

OPTIMUM RESOURCE ALLOCATION IN MULTIPATH AD HOC NETWORKS

by

Petar Djukic

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

Copyright © 2003 by Petar Djukic

Abstract

Optimum Resource Allocation in Multipath Ad Hoc Networks

Petar Djukic

Master of Applied Science

Graduate Department of Electrical and Computer Engineering

University of Toronto

2003

We devise a mechanism for reliable packet transmission in Wireless Ad Hoc Networks. The implementation is based on *Path Diversification*, which uses multipath diversity in the network and erasure codes to provide guaranteed reliability. We show that when no information is available about the status of the network, distributing the load over all the available paths can increase network reliability significantly over single path transmissions.

We show how to collect information about the status of network paths and energy consumption in the network. When the information about the network is available, we identify and solve four QoS optimization problems specific to path diversification in ad hoc networks. We show how to allocate network resources for maximized reliability, maximized efficiency, minimized energy use, and maximized network lifetime. For the first two, we give exact polynomial time algorithms, we approximate the second two as linear programs.

Contents

1	Introduction	1
1.1	Challenges of Ad Hoc Networks	3
1.2	Quality of Service in Ad Hoc Networks	5
1.2.1	Packet Loss (Reliability)	6
1.2.2	Efficiency	7
1.2.3	Energy Consumption	7
1.2.4	Network Lifetime	8
1.3	Path Diversified Reliable Transport Layer	8
1.4	Path Diversification	10
1.5	Resource Allocation for Optimum QoS	13
1.6	Related Work	14
1.6.1	Reliable Transport in Wireless Ad Hoc Networks	15
1.6.2	Multipath Routing	16
1.6.3	Information Dissemination	17
1.6.4	Connection Stripping/Splitting	18
1.6.5	Energy Management in Ad Hoc Networks	19
2	Mathematical Preliminaries	23
2.1	Path Diversification	23

2.2	Network Model	25
2.3	Network Reliability P_{succ}	30
2.3.1	Exact P_{succ}	31
2.3.2	Poisson Approximation of P_{succ}	35
3	Optimum Resource Optimization	37
3.1	Blind Path Diversification	37
3.2	Maximization of P_{succ} (P1)	43
3.2.1	Optimization of P_{succ} through the Poisson Approximation	47
3.2.2	Optimization of P_{succ} through the Normal Approximation	49
3.3	Maximization of Efficiency (P2)	50
3.4	Minimization of Energy Consumption (P3)	55
3.5	Maximization of Network Lifetime (P4)	61
4	Numerical Results	65
4.1	Results for P1, Maximization of P_{succ}	65
4.1.1	Resource-Unconstrained optimization of P_{succ}	66
4.1.2	Constrained optimization of P_{succ}	70
4.2	Results for P2, Maximization of Efficiency	74
4.3	Results for P3, Minimization of Energy Consumption	77
4.4	Results for P4, Maximization of Network Lifetime	82
5	Conclusion	87
A	Convexity of $Q(\mathbf{m}, K)$	89
B	Efficiency of Selective-Repeat ARQ Protocol	91
	Bibliography	92

List of Tables

4.1 The Log-Normal Transformation	66
---	----

List of Figures

1.1	Ad Hoc Networks vs. Infrastructure Networks	1
1.2	The Logical View of the Network	9
1.3	The Reliable Transport Layer	9
1.4	Connection Splitting Trade-Off	11
1.5	Data Dissemination Trade-Off	12
2.1	Mathematical Model of Path Diversification	24
2.2	Long Term TCP Throughput vs. the Reliability	31
3.1	Markov Chain Channel Model	38
3.2	Performance of Blind Diversification for Gilbert-Elliott Channel	40
3.3	Performance of Blind Diversification for Markov-Chain Channel	41
3.4	Minimum Reliability of Blind Diversification for the 5-state Markov-Chain Channel	42
3.5	Efficiency of Selective-repeat ARQ with Blind Diversification	43
3.6	Minimum Efficiency δ	53
3.7	$\alpha(k)$ as the function of ϵ	57
3.8	$s(\epsilon)$ vs. ϵ	60
4.1	Reliability of <i>Path Diversification</i>	67
4.2	Comparison to the Normal and Poisson Approximation	69
4.3	QoS Constrained Reliability vs. Efficiency	70

4.4	Blocking Probability vs. Efficiency	71
4.5	QoS Constrained Reliability of <i>Path Diversification</i>	73
4.6	Minimum Reliability vs. Maximum Efficiency	74
4.7	Minimum Reliability vs. Maximum Efficiency	75
4.8	Minimum Reliability vs. Blocking Probability	76
4.9	Minimum Energy vs. Minimum Reliability ϵ	77
4.10	ΔK as a Function of ϵ and Energy Range	78
4.11	Error in Energy Minization	79
4.12	Normalized Minimum Energy	80
4.13	Efficiency at Minimum Energy	81
4.14	Decrease in Network Lifetime with Network Reliability	82
4.15	Effect of P4 on Efficiency	84
4.16	The Effect of Energy Consumption on Network Lifetime	85

List of Algorithms

1	EVALUATE-PSUCC($\mathbf{m}, \mathbf{q}, K$)	32
2	MAXIMIZE-PSUCC($\mathbf{q}, K, \mathbf{M}_{th}$)	45
3	OPTIMIZE-UTILIZATION($\mathbf{M}_{th}, \mathbf{q}, \epsilon, \delta$)	54

Chapter 1

Introduction

Wireless Ad Hoc Networks are a collection of two or more devices equipped for wireless communications and networking capabilities. The nodes can communicate directly with nodes in their range, or communicate with nodes outside of their range by using other nodes to forward their packets [1]. Ad hoc networks are different from infrastructure wireless networks in that in infrastructure networks all peer-to-peer communications between nodes are done through the Access Point as shown in Figure 1.1. Ad hoc networks are self-organizing in a sense that all the routes in the network have to be discovered and established by the nodes in the network through direct peer-to-peer communications.

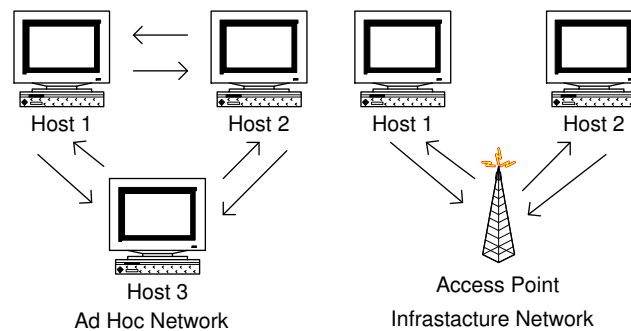


Figure 1.1: Ad Hoc Networks vs. Infrastructure Networks

Some examples of ad hoc networks include:

- *Military packet radio networks*

The original use for ad hoc networks came from the early military wireless packet networks. [1] describes PRNET, the packet radio network used in the military in the 1970's. At the time, the manufacturing cost and size of devices made packet radio networks impractical. However, the situation has changed recently. [2] describes the new military radio network, Future Combat System (FCS). This is a hierarchical network with an ad hoc sensor network at the bottom. Foot soldiers are also connected between themselves in an ad hoc network and hooked up to the tactical command vehicles, working all the way up to satellite links to the headquarters. This network was supposed to be implemented in year 2002. Another application of this type of network in civilian domain may be in emergency services.

- *Sensor networks*

Sensor networks consist of many, normally very small size devices that monitor a certain phenomena. Terminals in a sensor network would normally work together to deliver the collective data to a sink which would forward them to the observer. [3] gives a survey of sensor networks, it gives examples of sensor networks for military, environmental, health and home applications.

- *Large scale ad hoc networks*

Another use for ad hoc network is “for the masses”. Such a network connects everyone on the planet in a single ad hoc network. [4] describes the use of large scale ad hoc networks in the Terminodes project. The goal of the Terminodes project is to identify and solve problems associated with large scale ad hoc networks. The idea is to divide the whole network into smaller partitions where the nodes self-organize themselves into ad hoc networks. The regions are connected through anchors that are connected through private high-speed connections.

The rest of the introduction shows the challenges of ad hoc networks, defines the set of Quality of Service (QoS) parameters used in the rest of the thesis, and gives the overview of the new mechanism for transport services that will be presented in the rest of the thesis. Next, we describe some of the specific challenges in ad hoc networks, motivate the need for a new transport service in ad hoc networks, and give an outline of the thesis.

1.1 Challenges of Ad Hoc Networks

Research in ad hoc network is well established with many works improving on deficiencies these networks. [1,3,4], each give an overview of some of the difficulties of implementing ad hoc networks. Three well-known problems in ad hoc networks are the lack of reliable packet delivery due to interference and movement of nodes, limited bandwidth due to channel restrictions, and limited node lifetime due to small battery size. Research in Quality of Service issues in wireless ad hoc network is mostly oriented to extending the lifetime of the network (in terms of power), fairness of access to the network, and dealing with issues of connections between heterogeneous nodes. This thesis is motivated by the lack of research in providing reliable transport services, and the lack of research in providing QoS guarantees for traffic flows in Mobile Ad Hoc Networks.

The thesis addresses the three problems by introducing the *Path Diversification* mechanism in the transport layer. In path diversification, each packet is decomposed into a number of smaller packets, which are called *fragments* in the thesis. The fragments are deployed over multiple parallel paths. To increase reliability, path diversification also uses an erasure code to generate extra parity fragments for each packet. Destination can recover the original packet when it receives a certain number of fragments. The thesis shows that path diversification allows us to provide *guaranteed* QoS in ad hoc networks, without changing the existing routing or MAC layer protocols. At the same time, path diversification increases the available bandwidth, and performs load balancing in the network and increases lifetime

of the network by distributing the cost of packet transmissions among nodes from different paths. Next, we review the deficiencies of ad hoc networks in the areas of unreliable communications, limited bandwidth capacity, and restrictions due to limited energy capacity.

Great difficulty in implementing ad hoc networks comes from frequent route changes, due to mobility of the nodes, and interference among nodes. The high packet loss rates and frequent topological changes make the transport layer unstable and limit the amount of traffic that the network can carry. For example Automatic Repeat Request (ARQ) approaches in ad hoc networks are ineffective due to high packet loss rates [5]. QoS reservation schemes are also ineffective due to frequent topological changes [6, 7]. The path diversification approach used in this thesis overcomes these problems by using erasure codes as well as multiple paths in the network to decrease the packet loss rate in the network. The thesis shows that path diversification provides a mechanism to provide guaranteed packet loss rates that can be arbitrarily low.

Ad hoc networks have limited bandwidth because of the wireless channel capacity. However, ad hoc networks suffer an additional problem due to the fact that all nodes share the same channel. [8] gives the limit for link capacity of stationary wireless ad hoc networks as $\lambda(n) = \Theta(W/\sqrt{n \log n})$, where W is the raw channel capacity for networks with n randomly distributed nodes with random traffic patterns. This link capacity assumes no overhead in the network. Clearly, it is inconceivable that ad hoc network will ever expand beyond a small number of nodes, and still be able to provide the full bandwidth available in the physical layer. In the view of this, the Terminodes project [4] sections the large scale network into smaller parts, and most sensor networks consist of a relatively small number of nodes. This thesis uses load balancing to address the problem. Intuitively, [8] points out to an extreme situation where many of the nodes have to communicate through the same node in order to reach the end of the network. However, if the network is dense and uses load balancing, this will not be an issue, since the load is always distributed among many nodes. Path diversifica-

tion provides load balancing and at the same time it is also a connection splitting technique that allows us to combine the bandwidth along multiple overlapping paths to provide larger bandwidths than what would normally be available.

Limited size of network terminals and cost of manufacturing makes the energy a limited resource in ad hoc networks. The network does not exist if the wireless terminals do not have the power to transmit. The current research addresses the problem of energy consumption and network lifetime in the network and MAC layers. The thesis shows how path diversification can be used in the optimal way to keep energy consumption minimized from the connection point of view and maximize the lifetime of the network from the connection point of view.

We define the set of QoS parameters used in the thesis to evaluate the performance of path diversification in Section 1.2. We give a context in which path diversification is used in Section 1.3 and a brief overview and motivation for path diversification in Section 1.4. Section 1.5 defines the four optimizations solved in this thesis. The four optimizations address the problems of ad hoc networks outlined in this section. We give the overview of related work in Section 1.6.

