 1, \cdots, K$ (see Figure 4.2). That is, we construct a solution $\bar{x}^*$ with disjoint paths and identical links to x^* by duplicating each shared link in x^*. Note that as a result, $\bar{x}^*$ may not be realizable. By applying Property 3 repeatedly for K times, we have that $D(\bar{x}^*) \leq D(x^*)$. Finally, from Case I, we have that $D(x_l^*) \leq D(\bar{x}^*) \leq D(x^*)$. □

In ALG-LB, $\mathcal{P}_h^l$ can be found by applying Dijkstra's algorithm to first find the lowest cost paths to each server in $\mathcal{S}_h$, with a time complexity of $O(|\mathcal{S}_h| \cdot |E'| \cdot \log |V|)$, and then choose the server (and the corresponding path) having the minimum cost path among all servers in $\mathcal{S}_h$, with a time complexity of $O(|\mathcal{S}_h|)$, $h = 1, 2$. Note that although the computed $\mathcal{P}_1^l$ and $\mathcal{P}_2^l$ may share links, we assume that they are completely disjoint in order to obtain a distortion lower bound. As a result, the solution x_l^* that achieves the lower bound may not be realizable.

4.3.3 A Distortion Upper Bound

Although the above lower bound is very useful in providing a close approximation for the lowest achievable distortion by jointly selecting the optimal servers and the corresponding optimal paths to them, Algorithm ALG-LB does not provide a usable set of servers and

1. Remove link(s) $\{i,j\}$ having $\rho \cdot b_{ij} < R$, $\forall\{i,j\} \in E$ to obtain a reduced graph $G(V,E')$;
2. Set the cost of link $\{i,j\}$ to $\log(1/p_{ij})$, $\forall\{i,j\} \in E'$;
3. Find the path $\mathcal{P}_1^u$ from a server $s_1^u \in \mathcal{S}_1$ to u in $G(V,E')$ that has the minimum cost among all paths to all $s_1 \in \mathcal{S}_1$;
4. From $G(V,E')$, remove link(s) $\{i,j\}$ having $\rho \cdot b_{ij} < 2R$, $\forall\{i,j\} \in \mathcal{P}_1^u$ to obtain a further reduced graph $G(V,E'')$;
5. Find the path $\mathcal{P}_2^u$ from a server $s_2^u \in \mathcal{S}_2$ to u in $G(V,E'')$ that has the minimum cost among all paths to all $s_2 \in \mathcal{S}_2$;
6. Compute $D(x_u^*)$, where $x_u^* = \{s_1^u, s_2^u, \mathcal{P}_1^u, \mathcal{P}_2^u\}$.

Figure 4.4: ALG-UB: Construct an upper bounding solution x_u^*.

paths for client u. In this section, we present an algorithm to construct a feasible solution that yields an upper bound on D.

Algorithm ALG-UB in Figure 4.4 can be used to construct a solution x_u^* that achieves an upper bound for D. As in ALG-LB, we set the cost of a link $\{i,j\}$ to $\log(1/p_{ij})$, $\forall\{i,j\} \in E$. Thus the minimum cost path has the best end-to-end loss characteristics. In addition, we also take into consideration the link bandwidth constraints, by removing those links that do not have sufficient bandwidth to support both descriptions when computing the optimal path to the second server set, in order to make a feasible solution. As in ALG-LB, $\mathcal{P}_1^u$ can be found by applying Dijkstra's algorithm with a time complexity of $O(|\mathcal{S}_1| \cdot |E'| \cdot \log|V| + |S_1|)$ and $\mathcal{P}_2^u$ can be found with a time complexity of $O(|\mathcal{S}_2| \cdot |E''| \cdot \log|V| + |S_2|)$. For the constructed solution x_u^*, we have the following result holding true.

Proposition 3. The distortion of x_u^* constructed in ALG-UB, $D(x_u^*)$, is an upper bound for the average distortion D defined in (4.8).

Proof. Clearly, $x_u^* = \{s_1^u, s_2^u, \mathcal{P}_1^u, \mathcal{P}_2^u\}$ is a feasible solution to Problem OPT-JRSS, since it satisfies all the constraints (4.10)–(4.14). Therefore, $D(x_u^*)$ must be an upper bound for D, which is the distortion of the global optimal solution x^*. □

The four-tuple $\{s_1^u, s_2^u, \mathcal{P}_1^u, \mathcal{P}_2^u\}$ provides a usable solution to Problem OPT-JRSS. We will show that the lower and upper bounds are very close to each other. In other words, the upper bound is near-optimal in all of the cases that we examined.