Chapter 2 defines the mathematical model of path diversification and gives assumptions used in the rest of the thesis for solving the four optimizations. Chapter 3 is the main contribution of this thesis and shows how path diversification can be used in the optimal way to solve the four problems we identify in Section 1.5. Chapter 4 presents the numerical results to show the performance of path diversification.

1.2 Quality of Service in Ad Hoc Networks

QoS has a different meaning depending on the context in which it is applied and analyzed. We now define a clear set of QoS parameters that will be used throughout the thesis to measure the effectiveness of path diversification in the transport layer. The QoS parameters

used in the thesis can also be *measured* for Service Level Agreement (SLA) enforcement. We will use a subset the QoS parameters defined in [9], and modify them to fit the model and assumptions in the thesis.

The QoS parameters we use are *Packet Loss (Reliability)*, *Efficiency*, *Energy Consumption*, and *Network Lifetime*. The first two QoS parameters are network based and relate to the services the network can provide. The second two QoS parameters are user-based and relate to the cost the user is willing to have a connection in the ad hoc network. We use the packet loss in the network to show that path diversification can increase the reliability of ad hoc networks to the level of wireline networks. However, we also need to compare the efficiency of the scheme to other ways of eliminating packet loss in the network. We use Energy and Network lifetime to show how path diversification can be adapted to achieve other QoS goals in the network; at the same time we show that path diversification provides a good mechanism to decrease energy use in the network and increase the lifetime.

1.2.1 Packet Loss (Reliability)

The well established QoS parameters in network research are delay and packet loss. We define the QoS in terms of delay and packet loss as follows:

$$\Pr[\text{Delay} > D_M] \leq \epsilon_D \quad \text{and} \quad \Pr[\text{Loss}] \leq \epsilon_L \quad (1.1)$$

In the transport layer protocol considered in this thesis the receiver node ignores all packets that take more than D_M time to get to the destination. So, we can set $\epsilon_D = 0$ and measure the basic QoS in the network in terms of packet loss ϵ_L . Improving this packet loss statistics is crucial to making ad hoc networks more feasible. For example, ARQ based transport protocols such as TCP must retransmit a large number of packets to ensure the reliable delivery, so there is a large overhead when too many packets are lost in the network. The

transport layer in this thesis decreases the number of acknowledgements by using erasure codes to recover the packets that may have lost fragments in the network. We will use packet loss and reliability in the thesis interchangeably.

1.2.2 Efficiency

We consider the utilization of the network resources as another QoS parameter in the network. The resources in the network are scarce, so it is important to improve the efficiency, or utilization of network resources. In this thesis, the efficiency is defined as follows:

$$\eta = \frac{\text{Effective Throughput}}{\text{Actual Throughput}}. \quad (1.2)$$

Where the effective throughput is the total amount of information offered by a connection to the network and the actual throughput is the information offered by the connection and the overhead of the transport layer considered together. We will compare the efficiency of path diversification to the efficiency of regular ARQ protocols used to combat the packet loss.

1.2.3 Energy Consumption

Energy consumption is a very important QoS parameter in ad hoc networks, due to the limited battery lifetime of the mobile terminals. Energy consumption in the node is directly related to the lifetime of the node, since the lifetime of the node depends on the node's available power. The energy consumption is also related to pricing in ad hoc networks. For example, charging connection in terms of used bandwidth may not be appropriate since it does not take into account the true cost of connections in the network. Charging by the amount of energy spent in the network may be a better pricing model for ad hoc networks.

We define energy consumption by a connection in the network as the total energy required to transmit all the packets for the connection through the network. We will show how to

minimize the amount of energy used by connections on a node and so increase the lifetime of the node. We also give a practical method for measuring energy consumption, and the amount of energy available on each path in the network.

1.2.4 Network Lifetime

We define the network lifetime as the time after which at least one node will lose all of its power. This is a widely accepted way to model network lifetime ([10, 11, 12]). Loss of a node has a serious impact on the whole network in terms of resources that are required to deal with the loss. Usually, this means that some of the routes need to be transferred and that other nodes need to pick the traffic from the node that is out of service. So, loss of a node seriously degrades the quality of service available in the network. The thesis will show how path diversification can be used to maximize the lifetime of the network.

1.3 Path Diversified Reliable Transport Layer

We now describe the context in which path diversification is used to increase reliability in wireless ad hoc networks. We use path diversification as the integral part of a new transport layer, the Path Diversified Reliable Transport Layer (PD-RTL). The PD-RTL provides more reliable network services to upper layers such as UDP and TCP. The implementation details of PD-RTL are not a part of this thesis. Here, we describe a context in which path diversification can be used in the network.

We start by explaining the logical division of the nodes in the network. The network is divided into Source and Sink Nodes (S-Nodes) and Forwarding Nodes (F-Nodes), as shown in Fig. 1.2. PD-RTL is located on each of the S-nodes and provides a socket interface to the applications using the network. On the S-Nodes PD-RTL, performs the identification of the available paths in the network, encoding of the data into the fragments, admission control

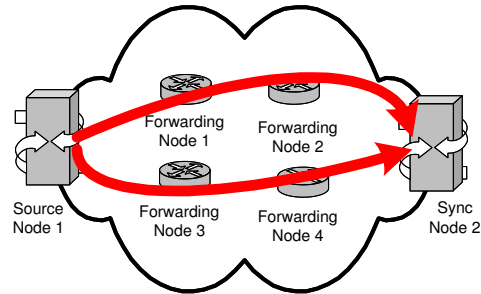


Figure 1.2: The Logical View of the Network

and the optimization of the network resources; the F-Nodes perform packet forwarding in the network. Conceptually, an F-Node is the part of the implementation on a node that performs the duties of the Network and MAC layers; the S-Node is the part of the node that implements the reliable transport.

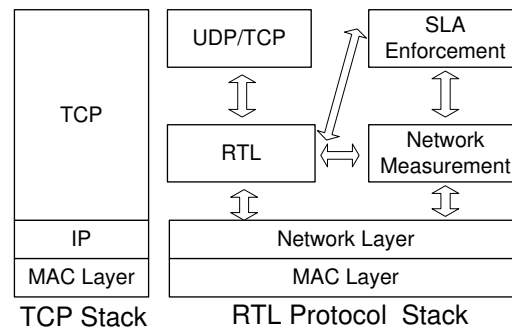


Figure 1.3: The Reliable Transport Layer

Fig. 1.3 shows the components of the PD-RTL and their relationship to the rest of the network. PD-RTL creates a more reliable network on top of the ad hoc network in order to improve the performance of the higher network layers. PD-RTL can be envisioned as either a sublayer of the transport layer or the network layer. On the source, PD-RTL performs the striping/splitting of TCP/UDP connection on multiple paths in the IP layer similar to [13] and it generates the extra parity fragments from the packets offered by the TCP/UDP connection. On the destination node, PD-RTL reassembles the packets from the received

fragments and hands them off to the higher layer. The “network” layer is implemented with a multipath routing protocol [14, 15, 16] which allows access to multiple paths in the network.

PD-RTL performs the optimization of the network resources as discussed in the thesis. PD-RTL relies on network measurements to find information about the network. We assume that PD-RTL can use either a probing mechanism [17] or an egress control mechanism [18, 19]. PD-RTL can also act as a distributed call admission controller for the network.

The “SLA Enforcement” module in Fig. 1.3 provides a mechanism to have agreements between the connections entering the network, and the PD-RTL. Based on these agreements PD-RTL can decide to drop connections if there are too many SLA violations in the transport layer.

The actual implementation details of PD-RTL are beyond the scope of this thesis. The main focus of the thesis is to show feasibility of using path diversification to implement such a reliable transport layer in ad hoc networks.

1.4 Path Diversification

In this work, we propose a strategy for reliable packet transmission that uses *path diversification* to guarantee reliability in wireless ad hoc networks. This technique uses *connection splitting* and *information dispersal* to achieve improved packet loss statistics. Connection splitting combines multiple parallel paths in the network to carry the data stream, and information dispersal uses erasure codes to replace packets that may have been lost. We now give the motivation for using path diversification, by giving some results for connection splitting and the data dissemination used alone.

The main problem the thesis solves is to increase the network reliability (decrease packet loss) in ad hoc networks. We would like to avoid packet retransmissions since this is very costly in wireless networks where link reliability is low. The intuition behind path diversification is twofold. First, path diversification uses erasure codes to recover the packets at

the receiver node. Second, path diversification uses the diversity of paths in the network to increase the reliability packet transmissions in the network.

We now motivate the need for multiple paths and erasure codes by using a very simple model in which all paths have the same packet loss probability, and accumulative packet losses are independent of each other. We will use a more precise network model to analyze the performance of path diversification in later chapters.

First, we show that we can increase the reliability in the network by transmitting identical data on multiple paths even without the use of erasure codes. We show this with Figure 1.4, where we plot the probability of not having to do any retransmissions. Clearly, we can achieve an arbitrarily high level of reliability in the network, as the number of paths n increases. However, if all connections in the network are “greedy”, in the way that the effective bandwidth of the source matches the effective capacity of the path, the efficiency of the scheme will be $\eta = 1/n$. So, as the reliability increases, so does the inefficiency.

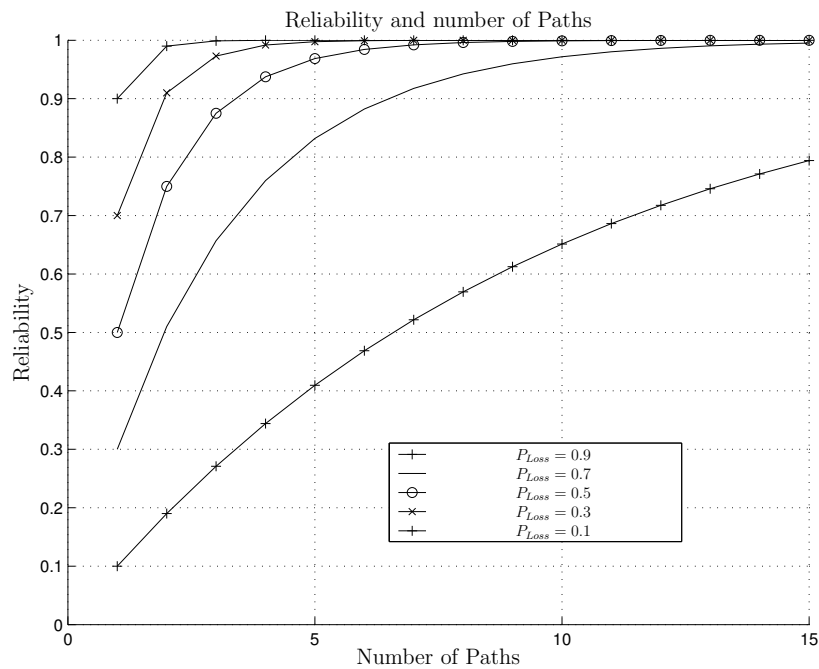


Figure 1.4: Connection Splitting Trade-Off

Second, we show that we can also increase reliability by using erasure codes. Our scheme uses information dispersal, proposed as an error correcting scheme for systems with multiple processors [20]. Information dispersal is used at the source node. The source node takes a data packet and it breaks it into fragments. The source node also generates additional parity fragments for the data packet. The additional fragments allow the receiver to reconstruct the original data stream even if some of the fragments are missing. Figure 1.5 shows the trade off between the efficiency of the information dispersal and the reliability of the path. We can increase reliability arbitrarily by using a high of parity fragments.

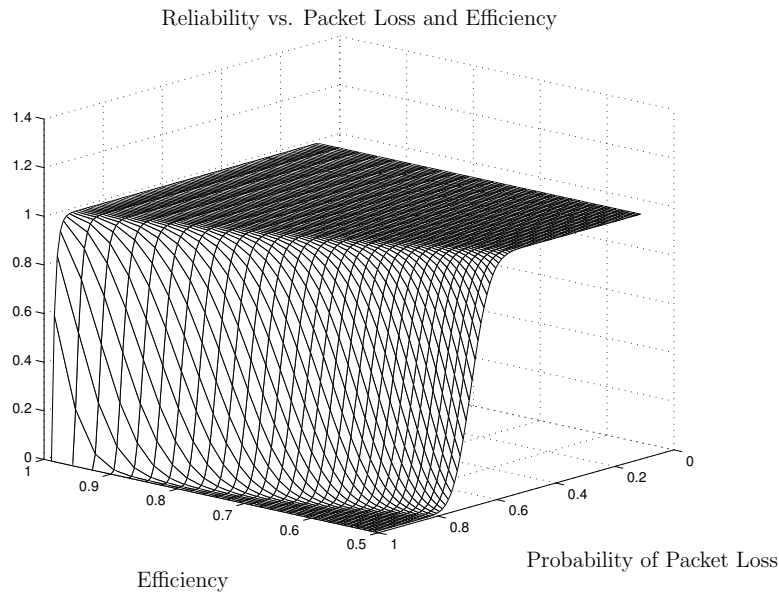


Figure 1.5: Data Dissemination Trade-Off

We show in Chapter 3 that we can combine the transmission over multiple paths with information dispersal to improve the efficiency of the individual schemes with path diversification.

1.5 Resource Allocation for Optimum QoS

This thesis has two objectives:

- We show that path diversification can increase reliability in an ad hoc network.
- We also show that path diversification can be used in an optimal way.

For the first objective of the thesis we assume that no prior information about the performance metrics along the paths is available. In this case, the transmitter simply distributes data uniformly over parallel paths. We call this *blind path diversification*. We show that blind path diversification outperforms single-path transmission. We show the performance of blind path diversification in Section 3.1.

For the second objective, we assume that some performance metrics about the parallel paths are available at the transmitter. These metrics are provided by a feedback from the network to the source or by some form of probing initiated by the source. We assume that the metrics are updated periodically. These metrics can be used to optimally assign the fragments into the available paths. In this case, we examine and solve the following four optimization problems related to the QoS management:

P1 Maximize the reliability in the network.

P2 Maximize the efficiency of path diversification, for a given reliability in the network.

P3 Minimize the energy used by connections, for a given reliability in the network and efficiency in the network.

P4 Maximize the lifetime of the network for a given reliability in the network, efficiency and maximum energy used in the network.

The source node uses the optimization algorithms associated with P1-P4 to periodically change the load on each path. We solve the four problems in Chapter 3.

We solve P1 in Section 3.2. We show that when there are no resource restrictions on the paths, the optimum way to allocate the packets is to transmit all the packets on the path with the lowest packet loss probability. We also show that when there are resource restrictions on the paths, the optimum way to allocate the packets is to use a “greedy” algorithm. The greedy algorithm allocates the maximum number of packets possible to the path with the smallest packet loss probability and then the maximum possible number of packets to the path with the second smallest packet loss probability, and so on.

We solve P2 in Section 3.3. We give an algorithm that performs a linear search for the least number of parity fragments for which the minimum required reliability is satisfied. The algorithm increases the number of parity fragments until it reaches the reliability threshold. Each new fragment is allocated with the greedy algorithm in P1.

We solve P3 in Section 3.3. The programming problem we arrive at initially is very difficult to solve due to the non-linearity of the reliability function, so we use an approximation. We show in Section 2.3 that reliability can be lower bounded by the Poisson cumulative distribution function. For P3, we use this lower bound of reliability to arrive at a linear programming problem for energy minimization. The linear program is solved to get the allocation of fragment on each path that minimizes energy use in the network.

We solve P4 in Section 3.3. Similar to P3, the initial programming problem we arrive at initially is very difficult to solve due to the non-linearity of the reliability function, so we use the Poisson approximation for this optimization as well. The linear program is solved to get the allocation of fragment on each path that maximizes the lifetime of the network. We present the numerical results for the four optimizations in Chapter 4.

1.6 Related Work

Path diversification was first proposed as a reliable routing technique in [21]. Similarly, [22] uses this technique at the physical layer. The deficiency of both is that they assume that

each packet is transferred with at most one transmission on each of the available paths. This requires a large number of paths to be available in the network in order to produce good reliability results.

Path diversification, as used in this thesis, was first proposed as a technique for wireless ad hoc networks in [23]. However, [23] did not show how to use this technique in the optimal way, or in a network with limited resources. In this work, we give algorithms that optimize the use of network resources for maximum reliability, maximum efficiency, minimum energy, and maximum lifetime in path diversified networks. We also give a practical way to model resources in the network. Next, we review current research in the area of reliable transport in ad hoc networks, multipath routing, information dissemination, connection stripping/splitting and energy management in ad hoc networks.

1.6.1 Reliable Transport in Wireless Ad Hoc Networks

Reliable transport in ad hoc networks has been a topic of recent research. The initial push for new transport services came with realizing that TCP is not an appropriate transport layer for ad hoc networks. TCP becomes unstable in networks with large packet loss since the congestion control is tied to the acknowledgment packets. In [24], the authors propose a change to the IEEE 802.11 MAC protocol to improve the performance of the TCP specifically. However, the paper does not make the change at the transport layer were the real problem is. In [25], the authors propose a new set of enhanced TCP/IP sockets for reliable network connections in mobile ad hoc networks. The enhanced sockets use *heartbeat* packets to detect link and path failures, however the work does not propose any technique to increase the level of reliability in the network. [6] and [7] both propose a QoS protocol for wireless ad hoc networks based on a reservation scheme. The protocols do not perform well in the presence of rapid topological changes in the network.

[5] gives a reliable transport protocol that disseminates information through the whole ad

hoc network. This transport layer is based on the idea that as the number of hops increases the likelihood of successful transmission decreases, so the protocol uses the intermediate nodes to provide packets requested through the negative acknowledgments. Eventually all the nodes in the network will have the transmitted information. [26] gives a transport layer protocol that improves on the performance of TCP. The idea behind the protocol is to decouple the congestion and reliability control in TCP since this makes TCP unstable in ad hoc networks. The protocol still uses acknowledgments to ensure reliable delivery of packets in the network.

[27] gives an overview of fault recovery schemes in computer networks and examines their usefulness for guaranteed performance networks. The paper also gives a new fault recovery technique and shows empirically that it outperforms other fault recovery techniques.

1.6.2 Multipath Routing

Multipath routing is crucial to the successful operation of path diversification. There is a large amount of research available in this area, showing that multipath routing is feasible. [28] and [29] give surveys of available multipath routing protocols. [28] also gives three multipath routing protocols that can be used in the Internet. [29] gives a path selection algorithm that chooses a set of *edge-disjoint* paths in wireless ad hoc networks with the lowest probability of path failure. This routing algorithm uses a heuristic to find the set of edge-disjoint paths with the lowest probability of failure. [30] gives properties of reliable networks and solves basic reliability problems in network routing.

There are many multipath routing algorithms available specifically for ad hoc networks. For example, DSR [14] is a source based algorithm that discovers all routes in the network by using broadcasts to propagate the route requests. The routing protocol also allows the source node to specify the exact route each packet should use. [15] gives an improved multipath version of DSR in which edge-disjoint routes are found and the alternate routes are known to

all the nodes on a path. [16] gives a modified version of AODV routing protocol [31] that finds multiple edge-disjoint routes in the network and makes them known to all the nodes in the network. [32] gives a QoS ticket based routing protocol for Ad Hoc Networks. The protocol searches multiple paths at once to find the path that satisfies the QoS requirements of the connections on the source node. The protocol reduces the overhead of broadcast multipath routing by limiting the number of paths searched at each node through the source issued “tickets”.

Path diversification is not dependent on the specific technique used in the network layer, as long as there is a set of multipaths between the source and destinations nodes.

1.6.3 Information Dissemination

Information dissemination refers to adding erasure code parity fragments to the original data. It was originally proposed as a load balancing technique, however it has since been used in other contexts. Various information dissemination approaches are shown in [20, 33, 34, 35]. These papers show how to use information dissemination for load-balancing and reliable multicast content delivery. [20] originally introduces this concept by using linear block codes to improve information sharing between multiple processors.

In [36] and [33], the authors describe a Digital Fountain delivery service for downloading information from multiple mirror site. The approach is based on the information dissemination with “tornado” Codes [37]. In this scheme, each of the mirror servers sends out a stream of packets generated from the dissemination of the same file. The client needs some subset of these packets to reassemble the original file. This approach eliminates the need for the acknowledgment of every packet, and greatly increase utilization of the multicast servers in the network. [38] analyzes a system where TCP is used to access multiple access servers at the same time. This approach was extended to wireless ad hoc networks in [34]. [39] gives an overview of the techniques for real-time data transport in single path networks. The

paper deals with protocols for real-time multimedia broadcasts where the start up latency is important to clients which listen on the broadcast channels. The paper improves the performance of [33] by introducing periodic broadcasts, and multiple channels to transmit the real-time broadcast traffic.

[35] uses the linear block codes to implement a collaborative multicasting system in wireless environments. The use of codes in the system provides higher reliability in the information exchange, as well as more efficient communications since the authors are not using an underlying reliable transport layer. This system can be used to provide a set of commonly used files to all nodes in the network. However, the system does not cover peer-to-peer communications, and it does not make Quality of Service guarantees such as bounds on Delay and Loss. The authors conjecture that efficiency of dissemination should be decreased with mobility in the network. This approach builds on [40], where simple data replication is used to create a distributed information sharing system in Ad Hoc Networks.

[41] compares several implementations of database access techniques in infrastructure wireless networks. The strategies are compared in terms of access time required to retrieve a records from the database, with the goal of decreasing the access time when the whole database is broadcast in regular intervals. Path diversification was recommended for channels with high error levels, however the paper does not fully explore the reliability benefits of path diversification, nor the optimal way to use it.

1.6.4 Connection Stripping/Splitting

Connection splitting was an important bandwidth aggregation approach in the early 1990's when the wireline equipment could not provide the bandwidth requested by users. [13] gives a framework for connection splitting (striping) and defines elements needed to implement it. [42] gives a bandwidth aggregation technique for telephone modems. [43] gives more example of connection stripping at the physical layer. [44] adapts this approach for a standard

specific to splitting ATM cells over multiple point-to-point physical links. However, with better equipment, this approach has evolved, so that it can be used even in the network layer. [45] uses connection splitting over multiple physical links in ATM networks to achieve better reliability.

[46] gives a protocol for connection splitting in ATM networks. The motivation behind this protocol was to use the ATM connections more efficiently by fragmenting the connections into smaller connections. The paper points out to the synchronization problem inherent to connection splitting, and tries to solve it through scheduling. In [47], the authors propose a bandwidth aggregation solution for TCP/IP links in ad hoc environments. Even though this solution uses network splitting it gives no QoS guarantees to traffic flows using the network.

1.6.5 Energy Management in Ad Hoc Networks

We now review several approaches to energy/power management in ad hoc networks. Energy aware routing has been a topic of research for some time. The current approaches involves schemes in which the energy consumption is made to be a part of the routing parameters, or schemes that involve minimizations in the MAC layer. Our approach of path diversification gives us the unique opportunity to manage energy/power at the transport layer, and so not have to modify the underlying layers.

[48] shows how to implement a directional antenna protocol in ad hoc networks which decreases the total energy use. The energy consumption is decreased in two ways with the use of directional antennas which use less energy to transmit, and with energy efficient routing in the network with directional antennas. [49] gives a MAC protocol that reduces energy consumption in the network regardless of the network layer. The protocol powers off the nodes which are not transmitting due to collision avoidance of the MAC layer. [50] calculates the transmission range (power level) in the ad hoc network for minimal bit error due to multiuser interference in the network. [51] and [52] give a set of optimizations using

convex programming; these optimizations maximize the throughput in a CDMA ad hoc network by optimizing the power transmission levels, given the multiuser interference in the network.

[10] gives a comparison of three different power management techniques in ad hoc networks, for the layers above the MAC layer. Two of the strategies include minimization of per packet energy, and maximization of time to network partition. The paper also uses a heuristic based cost function to implement energy-aware routing through the shortest distance routing. [11] gives comparison of several routing protocols that minimize the energy use and maximize the lifetime of the ad hoc network. The paper points out the differences between energy minimal protocols and the max-min fairness based protocols and through heuristic arguments creates a new protocol to satisfy both. [11] also gives routing protocol similar to [10]. This protocol adds energy constraints to make sure that the energy spent in the network is also limited. In [53] and [12], the authors give several results on how to model the system lifetime of ad hoc networks when the total energy in the network is the limiting factor for the lifetime. In [12], the authors show how to modify the Bellman-Ford algorithm to achieve the maximum lifetime under the assumption of static energy levels at the time of transmission. [54] gives a routing protocol designed to maximize the lifetime of the network. The protocol implements load balancing for node communications by randomly assigning cluster head duties among the nodes at different time intervals. The randomization balances the amount of time each node spends as the main link to the rest of the network. [55] gives a protocol that works on top of the existing routing protocols to enhance the energy performance. The protocol dynamically changes the routes to select routes with lower energy cost. [56] gives a network layer protocol that chooses a set of nodes to form a backbone and powers off the nodes that are not essential to keep the network connected.