4.4 Numerical Results

In this section, we examine the performance of the proposed distortion bounds via a set of experiments. In each experiment, we generated an ad hoc network topology by placing a number of nodes at random locations in a square region. Connectivity was determined by the distance coverage of each node's transmitter (set to 250 m in all the following experiments). The client node and server nodes were randomly chosen.[1] For all of the experiments reported, the success probability p_{ij} was randomly chosen from $[0.9, 0.995]$ according to a uniform distribution, $\forall\{i,j\} \in E$. We set the variance σ^2 to 1, since it does not influence routing and server selection decisions. Other parameter settings will be introduced in the following when the results are being discussed. The computation time was in tens of milliseconds for all the experiments performed.

4.4.1 Optimality of the Distortion Bounds

One important performance concern is the optimality of the proposed lower and upper distortion bounds. Due to the complex nature of Problem OPT-JRSS, a closed-form optimal solution is not obtainable. But the global optimal solution may be numerically obtained via an exhaustive search for small-sized networks.

The distortion bounds for two 15-node networks found by ALG-UB and ALG-LB are presented in Tables 4.2, as well as the global optimal distortion values found by an exhaustive search. We also varied the video description rate and mean burst length to examine their impact. In these experiments, the available bandwidth of each wireless link b_{ij} was randomly chosen from $[128Kbps, 448Kbps]$ according to a uniform distribution, in steps of $64Kbps$, $\forall\{i,j\} \in E$. We observe that for all the cases, the global optimal distortion (found by exhaustive search) always lies between the corresponding lower and upper bounds. In addition, the difference between the bounds is negligible. In Table 4.2, the largest difference between the lower and upper bounds is 0.0061, giving a relative difference of 1.3%.

[1] We avoided the trivial cases where the servers are within two hops from the client node.

We also performed extensive simulations for larger sized networks (i.e., 50-, 80-, and 100-node networks) where exhaustive search is impractical, and for different MD description rates R. The results are shown in Table 4.3. Again, the proposed bounds were very close to each other in all of the cases examined. In many cases, the lower and upper bounds yield the same distortion value, implying that they are actually the global optimal solutions. The maximum relative difference between the lower and upper bounds in Table 4.3 is 6.4% (the 80-node network with $R = 384Kbps$), indicating the near-global optimality of the derived upper bounding solutions.

Clearly, the proposed bounds can provide an excellent estimation for the global optimal solution. The servers and the corresponding paths found by ALG-UB yield a highly competitive solution to Problem OPT-JRSS. In addition, since ALG-UB is based on Dijkstra's algorithm, the computation time for each run was in tens of milliseconds using a Pentium-4 2.4 GHz computer (with 512 MB memory). The proposed algorithms are computationally efficient and are suitable for joint routing and server selection for large-sized ad hoc networks.

Table 4.2: Comparison of the proposed bounds for two 15-node networks

	$R = 128Kbps$		$R = 192Kbps$	
l_{ij}	[2,6]	[10,25]	[2,6]	[10,25]
Upper Bound	0.5928	0.5931	0.4752	0.4758
Exhaustive Search	0.5916	0.5917	0.4721	0.4722
Lower Bound	0.5897	0.5897	0.4697	0.4697
SP	0.7976	0.7104	0.7247	0.6136
Heuristic	0.7976	0.7104	0.7247	0.6136
Distortion	0.7883	0.6846	0.7247	0.5816

Table 4.3: Comparison of the upper and lower bounds for different networks: $l_{ij} \in [2, 6]$, $\forall\{i, j\} \in E$

R (Kbps)	64	128	192	256	320	384
UB(50n)	0.756	0.589	0.496	0.407	0.350	0.302
LB(50n)	0.756	0.589	0.485	0.396	0.341	0.289
UB(80n)	0.755	0.596	0.478	0.404	0.348	0.316
LB(80n)	0.755	0.587	0.467	0.402	0.342	0.297
UB(100n)	0.758	0.593	0.477	0.385	0.316	0.328
LB(100n)	0.758	0.592	0.473	0.385	0.316	0.309

4.4.2 Comparison with Existing Algorithms

To compare with existing server selection schemes, we implemented the following three server selection algorithms proposed in [5] for MD video streaming in CDN.

1. Shortest Path (SP): pick the closest server (in terms of hop count) from each server set.

2. Heuristic: compute a score, $r_{mn} = (L_m + L_n)/2 + L^J_{mn}$, for each pair of servers $\{s_m, s_n\}$ having complementary descriptions, where L_m (L_n) is the path length in hop-count (i.e., for the default or shortest path) from server s_m (s_n) to u, and L^J_{mn} is the number of joint links. Then, choose the server pair having the lowest score.

3. Distortion: calculate the expected distortion for each server pair having complementary descriptions. Then, choose the pair that yields the lowest distortion.

Note that although the Distortion algorithm performs a server selection based on the average video distortion, it does not take advantage of using an optimal routing for MD video. In other words, the distortion is computed for a server pair using two *default* paths (e.g., the shortest paths) from the two servers to the client.

The distortion value obtained by the three algorithms are also presented in Table 4.2. It can be observed that the Heuristic algorithm usually has a performance no worse than SP, while the Distortion algorithm has the best performance among the three. In Tables 4.2, we can see that ALG-UB outperforms all the three existing algorithms with a significant margin, implying that the latter three algorithms may not be suitable for wireless ad hoc networks, although they have been shown to be quite effective in the Internet. In wireless ad hoc networks, links have highly diverse qualities. Therefore, only considering hop-count in server selection would not produce good perceived MD video quality. For the Distortion algorithm, although it selects servers based on the computed distortion values, it does not necessarily provide good results since it only considers the default routes from the servers to the client. It may not be efficient in handling the cases when there are low quality links (e.g., low available bandwidth or high loss rates) in the default routes.

Since the Distortion algorithm has the best performance among the three existing algorithms, we further compared its performance with ALG-UB by transmitting MD video in a 50-node ad hoc network. There were 10 servers in each server set. We chose a similar time-domain partitioning coding scheme as we did in the previous two chapters.

For the experiment, the 400-frame QCIF [176×144 Y pixels/frame, 88×72 Cb/Cr pixels/frame] sequence "Foreman" was encoded at 15 fps and $192Kbps$ for each description. The descriptions were then packetized (one GOB per packet) and transmitted over the paths found by the algorithms.

The PSNRs of the reconstructed video frames are plotted in Figure 4.5. It can be observed that during the period of Frame 65 to 92 and the period of Frame 270 to 290, the ALG-UB curve suffers big drops. By examining the packet loss trace, we found that these valleys were caused by bursty, concurrent loss of packets from both descriptions. Furthermore, the PSNR curve obtained by ALG-UB is well above that obtained by the Distortion algorithm for most of the frames. We also plot the decoded Frame 229 obtained by the two algorithms in Figure 4.6 to illustrate the visual quality. It can be seen that the image delivered by ALG-UB has a much better quality than that delivered by the Distortion algorithm. The average PSNRs obtained by ALG-UB and Distortion are $29dB$ and $21.9dB$, respectively. By

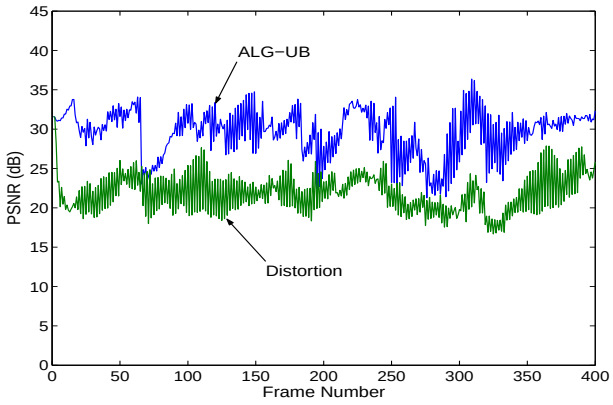


Figure 4.5: PSNRs of reconstructed frames obtained by Algorithm ALG-UB and the Distortion scheme.

jointly optimizing the routing and server selection decisions, an $8.1dB$ gain in average PSNR has been achieved, which demonstrates the efficacy of the joint routing and server selection approach for MD video in wireless ad hoc networks.

(a) Frame 229 (ALG-UB).

(b) Frame 229 (Distortion).

Figure 4.6: Reconstructed Frame 229 at the client node.

4.4.3 Increasing Video Rates

We examine the impact of the description rate R in this subsection. For a 100-node network having $|\mathcal{S}_1| = |\mathcal{S}_2| = 15$, we computed the average video distortion values using the algorithms, while increasing R from $64Kbps$ to $384Kbps$ in steps of $64Kbps$. The link bandwidth b_{ij} was randomly chosen from $[64Kbps, 576Kbps]$, also in steps of $64Kbps$.

The computed distortion values using the algorithms are plotted in Figure 4.7. When R was increased, there was less coding distortion caused by the lossy encoder. On the other hand,

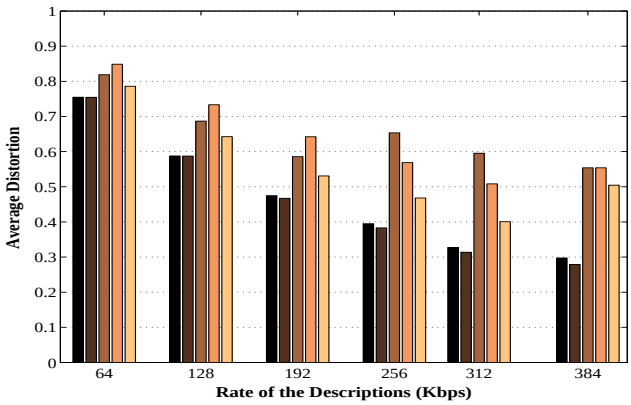


Figure 4.7: Average distortions for increasing description rate R. From left to right for each value of R: ALG-UB, ALG-LB, SP, Heuristic, Distortion.

a larger R makes more links having $b_{ij} < R$ unusable in the routing. Again, the upper and lower bounds are very close to each other, although it can be observed that the difference increases slightly when R gets large. In the worst case (when $R = 384Kbps$), the difference between the bounds is 0.0188, giving a 6.3% relative difference. The proposed algorithms also outperform the existing three schemes by a large margin for all the description rates examined.