[57] gives a protocol that combines the physical and network layers to achieve minimum energy consumption in the network. First, the protocol discovers the cost of transmission on

a single hop of all the nodes in the network, and then it uses the Bellman-Ford algorithm to find the minimum energy connected network. [58] uses an approach similar to [57], however this paper uses directional information about incoming signals to calculate the minimum energy topology in the network.

Chapter 2

Mathematical Preliminaries

In this chapter, we give the mathematical preliminaries needed for the rest of the thesis. First, we will give a precise description of path diversification. Second, we give the assumptions made about the network and the measurement services provided by the network needed to perform the optimizations in Chapter 3. The assumptions are needed in order to have a model with which we perform optimizations P1-P4. We show how each of the assumptions has a reasonable explanation in ad hoc networks. Third, using this model we show how the exact network reliability can be calculated with a polynomial time algorithm and how the reliability can be approximated with the Poisson approximation. We use the properties of the algorithm and the approximation in Chapter 3 when we solve optimizations P1-P4.

2.1 Path Diversification

We propose a new mechanism, which uses a technique we call *path diversification* to achieve an improved packet loss statistics when there exists a large packet loss in the network. The mechanism splits the data packets into fragments when they arrive at the source node, and distributes the fragments on multiple parallel paths in the network. The packets are reassembled at the destination node. We have shown in Section 1.4 that in order to make

this mechanism efficient we need to use erasure codes to split the packets into fragments at the source node, and reconstruct them at the destination. Our mechanism uses the erasure code given in [20]. We call this mechanism path diversification because the fragment are sent to the destination node through separate paths in the network.

We now give a mathematical model of path diversification. The connection generates a packet of size bM that it wishes to transmit to the destination. The source node breaks up the packet into M fragments of size b , generates K fragments of parity and transmits the total of $M+K$ packets to the destination. The destination must receive at least M fragments in at most D_M time units, so that the transmission is considered to be “successful”, i.e. the packet is received in time.

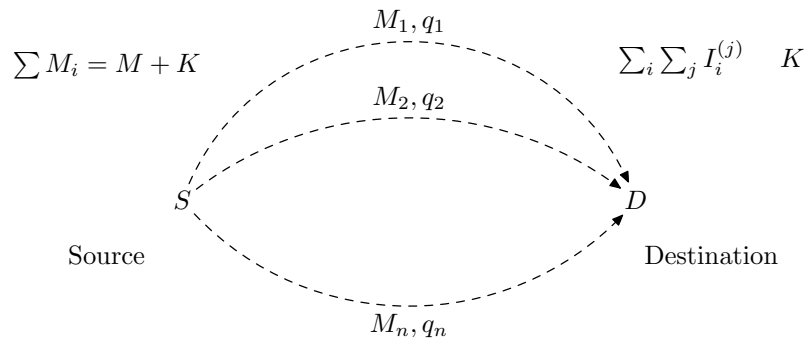


Figure 2.1: Mathematical Model of Path Diversification

The $M + K$ packets are subdivided into n non-overlapping sets with M_i packets in set i , where M_i packets are transmitted on path i . We will denote with $\mathbf{m} = [M_1, M_2, \dots, M_n]^T$ the allocation vector for the connection over the n paths in the network. We show this in Fig. 2.1. The probability that the packet can be reconstructed is the probability that more than M of the fragment transmissions succeed, or equivalently that less than K transmissions fail.

We can express network reliability or probability of success as:

$$P_{\text{succ}} = \Pr[\text{Number of Received Fragments} \geq M]. \quad (2.1)$$

The efficiency of path diversification is defined as:

$$\eta = \frac{\text{Effective Throughput}}{\text{Actual Throughput}} = \frac{M}{M + K}. \quad (2.2)$$

We ignore the overhead introduced by the rest of the network.

We will show in Section 3.1 that path diversification, without any assumptions about the network, improves the packet loss (network reliability). However, in the rest of Chapter 3, we will show how to improve the performance of path diversification when more information about the network is known. Next, we show how this information can be collected by probing the network.

2.2 Network Model

We model a wireless ad hoc network as a random graph with flawless vertices and failure-prone edges. We make the assumption of flawless vertices because it is possible to efficiently transform any directed random graph with failure-prone vertices into a graph with flawless vertices [59]. The conversion algorithm replaces each failure-prone vertex with an equally failure-prone edge and two flawless vertices. We assume that the edge failures are independent. This means that the movement of the nodes in the network is independent (this is a commonly made assumption [29]). We assume that we have n edge-disjoint [30] paths available for consideration between the source and destination nodes. We discussed in Section 1.6 that there are many multipath algorithms available for ad hoc networks which find the edge-disjoint paths even in a network with mobile nodes. A path is considered to be “failed” if a packet transmitted on the path does not reach the destination. Two edge-disjoint paths have no edges in common, so the failures of the two paths will be independent.

We assume that the network has a mechanism that allows us to collect statistical information, such as path failure statistics and energy information about the paths. For example,

we may use a probing technique similar to [17] or an egress probing technique such as [18]. The network measurements are updated every T_w seconds. Ideally, this period is less than or close to the network variation time. Next, we show how the delay, packet loss, and energy consumption in the network can be provided through probing and simple statistical collection at each node. The information is used to improve the performance of path diversification in optimizations P1-P4.

Delay the Network

We assume that through network measurements we have access to the maximum number of fragments we can transmit on a path in time period D_M , $M_i^{(d)}$. We will denote with $\mathbf{M}_d = [M_1^{(d)}, M_2^{(d)}, \dots, M_n^{(d)}]^T$ the vector of the packets we can transmit on n paths, as constrained by the delay in the network. We can get the maximum number of fragments in a variety of different ways. For example, we can use the measured service curve [17, 18] to find the number of packets that can be transmitted in certain time extent on the path. We explain the procedure for extracting $M_i^{(d)}$ from the service curve with the ϵ -effective service curve [17] even though we can do this with any other service curve. The ϵ -effective service curve for path i , S_i^ϵ , is defined as follows:

$$S_i^\epsilon(t) = l_i \leftrightarrow \Pr[\mathcal{D}_i(l_i) > t] \leq \epsilon \quad (2.3)$$

where l_i is the number of fragments of size b transmitted on path i and $\mathcal{D}_i(l_i)$ is the total delay experienced by all the packets during the transmission of l_i consecutive fragments on path i . The destination node ignores all packets that take more than $D_M \leq T_w$ seconds. So, we will incorporate the factor of ϵ into the measurement of probability of fragment loss and treat the service curve as a deterministic service curve [60], i.e. $M_i^{(d)} = S_i^0(D_M)$. The source nodes need the delay information in order to make sure the network is not overloaded and to make sure the loss due to congestion delay is kept at a reasonable level.

Packet Loss in the Network

We will denote by $p_i(j)$ the probability that fragment j of size b on path i may be lost or delayed for more than D_M seconds. The value of $p_i(j)$ is obtained through statistical analysis of network behavior every T_w seconds. We use $\mathcal{I}_i^{(j)}$ as the indicator random variable:

$$\begin{aligned} p_i(j) &= \Pr[\mathcal{I}_i^{(j)} = 1], \\ q_i(j) &= 1 - p_i(j) = \Pr[\mathcal{I}_i^{(j)} = 0]. \end{aligned} \tag{2.4}$$

The probability of packet loss can be measured at the destination node by keeping track of the number of packets lost on each of the paths. This means that the protocol used to fragment the packets and reassemble them at the destination node should include in each fragment header, the information about the number of fragments expected on each path, as well as framing information to index the fragments on each path. Alternatively, each of the nodes on the path can keep track of its own packet loss and the upper bound for path loss can be calculated by using the formula for reliability of serial structures [61].

The probability of packet loss on each path is a random process that is observed at the source node. If the source node had perfect information about fragment loss it would know the exact future probability of loss for each segment it is transmitting on the path and be able to allocate the fragments on each path to maximize the probability of successful packet transmission. In practice, the source node should predict the future behaviour on each path from the past samples. The prediction procedure is beyond the scope of this thesis. However, we assume that we do have access to the predicted values $\hat{p}_i(j)$ for $p_i(j)$ in (2.4). We assume that the prediction procedure approximates $\hat{p}_i(j)$ in a way that minimizes the error of approximating the dependent variables $\mathcal{I}_i^{(j)}$ as independent (for example [62] gives an estimate for $\hat{p}_i(j)$ that minimizes the maximum error). The approximation of $\mathcal{I}_i^{(j)}$ as independent allows us to efficiently and exactly calculate the upper bound on the reliability.

We also assume that as a part of the prediction process the source node has access to

the validity period of the observed samples for the fragment loss on each path, $M_i^{(p)}$. $M_i^{(p)}$ is the number of fragments that can be sent on the path and still be “reasonably” sure that the fragment loss approximates the true value of packet loss on the path. We denote $\mathbf{M}_p = [M_1^{(p)}, M_2^{(p)}, \dots, M_n^{(p)}]^T$.

In order to simplify the calculations in subsequent sections, we will assume that $p_i(j) = p_i$ and $q_i(j) = 1 - q_i$. This assumption will allow us to express the optimization problems P1-P4 as integer optimization problems with n variables. However, we can easily modify the integer programming optimizations, into $\{0, 1\}$ -integer optimizations with $M + K$ variables to take into account the differing values $p_i(j)$ on a single path.

Energy Consumption in the Network

We assume that every node in the network has the ability to measure the average amount of energy it uses to transmit a bit of information $e_i^{(k)}$ and the amount of available energy on the node $E_i^{(k)}$, where we indexed the node as “node k on path i ”.

With network probing we also have access to the per-bit energy required to transfer a packet between the source and the destination, $e_i^{(b)}$. The per-bit energy consumption on a path can be determined by adding up the energy required to transfer a bit at every node on the path:

$$e_i^{(b)} = \sum_{k=1}^{n_i} e_i^{(k)}. \quad (2.5)$$

where n_i is the number of nodes on path i . The vector of per bit energy consumption in the network is given by $\mathbf{E}_b = \{e_1^{(b)}, e_2^{(b)}, \dots, e_n^{(b)}\}$.

With network probing we also have access to the maximum number of fragments that can be transmitted on each path before the energy on the path runs out, $M_i^{(e)}$:

$$M_i^{(e)} = \min_{1 \leq k \leq n_i} \left\{ \frac{E_i^{(k)}}{be_i^{(k)}} \right\}. \quad (2.6)$$

We denote with \mathbf{M}_e the vector of maximum number of fragments we can transmit on each path $\mathbf{M}_e = [M_1^{(e)}, M_2^{(e)}, \dots, M_n^{(e)}]^T$.

Given this information about each path in the network we can calculate the maximum number of packets that can be transmitted on the paths:

$$\mathbf{M}_{th} = \min(\mathbf{M}_p, \mathbf{M}_e, \mathbf{M}_d) \quad (2.7)$$

where $\mathbf{M}_{th} = [M_1^{(th)}, M_2^{(th)}, \dots, M_n^{(th)}]^T$, and \min is the minimum of all the elements of the vectors with the same index. We will use the vector \mathbf{M}_{th} as the basic constraint on the number of packets that can be transmitted on each path in optimizations P1-P4.

Path Lifetime

The lifetime of each path can also be found through the probing mechanism by finding the minimum lifetime of all the nodes on the path:

$$T_i = \min_{1 \leq k \leq n_i} \left\{ \frac{D_M E_i^{(k)}}{b M_i e_i^{(k)}} \right\} = \frac{D_M}{b M_i} \min_{1 \leq k \leq n_i} \left\{ \frac{E_i^{(k)}}{e_i^{(k)}} \right\} = \frac{D_M}{M_i} M_i^{(e)} \quad (2.8)$$

where M_i is the number of fragments transmitted on the path and $M_i^{(e)}$ is defined in (2.6) as the maximum number of fragments that can be transmitted on the path due to energy constraints. We base (2.8) on the observation that the rate offered on the path is given by $R_i = b M_i / D_M$, and the lifetime of the path is given by the amount of time the source needs to use up all the energy on the path. We assume that the rate offered by the connection is given by:

$$R = \sum_{i=1}^n R_i = \frac{b}{D_M} (M + K), \quad (2.9)$$

and that there always packets backlogged at the source node. We also assume that the connection is guaranteed transmission of at least $M_i^{(e)}$ fragments on the path. For example,

each node k on the path can assign $E_i^{(k)}$ as the number of fragments this connection can use. This way we are guaranteed that the path lifetime in (2.8) will be exactly T_i , and not less because other connections may be using the energy on the same node.