4.5 Practical Implications

In practice, the joint routing and server selection scheme can also be incorporated into existing distributed routing protocols for wireless ad hoc networks. Existing routing protocols can be roughly categorized as *proactive*, where a consistent and up-to-date view of the network is always maintained, and *reactive*, where route discovery is performed on-demand. For proactive routing protocols (e.g., OLSR [19]), we can define a new type of Link State Advertisement (LSA), in addition to the original types that report link states and statistics, to represent the availability of video descriptions at each node. Then, a client node can determine the two server sets from the received LSAs and use Algorithm ALG-UB to quickly find near-optimal servers and paths to them.

Under reactive routing protocols (e.g., DSR [32]), we can let the client node broadcast Video

Request (VREQ) messages (rather than Route Request (RREQ) messages in the original DSR) to the network in order to discover nodes that host one or both of the video descriptions. Such a node, after receiving the VREQ message, will return a Video Reply (VREP) message (rather than Route Reply (RREP) in the original DSR) to the client, carrying information on which description(s) it has, link statistics, and path information. After receiving a number of such VREPs, the client can construct a partial view of the network and the server sets, and then run Algorithm ALG-UB to select the best servers along with associated routes to them.

4.6 Related Work

Caching and server replication are common techniques for providing scalable distributed service over the Internet. The single server selection problem, i.e., how to select a server from a set of mirror sites for a client request so as to provide the "best" service for the client, has been studied over the years (e.g., see [20] and the references therein). In existing server selection schemes, either the client or the servers monitor the server loads and/or network performance (e.g., round trip times (RTT) from the servers to the client) and then select a "best" server that has the lowest load or the lowest delay based on these measurements [20]. These schemes are mainly designed for data applications (e.g., web service) and do not explicitly attempt to optimize video quality. Moreover, the important optimal routing problem has not been addressed.

As discussed, Apostolopoulos *et al.* presented an interesting study of server selection for MD video in the context of CDN networks in [5]. It has been shown that server selection for MD video streaming provides an effective means of exploiting the path diversity provided by CDN. As a result, significant reduction in video distortion has been observed [5]. However, similar to existing approaches for single server selection, the proposed MD server selection algorithms only consider default routes on selecting optimal servers. As a result, the achieved optimal solution by the algorithms in [5] are for a much smaller solution space. Optimal routing, which can further improve MD video quality, has not been considered.

An empirical path distortion model was presented in [5] that computes received SD or MD video distortion from link loss characteristics. Recently, a similar path distortion model was introduced in [12] for MD video in overlay networks, which takes into account more link statistics, such as delay, jitter, and bandwidth, in addition to loss. In this work, we took a similar methodology as these papers, e.g., modeling the loss process on the joint portion of two paths as a Markov chain and modeling the losses on the disjoint portions as Bernoulli events.

4.7 Summary

In this chapter, we studied the important problem of jointly selecting servers and determining optimal routes for MD video streaming in wireless ad hoc networks. We took an application-centric, cross-layer approach to formulate the joint routing and server selection task as a combinatorial optimization problem that minimizes the received video distortion. We derived a lower bound and an upper bound for the best achievable video distortion. The upper bound was demonstrated to produce a near-optimal pair of servers along with a pair of corresponding paths. The proposed approach can be easily incorporated into existing routing protocols for ad hoc networks. Our extensive numerical results show that the bounds are very close to each other for all the cases studied, indicating the near-global optimality of the derived upper bounding solution. We also observed significant gains in video quality achieved by the proposed approach over existing server selection schemes. This justifies the importance of jointly considering routing and server selection for optimal MD video streaming in wireless ad hoc networks.

Chapter 5

Conclusions and Future Work

5.1 Summary of Thesis Research

In the previous chapters, we studied three important problems regarding MD video transport in wireless ad hoc networks and proposed corresponding solutions for these problems:

- Multipath routing for MD video over wireless ad hoc networks.
- MD video multicast in wireless ad hoc networks.
- Joint routing and server selection for MD video streaming in wireless ad hoc networks.

In Chapter 2, we studied the important problem of optimal multipath routing for MD video. We formulated the multipath routing problem from an application-centric, cross-layer perspective. A GA-based approach was found to be eminently suitable to address such optimal routing problems, which have complex objective functions and exponential solution spaces. With extensive simulation results, we found that this approach offers near-optimal solutions. We also developed a tight lower bound for video distortion, which can be used to evaluate the performance of a GA-based solution as well as to establish its termination criteria. Although a GA-based approach is a centralized algorithm in nature, we showed that it is amenable for distributed implementation in ad hoc routing protocols. This work provides an important

methodology for addressing complex cross-layer optimization problems, particularly those involving the application and network layers.