2.3 Network Reliability P_{succ}

In this section we show how to calculate the exact value for network reliability, P_{succ} , and how to approximate P_{succ} with the Poisson approximation. Network reliability is important for the operation of upper layer protocols. In order to motivate the need for enhanced network reliability we show the performance of long term throughput of TCP as the reliability in the network changes in Fig. 2.2. We have used the approximate TCP performance model of [63]. The reliability in Fig. 2.2 is the probability of successful packet transmission. [63] shows with empirical data that this model of long-term TCP throughput is fairly good, the discussion of the model is beyond the scope of this thesis. However, it is worth noting that [63] shows that it is quite accurate for high values of reliability. We use the figure to point out that the increase in reliability increases the long term throughput of TCP exponentially, so network reliability is the most important factor in the design of protocols for ad hoc networks. For example, an increase of reliability of $0.95 \rightarrow 0.999$ (an increase of about 5%), increases the throughput more than 5 times (400%).

Given the model of Section 2.2 and the rules for path diversification we express the network reliability as follows:

$$P_{\text{succ}} = \Pr[\mathcal{W} \leq K] = \Pr\left[\sum_{i=1}^n \sum_{j=1}^{M_i} \mathcal{I}_i^{(j)} \leq K\right] \quad (2.10)$$

where \mathcal{W} is a random variable indicating the total number of lost packets, and $\mathcal{I}_i^{(j)}$ are indicator variables showing wheather packet transmission j on path i was successful.

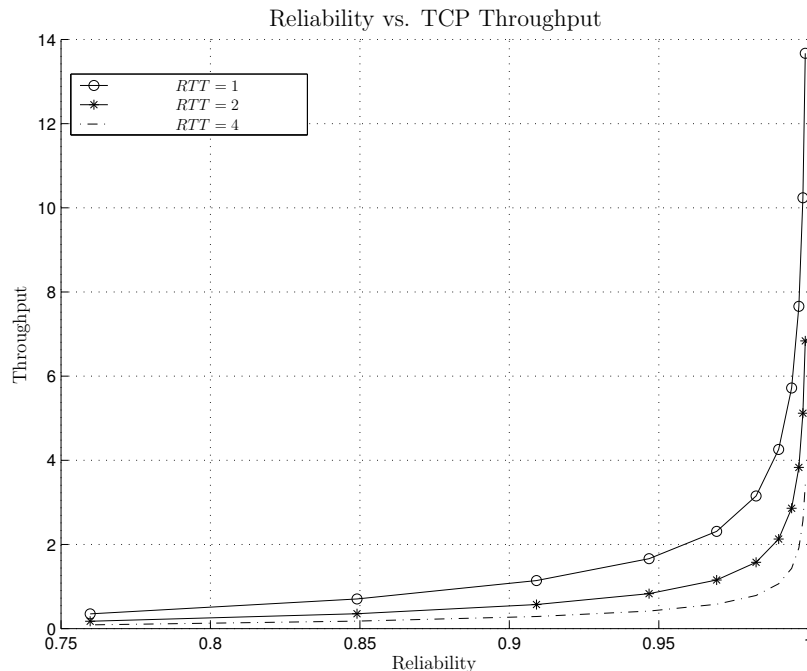


Figure 2.2: Long Term TCP Throughput vs. the Reliability

2.3.1 Exact P_{succ}

We give a polynomial time algorithm for the exact calculation of P_{succ} , we will use the results from the derivation of this algorithm later in Section 3.2 as a fundamental result for the implementation of optimizations P1 and P2. There are also two approximations for P_{succ} used elsewhere in the literature, the Normal approximation [64, 23] and the Poisson approximation [62, 65, 66]. We will compare the exact results with the results for the two approximations in Section 3.2.

We can show that the calculation of the exact value for P_{succ} is directly related to the calculation of the reliability of algebraic structures, which is in general NP-Complete. There are many works available on the issue of reliability of algebraic structures [61, 67, 59, 68]. In this work, we will use the special properties of k -out-of- n structures to calculate the exact probability of successful packet transmission and to optimize the transmission of packets on a path. [69] gives an algorithm for the direct calculation of the probability of failure of

Algorithm 1 EVALUATE-PSUCC($\mathbf{m}, \mathbf{q}, K$)

```

1:  $k \leftarrow 1, \forall j : c_j \leftarrow 0, c_1 \leftarrow 1$ 
2:  $P_{\text{succ}} \leftarrow 0, n \leftarrow \text{length}(\mathbf{m})$ 
3: for  $i \leftarrow 1$  to  $n$  do
4:   for  $l \leftarrow 1$  to  $M_i$  do
5:     for  $j \leftarrow \min\{K, k + 1\}$  to 2 do
6:        $c_j \leftarrow q_i c_j + p_j c_{j-1}$ 
7:     end for
8:      $c_1 \leftarrow q_i c_1$ 
9:      $k \leftarrow k + 1$ 
10:  end for
11: end for
12: for  $i \leftarrow 1$  to  $K$  do
13:    $P_{\text{succ}} \leftarrow P_{\text{succ}} + c_i$ 
14: end for
15: return  $P_{\text{succ}}$ 

```

k -out-of- n structures with independent component reliabilities. In this section, we show how that algorithm can be modified to find P_{succ} in polynomial time.

The algorithm is based on the use of the moment generating function for the sum of $M + K$ independent Bernoulli variables. We showed in Section 2.2 how to model the packet transmissions with independent Bernoulli random variables $\mathcal{I}_i^{(j)}$. By (2.10), the moment generating function for the number of lost packets, \mathcal{W} , is given by:

$$G_{\mathcal{W}}(z) = \prod_{i=1}^n (q_i + p_i z)^{M_i} = \sum_{i=0}^{M+K} c_i z^i. \quad (2.11)$$

So, we can calculate P_{succ} by using the moment generating function as follows:

$$P_{\text{succ}} = \Pr[\mathcal{W} \leq K] = \sum_{i=0}^K c_i \quad (2.12)$$

where each c_i is given by (2.11). We use the multiplication technique used in [69], where we perform $M + K$ iterations, and at each iteration k we recalculate the coefficients to get $c_j^{(k)}$.

We use the following recursive relation to find the coefficients at each iteration:

$$c_j^{(k+1)} = q_{k+1}c_j^{(k)} + p_{k+1}c_{j-1}^{(k)}. \quad (2.13)$$

Using this relation, we derive Algorithm 1. The algorithm uses the same multiplication technique used in [69], where we perform $M + K$ iterations, and at each iteration k the algorithm recalculates the coefficients to a set of coefficients $c_j^{(k)}$ with the use of (2.13). After the $M + K$ iterations the first K coefficients are added to obtain the value for P_{succ} .

Theorem 1 (Correctness Of Algorithm 1). *Algorithm 1 gives the exact value for P_{succ} .*

Proof. First we show that inner loop and the outer loop calculate coefficients c_i for the distribution of \mathcal{W} . We use $\mathcal{W}^{(k)}$ to denote the random variable indicating the number of successes in k trials. Assume that $M_1^{(k)}$ trials have probability probability of q_1 of failure, $M_2^{(k)}$ probability of q_2 of failure, and so on, and $\sum_{i=1}^n M_i^{(k)} = k$. By induction:

1. For $M + K = 1$ we get the coefficients of $c_2 = p_1$, $c_1 = q_1$. For a single trial \mathcal{W} is a Bernoulli random variable, so the moment generating function is correct.
2. We assume that $g_{\mathcal{W}^{(k)}}(z)$, the moment generating function for $\mathcal{W}^{(k)}$, is correct. We need to show that $g_{\mathcal{W}^{(k+1)}}(z)$ is correct.

For $M + K > k + 1$ assume for some k that coefficients c_i are correct for $g_{\mathcal{W}^{(k)}}(z)$ which is the moment generating function for number of failures when there are k trials. We need to show that $\mathcal{W}^{(k+1)}$ is correct. First we note that $g_{\mathcal{W}^{(k+1)}}(z)$ is given by:

$$g_{\mathcal{W}^{(k+1)}}(z) = (q_i z + p_i)g_{\mathcal{W}^{(k)}}(z). \quad (2.14)$$

Where we assume that the loop has up to reached M_i , $i < n$ trials. So, for every coefficient c_j^{k+1} (2.13) is indeed true, and the inner loop does calculate this recurrence.

The $\min(K, k + 1)$ in the inner loop does not change the value of P_{succ} since we only add up the coefficients c_i up to K . So, if P_{succ} for k is true, so must be P_{succ} for $k + 1$.

This proves the coefficients c_j are calculate correctly.

We also note that the algorithm adds up the first K coefficients, as shown by (2.13). This proves the theorem. \square

Theorem 2 (Complexity Of Algorithm 1). *The complexity of the Algorithm 1 is $\Theta(K(M + K))$.*

Proof. We assume that the addition and multiplication are equally difficult to perform. We note that the two outer loops execute the inner loop $M + K$ times. So, the total time for the algorithm to calculate P_{succ} for $M + K$ total components is given by:

$$\begin{aligned}
T(M + K) &= \sum_{k=1}^{M+K} \left[\sum_{j=2}^{\min(K, k+1)} 3 + 2 \right] + K \\
&= \sum_{k=1}^{K-1} \sum_{j=2}^{k+1} 3 + \sum_{k=K}^{M+K} \sum_{j=2}^K 3 + \sum_{i=1}^{M+K} 2 + K \\
&= 3 \sum_{k=1}^{K-1} k + 3M(K - 1) + 2(M + K) + K \\
&= \frac{3}{2}K^2 + 3MK + 2M + \frac{3}{2}K - 3
\end{aligned} \tag{2.15}$$

We note that:

$$\frac{3}{2}K^2 + \frac{3}{2}K - 3 = \Theta(K^2) \quad \text{and} \quad 3MK + 2M = \Theta(MK) \tag{2.16}$$

So,

$$\frac{3}{2}K^2 + 3MK + 2M + \frac{3}{2}K - 3 = \Theta(K^2) + \Theta(MK) = \Theta(K^2 + MK) \tag{2.17}$$

which proves the theorem. \square

Algorithm 1 is more efficient than the algorithm in [69] which if it was just rewritten to calculate P_{succ} would have the complexity of $\Theta((M + K)^2)$.

2.3.2 Poisson Approximation of P_{succ}

We now show how the network reliability can also be approximated with Poisson approximation for the sum of independent Bernoulli random variables. We will use this approximation in later section to simplify optimization P3. However we will also show that the Poisson approximation can be used effectively for other optimizations. The Poisson approximation was considered in [65]. In that paper, the author also examines the reliability of k -out-of- n structures for independent components. P_{succ} was bounded with the Poisson distribution as:

$$Q(\mathbf{m}, K) \leq P_{\text{succ}} \leq Q(\mathbf{m}, K) + \frac{1}{2} \sum_{i=1}^M M_i \ln^2(q_i) \quad (2.18)$$

where,

$$Q(\mathbf{m}, K) = \sum_{j=0}^K \frac{e^{-\lambda(\mathbf{m})} [\lambda(\mathbf{m})]^j}{j!} \quad (2.19)$$

and,

$$\lambda(\mathbf{m}) = \sum_{i=1}^n \ln(q_i^{-M_i}) = -\mathbf{m}^T \ln(\mathbf{q}) \quad (2.20)$$

and $\ln(\mathbf{q}) = [\ln(q_1), \ln(q_2), \dots, \ln(q_n)]^T$ is the natural logarithm of the vector of probabilities of success. [65] shows the tightness of the approximation, and that it is appropriate to use $Q(\mathbf{M}, K)$ for the purpose of approximating the probability of success in k -out-of- n algebraic structures, which is equivalent to the probability of successful reception of M out of $M + K$ packets.