In Chapter 3, we proposed a multicast scheme for MD video over ad hoc networks, using multiple source trees. Furthermore, each video description is coded into multiple layers in order to cope with diversity in wireless link bandwidths. The resulting MD multicast routing problem was formulated into a combinatorial optimization problem. Due to the complex nature of problem structure, we developed efficient metaheuristic algorithm instead of exact analytic solution. We found that Genetic Algorithms are also highly efficient for such problems and consequently developed a GA-based solution procedure for this problem. Simulation studies showed significant gains in video quality achieved over existing approaches for a wide range of network operational conditions.

In Chapter 4, we presented the important problem of jointly selecting servers and determining optimal routes for MD video streaming in wireless ad hoc networks. We took the similar application-centric, cross-layer approach to formulate this problem as a combinatorial optimization problem that minimizes the received video distortion. We derived a lower bound and an upper bound for the best achievable video distortion. The upper bound was demonstrated to be very close to the lower bound, indicating that it produces a near-optimal pair of servers along with a pair of corresponding paths. We also observed significant gains in video quality achieved by the proposed approach over existing server selection schemes. This justifies the importance of jointly considering routing and server selection for optimal MD video streaming in wireless ad hoc networks. The proposed approach can also be easily incorporated into existing routing protocols for ad hoc networks.

5.2 Future Research Directions

To solve multipath routing for MD video and MD video multicast problems, we proposed GA based metaheuristic approaches. In most cases, they have demonstrated themselves to be quite efficient solutions to this type of highly complex combinatorial optimization problems. However, in our investigation we found sometimes they have a tendency of premature conver-

gence, which means the evolution of GA is too fast and the final solution is suboptimal. To combat the premature convergence problem and further improve GA's performance, we studied some enhancement schemes and found a niching method family, including *fitness sharing, crowding* and *clearing*, is efficient to address this problem [49, 53]. In particular, we found fitness sharing has the best performance among them. Since most of the anti-premature methods proposed in literature are for continuous real-number optimization problems, in our future research, we will focus our efforts on how to combat the premature convergence of GA in combinatorial optimization problems (the cases of the problems studied in this thesis).

To solve the combinatorial optimization problems discussed in the previous chapters, if we could obtain some kind of analytic solutions, it would be easier to analyze and improve the performance systematically. In recent research, we found a novel *Reformulation-Linearization Technique* (RLT) [55] [59] is promising to be used in our problems. RLT has a solid mathematical foundation and can solve the optimization problems analytically. It has been applied in many complex combinatorial optimization problems [56]. In our preliminary investigation, after careful approximation and relaxation we formulated the multipath routing for MD video problem into a quadratic programming problem. Then we may solve the resulting quadratic programming problem using RLT [59]. We will be continuing our research efforts on this track.

Bibliography

[1] The QoS with the OLSR protocol homepage. [online]. Available: http://qolsr.lri.fr/.

[2] E. Aarts and J. Korst. *Simulated Annealing and Boltzman Machines*. John Wiley & Sons, New York, NY, 1989.

[3] C.W. Ahn and R.S. Ramakrishna. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Trans. Evol. Comput.*, 6(6):566–579, Dec. 2002.

[4] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin. Multiple description coding in networks with congestion problem. *IEEE Trans. Inform. Theory*, 47(3):891–902, Mar. 2001.

[5] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee. On multiple description streaming with content delivery networks. In *Proc. of IEEE INFOCOM*, pages 1736–1745, New York, NY, June 2002.

[6] J.G. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In *Proc. SPIE VCIP*, pages 392–409, Jan. 2001.

[7] J.G. Apostolopoulos and S.J. Wee. Unbalanced multiple description video communication using path diversity. In *Proc. IEEE ICIP*, pages 966–969, Oct 2001.

[8] J.G. Apostolopoulos, T. Wong, W. Tan, and S. Wee. On multiple description streaming in content delivery networks. In *Proc. IEEE INFOCOM*, pages 1736–1745, June 2002.

[9] T. Back, D. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford University Press, New York, NY, 1997.

[10] N. Banerjee and S.K. Das. Fast determination of QoS-based multicast routes in wireless networks using genetic algorithm. In *Proc. IEEE ICC*, pages 2588–2592, June 2001.

[11] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, Inc., New York, NY, second edition, 1990.

[12] A.C. Begen, Y. Altunbasak, and O. Ergun. Multi-path selection for multiple description encoded video streaming. *EURASIP Signal Processing: Image Communication*. to appear.

[13] A.C. Begen, Y. Altunbasak, and O. Ergun. Fast heuristics for multi-path selection for multiple description encoded video streaming. In *Proc. IEEE ICME*, pages 517–520, July 2003.

[14] A.C. Begen, Y. Altunbasak, and O. Ergun. Multi-path selection for multiple description encoded video streaming. In *Proc. IEEE ICC*, pages 1583–1589, May 2003.