The Poisson Approximation is a good candidate to replace the Algorithm 1, since very good values of P_{succ} can be obtained in relatively few steps. For example, with just 10 iterations we can calculate $Q(\lambda(\mathbf{m}), K)$ with precision of less than 10^{-6} (note $10! > 3 \times 10^6$).

Chapter 3

Optimum Resource Optimization

In this chapter, we study the problem of path diversification in two cases. First, we assume that the source does not have any knowledge about the status of the parallel paths. In this case, the source distributes the segments uniformly over the paths. We refer to this technique as the *blind path diversification*. We discuss the performance of blind path diversification in Section 3.1 and show that it outperforms the single-path transmission. Second, we assume that some performance metrics of the parallel paths are available at the source (we have shown how to obtain these performance metrics about the paths in Section 2.2). In this case, the source can use one of optimal approaches P1-P4 to distribute the segments over parallel paths, every time it has new information about the paths. We show how to perform the optimizations P1-P4 in Sections 3.2-3.5. Each optimization results in the allocation vector $\mathbf{m} = [M_1, M_2, \dots, M_n]^T$ and the number of parity packets K , needed to satisfy the QoS constraints and minimize or maximize the objective function.

3.1 Blind Path Diversification

In this section, we examine the performance of blind path diversification. In blind path diversification, no knowledge of the performance of parallel paths is available at the source.

Assume that there are M segments offered by the connection and K parity segments generated by the source node. The source node distributes the segments between the n available paths. In blind path diversification, the source deploys $\frac{M+K}{n}$ segments on each path; for simplicity, we assume that $\frac{M+K}{n}$ is an integer number. We argue that blind path diversification has a higher probability of success than the single-path transmission for large values of K .

We model the wireless channel as a Markov chain with multiple states. Each state corresponds to a certain probability of fragment loss. This is a commonly used model in wireless networks [70] since this channel is statistically similar to the actual channel in wireless networks. We show a sample model with 4 states in Fig. 3.1. Good channel corresponds to a low value of fragment loss, $p_i(3)$, and the bad channel corresponds a high value of fragment loss $p_i(0)$. If the channel has only two states it corresponds to the Gilbert-Elliott model [71]. The Gilbert-Elliott channel is commonly used to model very bursty connections, and can be used for modelling wireless connections in ad hoc networks [72]. An alternative model uses multiple states and has been proposed in [73] to replace the Gilbert-Elliott model in wireless networks. In both models, there are GOOD and BAD states, corresponding to good and bad channels. We study the blind path diversification in two extreme cases using these models.

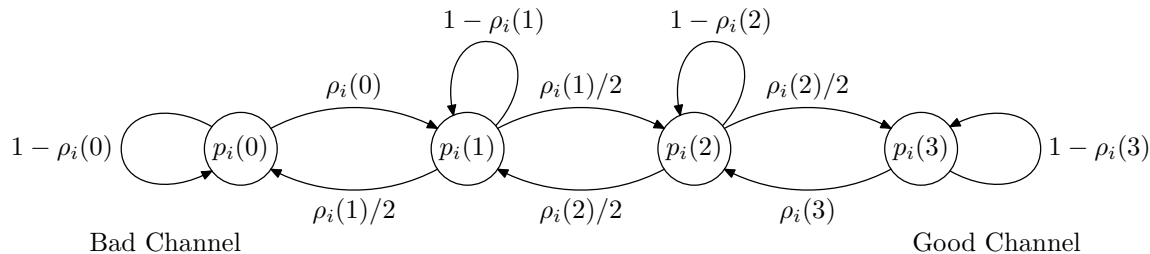


Figure 3.1: Markov Chain Channel Model

In the first case, we assume that K is very small compared to M . This case corresponds to a high value of efficiency. In a multiple-path transmission, since K is small we need most of the segments from each path to arrive at the destination in order to reconstruct the

original packet. For a channel that is in one of the BAD states, most of the transmitted segments will be lost. So, we will need to have all parallel paths to be in GOOD state simultaneously. Since the paths move independently among the states, the probability of this event will be small. On the contrary, for a single-path transmission, when the channel is in GOOD state, most of the segments will succeed. Since the probability of having a GOOD state along a single path is higher than the probability of having simultaneous GOOD state across multiple paths, we can conclude that single path transmission will result in better performance. Note that in this case, since K is small, the erasure coding is not effective.

In the second case, we assume that K is large compared to M . This case corresponds to a low value of efficiency. For large values of K , the packet can be reconstructed even if few of the channels are in GOOD state. Note that the probability of having a single channel in the BAD state is more than that of having multiple channels simultaneously in the BAD state. So, we can expect that for higher K , multipath transmission will perform better.

We perform Monte-Carlo simulations in which we transmit $N = 5000$ packets over a Gilbert-Elliott channel. The channel has a GOOD and a BAD state. Packets are much more likely to be lost in the BAD state than in the GOOD state. In GOOD state, $q_i = 0.7$, and in BAD state $q_i = 0.3$. The transitions from the GOOD state to the BAD state form a Markov Chain. For simplicity, we assume the channel only changes states between the transmissions of packets. We distribute the fragments uniformly over the n paths. In Fig. 3.2, we show the likelihood of packet recovery as the number of paths increases. It is clear that for large values of K , the performance of the erasure code increases with the number of paths.

Note that in practice, we would like to have a high probability of success. Therefore, we should operate the network in a region that corresponds to P_{succ} higher than a threshold. On the other hand, we would like to have an acceptable level of efficiency. These two constraints considered together will need a minimum number of parallel paths in the network. For instance, in Fig. 3.2, for $P_{\text{succ}} \geq 0.9$, and $\eta \leq 0.42$, we will need at least $n = 10$ paths. In

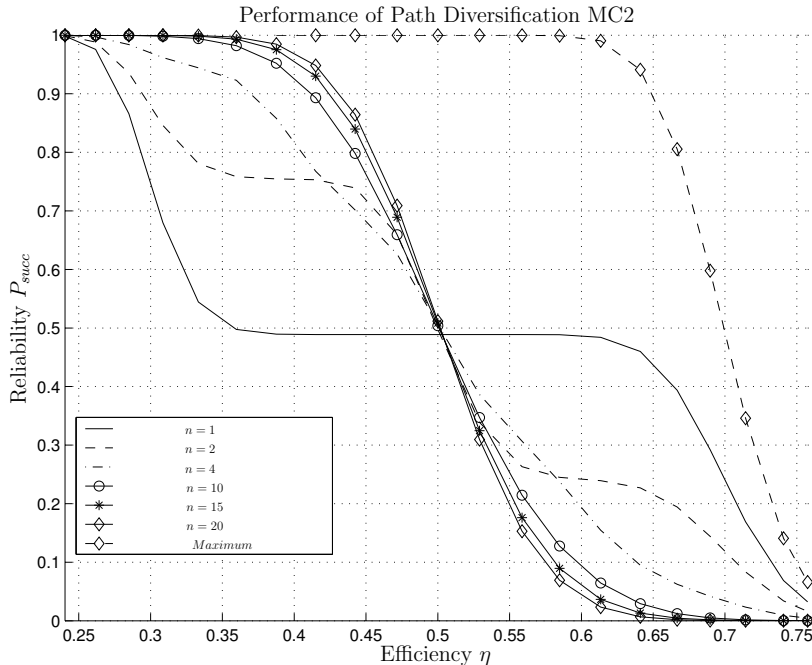


Figure 3.2: Performance of Blind Diversification for Gilbert-Elliott Channel

this figure, we also show a case in which at each time instant all segments are transmitted on the best path. This case has been denoted by “Maximum” in the graph. Note that sending all segments on the best path substantially increases the performance.

We also show the performance of multipath diversity in Fig. 3.3 for a five-state Markov Chain channel. The figure shows the average reliability that can be achieved with blind path diversification. In this model, the channel changes gradually between the GOOD and BAD states, and the states form a Markov Chain with $\mathbf{q} = [0.3, 0.4, 0.5, 0.6, 0.7]^T$. We also assume that the channel only changes state between 3. packet transmissions. Note that for this channel, we will also need to have multiple paths to achieve both high probability of success and acceptable efficiency.

We now show how blind path diversification performs when it is used with the selective-repeat ARQ protocol. selective-repeat ARQ protocol can be used to approximate the efficiency of TCP since only the lost packets are retransmitted in both protocols. We show in

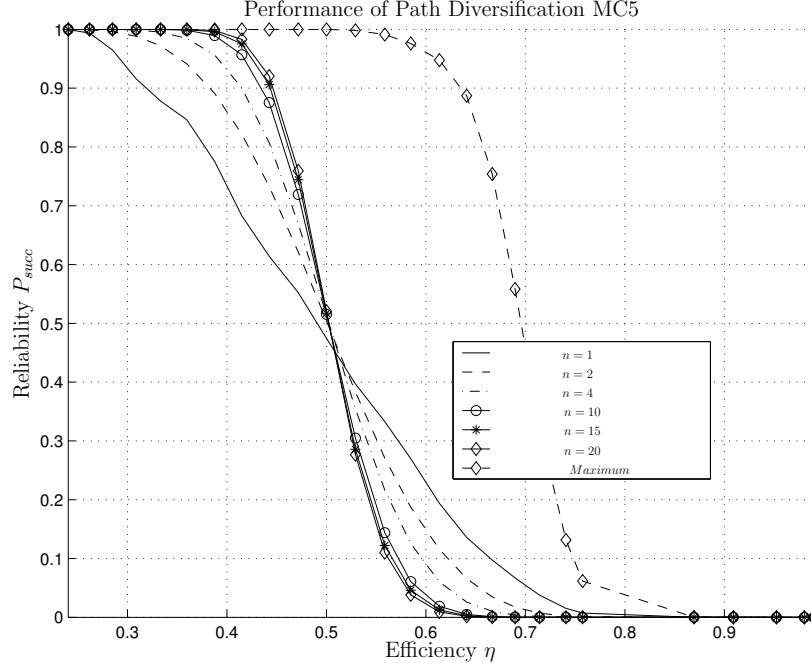


Figure 3.3: Performance of Blind Diversification for Markov-Chain Channel

Appendix B that the efficiency of selective-repeat ARQ approximated as:

$$\eta_{SR} = \bar{q} \quad (3.1)$$

where \bar{q} is the average packet loss in the network. We use the same approximation as [74] to derive this value for the efficiency. We also show in Appendix B that the efficiency of selective-repeat ARQ in path diversified networks can be approximated as:

$$\eta_{SR-D} = \epsilon \frac{M}{M+K} = \epsilon \eta \quad (3.2)$$

where η is the efficiency of path diversification and ϵ is the minimum reliability in the network. We show the plot of the minimum reliability we can achieve with blind path diversification in Fig. 3.4.

We show the efficiency we can expect from selective-repeat ARQ in Fig. 3.5. We can see

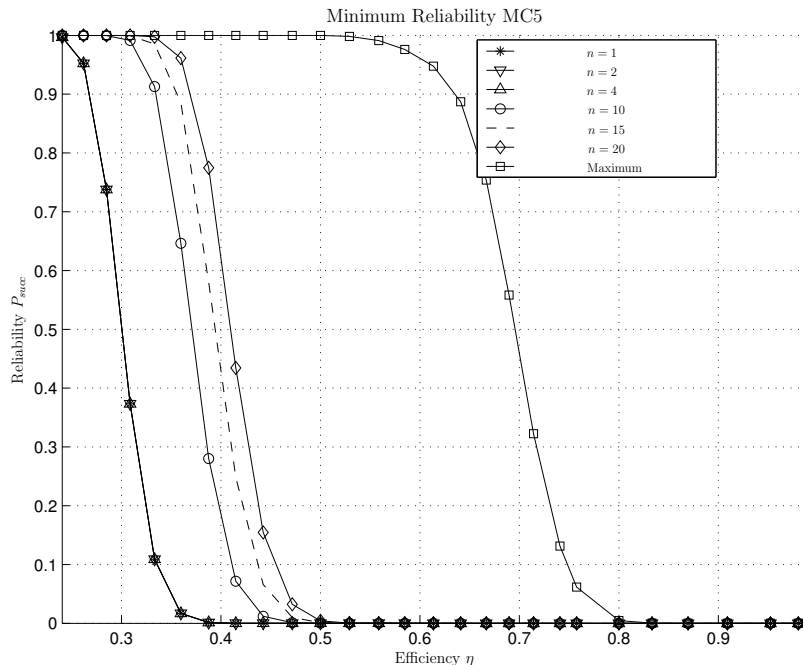


Figure 3.4: Minimum Reliability of Blind Diversification for the 5-state Markov-Chain Channel

from the figure that blind path diversification cannot match the efficiency of selective-repeat ARQ. However, it is important to note that the efficiency as defined in this thesis relates to the overhead introduced to transmit the data stream correctly, and not to the decrease in throughput due to the use of the scheme. In fact, we have shown in Section 2.3 that due to the limitations of TCP we can expect a significant increase of TCP throughput with blind path diversification, since we can make the minimum reliability arbitrarily high as shown in Fig. 3.4. Nevertheless, Fig. 3.5 shows that we may be able to achieve efficiency better than the selective-repeat ARQ efficiency if we optimize the way packets are distributed.

We show how the performance of “blind” diversification can be improved in the next chapter. So far, we have assumed that no knowledge of the performance along the parallel paths is available at the source node. We saw in Fig. 3.2 and Fig. 3.3 that with this approach it is not possible to increase the efficiency beyond 40% for a large value of reliability. We saw in both figures that if we could find the path with the maximum q_i and transmit all segments

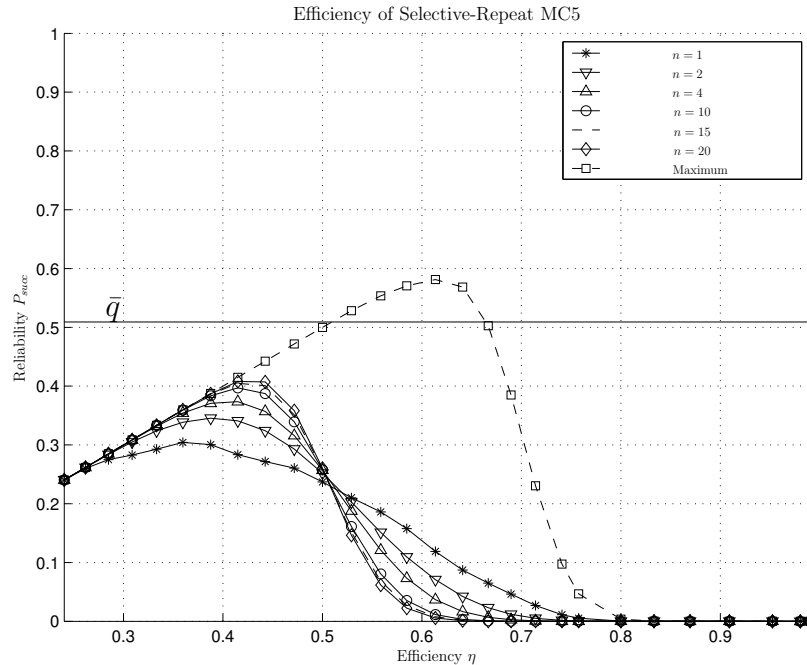


Figure 3.5: Efficiency of Selective-repeat ARQ with Blind Diversification

along that path, we could have had a much higher efficiency. In the rest of the thesis, we will assume that some performance metrics are available at the source node and will try to use these metrics to devise optimum schemes for the distribution of segments along parallel paths.

3.2 Maximization of P_{succ} (P1)

In this section, we modify blind path diversification to maximize the network reliability. We have shown in Section 2.3 the improvement of long term TCP throughput with the increase network reliability. So, this optimization is very important to make ad hoc networks accessible to regular traffic. We defined this maximization as problem P1 in Section 1.5 and we defined network reliability as the probability of successful packet transmission in Section 1.2.1. We will use the results of this section in later sections when we optimize the efficiency, energy use, and network lifetime.

Optimization P1 minimizes the number of retransmitted packets in the network, by maximizing P_{succ} the probability of successful packet transmissions. We solve the optimization with a polynomial-time “greedy” algorithm. The algorithm also uses constraints on the paths to implement load balancing in the network.

Formally, we define optimization P1 as the following mathematical program:

$$\begin{aligned} & \text{Maximize: } P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) \\ & \text{Subject to: } \mathbf{m}^T \mathbf{1} = K + M \\ & \quad 0 \preceq \mathbf{m} \preceq \mathbf{M}_{th} \end{aligned} \tag{3.3}$$

where $\mathbf{1}$ is a vector of 1s, and \preceq indicates a member-wise comparison. We assume that the source node has access to the information about path reliability, and the maximum number of packets that can be sent on each path.

We first note that P_{succ} is a monotonically increasing function of K . This is clear from (2.10) since P_{succ} is defined as the cumulative distribution function (c.d.f.) of the random variable \mathcal{W} . We will assume that K is fixed since P_{succ} can always be increased by increasing K . The optimization of P_{succ} is constrained by the maximum number of packets that can be transmitted on each path. The thresholds on each path are chosen with the use of (2.7) as shown in Section 2.2.

Intuitively, the probability of success should be maximized when the fragments are transmitted on the paths with the highest probability of successful fragment transmission, q_i . In an ideal situation, the source node would have the perfect information about each of the paths in the network and it would always be able to transmit each fragment on the path with the highest value of q_i . However, the source will never have the perfect information about each of the paths. We approximate this by fixing q_i on each path for some period $M_i^{(p)}$, which means we expect the probability of fragment loss to be constant for the next $M_i^{(p)}$ transmissions. We also limit the number of fragments that can be transmitted on each

Algorithm 2 MAXIMIZE-PSUCC($\mathbf{q}, K, \mathbf{M}_{th}$)

Require: $\forall i > j, q_i \geq q_j$ **Ensure:** $\mathbf{m}^T \mathbf{1} = M + K$ $i \leftarrow 1, M_{\text{total}} \leftarrow 0, \forall i, \mathbf{m} \leftarrow \mathbf{0}$ 2: **while** $i \leq n$ and $M_{\text{total}} \neq M + K$ **do** $\mathbf{m}(i) \leftarrow \max(M_i^{\text{th}}, M + K - M_{\text{total}})$ 4: $M_{\text{total}} \leftarrow M_{\text{total}} + \mathbf{m}(i)$ $i \leftarrow i + 1$ 6: **end while****if** $M_{\text{total}} \neq M + K$ **then**8: $\mathbf{m} = \emptyset$ **end if**10: **return** \mathbf{m}

path with QoS constraints (2.7), that includes $M_i^{(p)}$, as $M_i^{(\text{th})}$.

Suppose the paths are sorted so that the increase in the index means there is a higher probability of fragments loss, p_i , on the path with higher i . We can find the optimal solution for optimization (3.3) with a “greedy” algorithm. The algorithm assigns the maximum number of fragments, $M_1^{(\text{th})}$, to the path with the lowest probability of failure (path 1), and then the maximum number of fragments to the path with the next lowest probability of failure, and so on, until all the packets are assigned to some path. For the special case of $\mathbf{M}_{th} = \infty$ the optimization becomes resource-unconstrained, similar to the optimization in [23], and all the fragments should be assigned to the first path. Formally, we perform the optimization with the use of Algorithm 2. We show the correctness of the algorithm in Theorem 3 by using the recursive relation (2.13). We show the complexity of the algorithm is $O(n)$ in Theorem 4.

Theorem 3 (Correctness Of Algorithm 2). *Algorithm 2 gives a packet allocation on the paths in the network that maximizes P_{succ} .*

Proof. Suppose that after allocating l packets, $M_i < M_i^{(\text{th})}$ and $M_j < M_j^{(\text{th})}$, $M_j^{(\text{th})} > 0$. Let us assign the packet $l + 1$ to path j , even though $p_j > p_i$. We assume, without loss of

generality, the case where $K \leq l$. At step $l + 1$ of the evaluation of P_{succ} , we have by (2.13):

$$\begin{aligned}
P_{\text{succ}}^{(l+1)}(\mathbf{q}, K) &= \sum_{j=1}^K c_j^{(l+1)} = q_{l+1} \sum_{j=1}^K c_j^{(l)} + p_{l+1} \sum_{j=2}^K c_{j-1}^{(l)} \\
&= q_{l+1} c_K^{(l)} + [q_{l+1} + p_{l+1}] \sum_{j=1}^{K-1} c_j^{(l)} \\
&= q_{l+1} c_K^{(l)} - c_K^{(l)} + \sum_{j=1}^K c_j^{(l)} \\
&= -p_{l+1} c_K^{(l)} + P_{\text{succ}}^{(l)}(\mathbf{q}, K)
\end{aligned} \tag{3.4}$$

Clearly, P_{succ} is going to be greater if $p_{l+1} = p_j$, rather than if $p_{l+1} = p_i$. So, we should assign M_i^{th} packets to path i .

The proof is based on the case where $p_j > p_i$. In the case where $M_i < M_i^{(\text{th})}$ and $M_j < M_j^{(\text{th})}$, $M_j^{(\text{th})} > 0$ and $p_j = p_i$ assigning the packet to either path results in an equally optimal solution which proves the theorem. □

Theorem 4 (Complexity Of Algorithm 2). *Complexity of Algorithm 2 is $O(n)$.*

Proof. We assume that all operations take an equal amount of time. We note that the worst case performance of the algorithm is given by $T_{\text{worst}}(n) = 4n$ when $\forall j : 1 \leq j \leq n - 1 \sum_{i=1}^j M_i^{\text{th}} < M + K$. Clearly, this means that the algorithm is $O(n)$. □

We can also solve the $\{0, 1\}$ -integer program with this technique. We arrive at the $\{0, 1\}$ -integer problem if the probabilities of fragment loss are not uniform on each path. For example, the probability of fragment loss may increase with consecutive fragment transmissions. We can use the method of Algorithm 2 with minor modifications. We consider every possible packet transmission and the fragment loss for this transmission, there are a total of $\sum_{i=1}^n M_i^{\text{th}}$ possible slots, for each fragment that can be sent on every path. We send the fragments in $M + K$ slots which have the highest probability of success. The complexity of

the algorithm changes to $\Theta(M + K)$. We omit the proof of this claim.

We now compare the optimal solution of Algorithm 2 with the optimizations that use the Poisson and the Normal Approximation of P_{succ} .

3.2.1 Optimization of P_{succ} through the Poisson Approximation

We showed in Section 2.3 how the P_{succ} can be calculated with the Poisson approximation for the sum of independent Bernoulli random variables $Q(\mathbf{m}, K)$ with (2.18). If we use the Poisson approximation for P_{succ} , optimization (3.3) becomes:

$$\begin{aligned} \text{Maximize: } Q(\mathbf{m}, K) &= \sum_{j=0}^K \frac{e^{-\lambda(\mathbf{m})} [\lambda(\mathbf{m})]^j}{j!} \\ \text{Subject to: } \mathbf{m}^T \mathbf{1} &= K + M \\ 0 &\preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \end{aligned} \tag{3.5}$$

and $\lambda(\mathbf{m})$ is given by (2.20).

We can perform the optimization P1 this way since $Q(\mathbf{m}, K)$ gives us the lower bound on P_{succ} . The lower bound guarantees that the solution of (3.5) will always be lower than the actual solution of (3.3). So, we can use the optimal solution of (3.5) as the sufficient condition to guarantee that the actual reliability will be higher than some threshold ϵ , even if the allocation itself is suboptimal.

We now show two methods to optimize the (3.5), first with optimize the problem with Convex optimization and second we optimiza the problem with an exact solution. We show that the exact solution of (3.5) results in Algorithm 2.