[15] L. Blazevic, S. Giordano, and J.Y. Le Boudec. Self organized terminode routing. *J. of Cluster Comput.*, 5(2):205–218, April 2002.

[16] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput. Surveys*, 35(3):268–308, Sept. 2003.

[17] J. Chakareski, S. Han, and B. Girod. Layered coding vs. multiple descriptions for video streaming over multiple paths. In *Proc. ACM Multimedia*, pages 422–431, Nov. 2003.

[18] P.A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. Technical report, Microsoft Research, MSR-TR-2001-35, Feb. 2001.

[19] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol, Oct. 2003. IETF RFC 3626.

[20] Y. Dong, Z.-L. Zhang, and Y.T. Hou. Server-based dynamic server selection algorithms. *Springer-Verlag Lecture Notes in Computer Science (LNCS)*, 2344:575–584, 2002.

[21] R. Elbaum and M. Sidi. Topological design of local-area networks using genetic algorithms. *IEEE/ACM Trans. Networking*, 4(5):766–778, Oct. 1996.

[22] D. Eppstein. Finding the k shortest paths. *SIAM J. on Comput.*, 28(2):652–673, Aug. 1999.

[23] 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. Syst., Man, Cybern. C*, 33(3):313–324, Aug. 2003.

[24] S.C. Ghosh, B.P. Sinha, and N. Das. Channel assignment using genetic algorithm based on geometric symmetry. *IEEE Trans. Veh. Technol.*, 52(4):860–875, July 2003.

[25] F. Glover and M. Laguna. *Tabu Search*. Kluwer-Academic, Boston, MA, 1997.

[26] N. Gogate, D. Chung, S.S. Panwar, and Y. Wang. Supporting image/video applications in a multihop radio environment using route diversity and multiple description coding. *IEEE Trans. Circuits Syst. Video Technol.*, 12(9):777–792, Sept. 2002.

[27] V. Goyal. Multiple description coding: compression meets the network. *IEEE Signal Processing Mag.*, 18:74–93, Sept. 2001.

[28] E. Gustafsson and G. Karlsson. A literature survey on traffic dispersion. *IEEE Network*, 11(2):28–36, Mar. 1997.

[29] J. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.

[30] 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*, pages 7–10, Oct. 2002.

[31] F.K. Hwang, D.S. Richards, and P. Winter. *The Steiner Tree Problem (Annals of Discrete Mathematics, Vol.53)*. North-Holland, 1992.

[32] D.B. Johnson, D.A. Maltz, and Y.-C. Hu. The Dynamic Source Routing protocol for mobile ad hoc networks (DSR), April 2003. IETF Internet Draft draft-ietf-manet-dsr-09.txt.

[33] A. Kopke, A. Willig, and H. Karl. Chaotic maps as parsimonious bit error models of wireless channels. In *Proc. IEEE INFOCOM*, pages 513–523, Mar. 2003.

[34] J. Liu, B. Li, Y.T. Hou, and I. Chlamtac. On optimal layering and bandwidth allocation for multisession video broadcasting. *IEEE Trans. Wireless Commun.*, 3(2):656–667, March 2004.

[35] N. Malpani and J. Chen. A note on practical construction of maximum bandwidth paths. *Inform. Processing Letters*, 83(3):175–180, Aug. 2002.

[36] 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*, March 2005.

[37] S. Mao, S. Kompella, Y.T. Hou, H.D. Sherali, and S.F. Midkiff. Optimal routing for multiple concurrent video sessions in wireless ad hoc networks. In *Proc. IEEE ICC*, May 2005.

[38] 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. Select. Areas Commun.*, 12(10):1721–1737, Dec. 2003.

[39] S. Mao, X. Lin, Y.T. Hou, and H.D. Sherali. Multiple description video multicast in wireless ad hoc networks. *ACM/Kluwer Mobile Networks and Applications Journal (MONET).* to appear.

[40] Prasant Mohapatra, Chao Gui, and Jian Li. Group communications in mobile ad hoc networks. *IEEE Computer*, 37(2):70–77, Feb. 2004.

[41] S. Murthy and J.J. Garcia-Luna-Aceves. Congestion-oriented shortest multipath routing. In *Proc. IEEE INFOCOM*, pages 1038–1036, May 1996.

[42] C.Y. Ngo and V.O.K. Li. Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms. *IEEE Trans. Commun.*, 51(9):1439–1441, Sept. 2003.

[43] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF), Feb. 2004. IETF RFC 3684.

[44] L. Ozarow. On a source coding problem with two channels and three receivers. *Bell Syst. Tech. J.*, 59(10):84–91, Dec. 1980.

[45] V.N. Padmanabhan, H.J. Wang, P.A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *Proc. ACM NOSSDAV*, pages 177–186, May 2002.

[46] P. Papadimitratos, Z.J. Haas, and E.G. Sirer. Path set selection in mobile ad hoc networks. In *Proc. ACM Mobihoc*, pages 1–11, June 2002.

[47] M. Papadopouli and H. Schulzrinne. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *Proc. of ACM MobiHoc 2001*, pages 117–127, Oct. 2001.

[48] Mehrdad Parsa, Qing Zhu, and J.J. Garcia-Luna-Aceves. An iterative algorithm for delay-constrained minimum-cost multicasting. *IEEE Trans. on Networking*, 6(4):461–474, Aug. 1998.

[49] A. Petrowski. A clearing procedure as a niching method for genetic algorithms. In *Proc. IEEE Int. Conf. Evolutionary Computation*, pages 798–803, 1996.

[50] A.R. Reibman, H. Jafarkhani, M.T. Orchard, and Y. Wang. Performance of multiple description coders on a real channel. In *Proc. IEEE ICASSP*, pages 2415–2418, March 1999.

[51] Sajama and Z.J. Haas. Independent–tree ad hoc multicast routing (itamar). *Mobile Networks and Applications*, 8(5):551–566, Oct. 2003.

[52] H.F. Salama, D.S. Reeves, and Y. Viniotis. Evaluation of multicast routing algorithms for real-time communication on high-speed networks. *IEEE J. Select. Areas Commun.*, 15(3):332–345, April 1997.

[53] B. Sareni and L. Krahenbuhl. Fitness sharing and niching methods revisited. *IEEE Trans. Evolutionary Computation*, 2(3):97–106, Sept. 1998.

[54] E. Setton, Y. Liang, and B. Girod. Adaptive multiple description video streaming over multiple channels with active probing. In *Proc. IEEE ICME*, July 2003.

[55] H.D. Sherali and W.P. Adams. A reformulation-linearization technique (rlt) for solving discrete and continuous nonconvex programming problems. *Mathematics Today, Special Issue on Recent Advances in Mathematical Programming*, XII-A:61–78, 1994.

[56] H.D. Sherali and W.P. Adams, editors. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Kluwer Academic, Dordrecht, Boston, MA, 1999.

[57] H.D. Sherali and P.J. Driscoll. On tightening the relaxations of miller-tucker-zemlin formulations for asymmetric traveling salesman problems. *Operations Research*, 50(4):656–669, July/Aug. 2002.

[58] H.D. Sherali, K. Ozbay, and S. Subramanian. The time-dependent shortest pair of disjoint paths problem: Complexity, models, and algorithms. *Networks*, 31(4):259–272, Dec. 1998.

[59] H.D. Sherali and C.H. Tuncbilek. A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7:1–31, 1995.

[60] M.C. Sinclair. Minimum cost wavelength-path routing and wavelength allocation using a genetic-algorithm/heuristic hybrid approach. *IEE Proc. Commun.*, 46(1):1–7, Feb. 1999.

[61] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE J. Select. Areas Commun.*, 17(8):1488–1505, Aug. 1999.

[62] A. Yener and C. Rose. Genetic algorithms applied to cellular call admission: Local policies. *IEEE Trans. Veh. Technol.*, 46(1):72–79, Feb. 1997.

[63] Q. Zhang and Y. Leung. An orthogonal genetic algorithm for multimedia multicast routing. *IEEE Trans. on Evolutionary Computation*, 3(1):53–62, April 1999.

Appendix A

Proofs

A.1 Proof of Property M1

Let $a = 2^{-2R_1} \leq 1$ and $b = 2^{-2R_2} \leq 1$. From (2.1) and (2.2), we have

$$\frac{1}{\sigma^2} \cdot \frac{\partial D}{\partial R_1} = -P_{00} \frac{2 \ln 2 \cdot ab^2}{(a + b - ab)^2} - 2 \ln 2 \cdot P_{01} \cdot a \leq 0$$

Similarly, we have $\frac{\partial D}{\partial R_2} \leq 0$ due to the symmetry in (2.1).

A.2 Proof of Property M2

For a disjoint path set $\{\mathcal{P}_1, \mathcal{P}_2\}$, we have that $p_{jnt} = 1$ and $\alpha = 0$. From (2.2) and (2.5), we have

$$\frac{1}{\sigma^2} \cdot \frac{\partial D}{\partial p_{dj}^1} = (a - 1) \left[\frac{(1 - p_{dj}^2)(a + b(1 - a)) + b^2 p_{dj}^2}{a + b(1 - a)} \right] \leq 0.$$

Similarly, we have $\frac{\partial D}{\partial p_{dj}^2} \leq 0$ due to the symmetry in (2.5).

A.3 Proof of Property M3

For solution $\hat{x} = \{\mathcal{P}_1, \mathcal{P}_2\}$ in Figure 2.2(a), let there be K joint links with parameters $\{\alpha_k, \beta_k\}$, $k = 1, \cdots, K$. We then have:

$$\frac{1}{\sigma^2} \cdot [D(\hat{x}) - D(\bar{x})] = \prod_{\{i,j\} \in \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)} p_{ij} \prod_{\{m,n\} \in \bar{\mathcal{J}}(\mathcal{P}_1)} p_{mn} \prod_{\{x,y\} \in \bar{\mathcal{J}}(\mathcal{P}_2)} p_{xy} \prod_{k=1}^{K-1} (1-\alpha_k)(1-p_K)(1-\alpha_K-\beta_K) \frac{(a+b)(1-a)(1-b)}{a+b(1-a)} \geq 0,$$

according to the "bursty" assumption in Property M3.

A.4 Proof of Proposition 1

The formation of the optimal solution $x^* = \{\mathcal{P}_1^*, \mathcal{P}_2^*\}$ could conform with one of the following two cases:

Case I: x^* is comprised of a pair of disjoint paths. From the construction procedure, x_l^* offers higher description rates than x^*, since b^* is optimal over all feasible paths. In addition, x_l^* offers higher end-to-end success probabilities for the descriptions than x^*, since p^* is optimal over all feasible paths. Then, we have that $D(x_l^*) \leq D(x^*)$ according to properties M1 and M2.

Case II: x^* is comprised of a pair of joint paths. Assume $\mathcal{P}_1^*$ and $\mathcal{P}_2^*$ share K links. Firstly, we construct a virtual solution $\bar{x} = \{\bar{\mathcal{P}}_1, \bar{\mathcal{P}}_2\}$, by (i) appending a copy of each shared link k to the disjoint portions of the two paths; and (ii) removing link k from the shared portion, $k = 1, \cdots, K$. That is, the resulting $\bar{\mathcal{P}}_h$ has the same set of links as $\mathcal{P}_h^*$, $h = 1, 2$, but $\bar{\mathcal{P}}_1$ is completely disjoint with $\bar{\mathcal{P}}_2$. As a result, x_l^* may not be feasible; an example of such construction operation is shown in Figure 2.2.

Secondly, we observe that the constructed solution $\bar{x}$ can provide a pair of description rates no lower than x^*, since the K links originally shared by the two descriptions are now used exclusively by each description. By applying Property M3 iteratively for K times, we have that $D(\bar{x}) \leq D(x^*)$.

Thirdly, from ALG-LB, we have that $b^* \geq B(\mathcal{P}_h^*)$, $h = 1, 2$ (recall that $B(\mathcal{P})$ is the end-to-end bandwidth of path $\mathcal{P}$). Note that $B(\bar{\mathcal{P}}_h) = B(\mathcal{P}_h^*)$, $h = 1, 2$, since they consist of the same set of links. Therefore, x_l^* dominates the constructed $\bar{x}$ in terms of end-to-end bandwidths (i.e., $b^* \geq B(\bar{\mathcal{P}}_h)$, $h = 1, 2$). Similarly, we can also show that x_l^* dominates $\bar{x}$ in terms of end-to-end success probabilities (i.e., $p^* \geq p_{dj}^h(\bar{\mathcal{P}}_h)$, $h = 1, 2$). According to properties M1 and M2, we have that $D(x_l^*) \leq D(\bar{x})$.

Finally, we have that $D(x_l^*) \leq D(\bar{x}^*) \leq D(x^*)$.

Vita

Xiaolin Cheng received his B.E. and M.E. degrees in automation from Tsinghua University, Beijing, China in 1997 and 2000, respectively. From 2000-2003, he was a senior engineer at Panasonic Beijing Labs, where he developed system software for digital TV set-top boxes. In August 2003, he joined Virginia Tech to continue his study in Computer Engineering in the Bradley Department of Electrical and Computer Engineering. His current research interests include multipath and multicast routing in wireless ad hoc networks and video transport over wireless ad hoc networks. Mr. Cheng is a student member of the IEEE.

List of Publications

Shiwen Mao, **Xiaolin Cheng**, Y. Thomas Hou, and Hanif D. Sherali, "Multiple description video multicast in wireless ad hoc networks," to appear in *ACM/Kluwer Mobile Networks and Applications Journal (MONET)*.

Shiwen Mao, **Xiaolin Cheng**, Y. Thomas Hou, Hanif D. Sherali, "Multiple description video multicast in wireless ad hoc networks," In *Proceedings BroadNets 2004*, Oct. 25-29, San Jose, CA.

Shiwen Mao, Y. Thomas Hou, **Xiaolin Cheng**, Hanif D. Sherali, "Multi-path routing for multiple description video over wireless ad hoc networks," In *Proceedings of IEEE Infocom 2005*, Miami, FL, March 13-17, 2005.

Shiwen Mao, **Xiaolin Cheng**, Y. Thomas Hou, Hanif D. Sherali, "Joint routing and server selection for multiple description video streaming in wireless ad hoc networks," In *Proceedings of IEEE ICC 2005*, Seoul, Korea, May 2005.