Convex Optimization with the Poisson Approximation

In order to optimize (3.5) with Convex optimization we need to change the constraint set to include only the allocation vectors for which $Q(\mathbf{m}, K)$ is concave. We show in Appendix A

how to make $Q(\mathbf{m}, K)$ a concave function for a fixed K , by adding the constraint:

$$K - \lambda(\mathbf{m}) \geq 0 \quad (3.6)$$

to the feasible set of \mathbf{m} . So, optimization (3.5) becomes:

$$\begin{aligned} \text{Maximize: } Q(\mathbf{m}, K) &= \sum_{j=0}^K \frac{e^{-\lambda(\mathbf{m})} [\lambda(\mathbf{m})]^j}{j!} \\ \text{Subject to: } \mathbf{m}^T \mathbf{1} &= K + M \\ K - \lambda(\mathbf{m}) &\geq 0 \\ 0 \preceq \mathbf{m} &\preceq \mathbf{M}_{\text{th}} \end{aligned} \quad (3.7)$$

Adding the constraint allows us optimize the the objective function $Q(\mathbf{m}, K)$ with the use of Convex programming [75]. Unfortunately, this convex optimization results in low values of efficiency since the new constraint limits the feasible set for which we can use convex optimization. We give an argument to show that point. Suppose, that the the mean of $\ln(\mathbf{q})$ is given by \bar{q}_{\ln} . We can then the mean of $\lambda(\mathbf{m})$ is given by:

$$\mathbf{E}\lambda(\mathbf{m}) = - \sum_{i=1}^n M_i \mathbf{E} \ln(q_i) = -\bar{q}_{\ln} \sum_{i=1}^n M_i = -\bar{q}_{\ln}(M + K) \quad (3.8)$$

So, we can use (3.6) to give find the expected value of efficiency for which we can solve (3.7):

$$K \geq \lambda(\mathbf{m}) \approx -\bar{q}_{\ln}(M + K) \leftrightarrow \frac{K}{M + K} \geq -\bar{q}_{\ln}(M + K) \leftrightarrow \eta \leq 1 + \bar{q}_{\ln} \quad (3.9)$$

However, in cases where the probability of P_{succ} needs to be large (so that η is relatively small), the approximate optimization may be more efficient to perform if the number paths in the network n is large.

Exact Calculation with the Poisson Approximation

We now show that the maximization through the Poisson approximation can be solved exactly by using the properties of the $Q(\mathbf{m}, K)$ function. We show in Appendix A that:

$$\frac{\partial Q}{\partial M_i} = -\frac{\partial \lambda}{\partial M_i} e^{-\lambda(\mathbf{m})} \frac{[\lambda(\mathbf{m})]^K}{K!} < 0 \quad (3.10)$$

which means that $Q(\mathbf{m}, K)$ is a decreasing function of $\lambda(\mathbf{m})$ for a fixed K . This means that we can maximize the probability of success by minimizing $\lambda(\mathbf{m})$. So, the optimization (3.5) becomes:

$$\begin{aligned} \text{Minimize: } & \lambda(\mathbf{m}) = \mathbf{m}^T \ln(\mathbf{q}) \\ \text{Subject to: } & \mathbf{m}^T \mathbf{1} = K + M \\ & 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \end{aligned} \quad (3.11)$$

Optimization (3.11) can be solved exactly for $\mathbf{m} \in \mathbb{R}^n$ with the use of Algorithm 2, since the sort of $\ln(\mathbf{q})$ and \mathbf{q} result in the same ordering of paths in the network. So, the Poisson approximation of P_{succ} and the exact optimization of P_{succ} yield the same resource allocation.

3.2.2 Optimization of P_{succ} through the Normal Approximation

The assumption of independent fragment transmissions was also used in [23], [64]. In that work, the optimization of P_{succ} was performed with the use of the Central Limit Theorem with the Normal approximation:

$$P_{\text{succ}}(\mathbf{m}, K) \approx 1 - \text{erf} \left(\frac{K - \sum_{i=1}^n q_i M_i}{\sqrt{\sum_{i=1}^n p_i q_i M_i^2}} \right) \quad (3.12)$$

The optimization of $P_{\text{succ}}(\mathbf{m}, K)$ becomes the maximization:

$$\begin{aligned} \text{Maximize: } & \frac{\sum_{i=1}^n q_i M_i - K}{\sqrt{\sum_{i=1}^n p_i q_i M_i^2}} \\ \text{Subject to: } & \mathbf{m}^T \mathbf{1} = K + M \\ & 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \end{aligned} \tag{3.13}$$

This approximation results in a suboptimal solution to the problem, since the Normal approximation is not a very good way to approximate P_{succ} , as will be shown later in Chapter 4.

There are two ways to perform the optimization (3.13). First, the approximation has a convex programming solution if the fractional programming theory is used [76]. Second, the authors of [23,64] chose to perform the optimization with a custom algorithm. The algorithm results in a packet allocation that is suboptimal. [23] also does not include a bound on the precision of the Normal approximation, for the case of unequal packet success probabilities. We will compare the exact results with [23] through simulations in Chapter 4.

3.3 Maximization of Efficiency (P2)

In this section, we solve problem P2, where we maximize the efficiency of path diversification. The efficiency η is a good way to measure the effectiveness of path diversification. For example, the overhead of the transport layer is directly related to the energy consumption in the network. We defined the efficiency as the overhead in the system η in Section 2.1. Once the efficiency of path diversification is maximized we see the improvement of the whole system including the upper layer protocols such as TCP.

Optimization P2 finds the maximum efficiency η for which path diversification can provide a requested level of network reliability, and the packet allocation scheme for which network reliability is satisfied. We optimize the efficiency η with a polynomial time algorithm based on the principles used in Algorithm 2.

Formally, we define optimization P2 as the following mathematical program:

$$\begin{aligned}
& \text{Maximize: } \eta(\mathbf{m}, K) = \frac{M}{M + K} \\
& \text{Subject to: } P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) \geq \epsilon \\
& \mathbf{m}^T \mathbf{1} = M + K \\
& 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}}
\end{aligned} \tag{3.14}$$

The optimization of η on its own is trivial (i.e. η is maximized when $K = 0$), so we perform the optimization with the constraint of minimum network reliability $P_{\text{succ}} \geq \epsilon$ where ϵ is a QoS parameter supplied by the connection.

We first note that $\eta(\mathbf{m}, K)$ is a monotonically decreasing function of K since M is constant. So, $\eta(\mathbf{m}, K)$ is maximized when K is minimized and the equivalent optimization is:

$$\begin{aligned}
& \text{Minimize: } K \\
& \text{Subject to: } P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) \geq \epsilon \\
& \mathbf{m}^T \mathbf{1} = M + K \\
& 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \\
& 0 \leq K \leq K_{\text{max}}
\end{aligned} \tag{3.15}$$

where we added the last constraint as the search criteria for the algorithm that finds the solution.

We are guaranteed to find the solution to optimization (3.15) for $K_{\text{max}} = \infty$, since $P_{\text{succ}}(\mathbf{m}, K, \mathbf{q})$ is a c.d.f. with the argument K . So, for $\epsilon < 1$ we are always going to find a feasible solution for some $K < \infty$. However, we limit the range of possible values of K by requiring that the efficiency be greater than some value $\eta(\mathbf{m}, K) \geq \delta$. Given δ we calculate

the maximum K , K_{\max} , as follows:

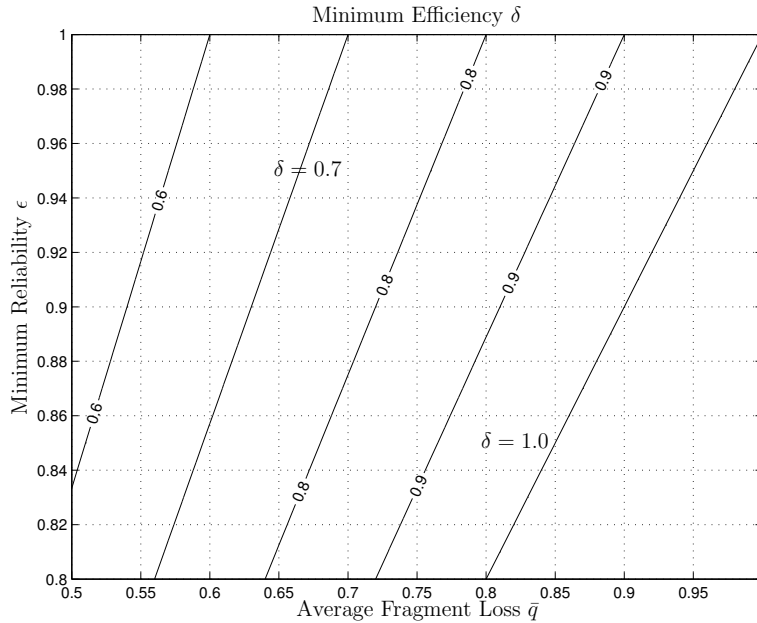
$$K_{\max} = \frac{1 - \delta}{\delta} M \quad (3.16)$$

We can determine an appropriate δ by using the properties of the upper layer transport layer. For example, if the higher level transport layer is TCP we can use selective-repeat ARQ to model the efficiency of the protocol. We have derived the efficiency of selective-repeat in Appendix B using the approximation of [74]. If we use (B.4) we can set the limit on the maximum number of parity packets that can still improve the efficiency:

$$\eta_{\text{SR}} \leq \eta_{\text{SR-PD}} \rightarrow \epsilon \frac{M}{M + K_{\max}} \geq \bar{q} \rightarrow \delta \geq \frac{\bar{q}}{\epsilon} \quad (3.17)$$

were we use \bar{q} to approximate empirical packet success rate on the channel. We arrive the left side of the equation by noting that the efficiency of the whole system should be greater than the efficiency of the selective-repeat ARQ. However, we may never be able to achieve this level of efficiency due to the assumptions we made in Appendix B so this estimate should only be used as a first order estimate for the minimum δ . Fig. 3.6 shows the acceptable δ for which the overall system is more efficient than the selective-repeat ARQ on its own. For example, for the average packet loss of $\bar{q} = 0.75$ and minimum reliability of $\epsilon = 0.94$ the efficiency of path diversification has to be larger than 0.8, to outperform the performance os selective-repeat ARQ. The acceptable region of δ is below and to the right of the the line indicating the minimum δ .

We can perform the optimization (3.15) with a linear search over at most K_{\max} items. Algorithm 3 shows how to perform the linear search. At every step if the solution is not found, the algorithm adds one more parity packet and assigns it to a path with the lowest probability of failure with available resources. The algorithm ensures that the search finds the optimal number of extra parity packets K since it assigns the packets in the most optimal

Figure 3.6: Minimum Efficiency δ

way. The algorithm may not find a solution for K in $0 \leq K \leq K_{\max}$. In such a case, we declare that the packet cannot be transmitted in the network. This case corresponds to the probability of blocking. However, if the connection is willing to accept lower reliability (decrease η), or pay for lower efficiency (increase K_{\max}) the blocking can be avoided.

The algorithm checks the validity of the solution by evaluating P_{succ} using Algorithm 1 proposed in Section 2.3.1. We also note that more efficient implementations of Algorithm 3 can be obtained if we use the recursive relationship (2.13). We will not discuss these techniques here.

Theorem 5 (Correctness Of Algorithm 3). *Algorithm 3 minimizes network utilization subject to $P_{\text{succ}} \geq \epsilon$.*

Proof. First, we note that P_{succ} is monotonically increasing function of K , so increase in K increases P_{succ} . We also note that every time we add a fragment we do that at the most optimal way to satisfy the P_{succ} constraint since we add it to the path with the highest probability of success that still has available space. This ensures that P_{succ} can only be

Algorithm 3 OPTIMIZE-UTILIZATION($\mathbf{M}_{th}, \mathbf{q}, \epsilon, \delta$)

Require: $\forall i > j, q_i \geq q_j$
Ensure: $\mathbf{m}^T \mathbf{1} \leq \mathbf{M}_{th}^T \mathbf{1}, i \leq n$

- 1: $M_{\text{total}} = 0, \mathbf{m} = \mathbf{0}$
- 2: $\mathbf{m} \leftarrow \text{MAXIMIZE-PSUCC}(\mathbf{q}, M, 0, \mathbf{M}_{th})$
- 3: $i \leftarrow \min\{\forall j : 1 \leq j \leq n, M_j \neq 0\} - 1$
- 4: $K_{\text{max}} \leftarrow \lfloor \frac{1-\delta}{\delta} M \rfloor$
- 5: **repeat**
- 6: **if** $i < n \wedge K \leq K_{\text{max}} \wedge M_i < M_i^{(th)}$ **then**
- 7: $K \leftarrow K + 1$
- 8: $M_i \leftarrow M_i + 1$
- 9: **else**
- 10: $i \leftarrow i + 1$
- 11: **end if**
- 12: **until** $P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) < \epsilon$
- 13: **if** $i > n \vee K > K_{\text{max}}$ **then**
- 14: **return** $\mathbf{m} \leftarrow \emptyset$
- 15: **else**
- 16: **return** $\mathbf{m} = \{M_1, M_2, \dots, M_n\}$
- 17: **end if**

increased by adding more parity fragments. Second, we note that the algorithm adds the parity fragment in increments of 1 at every step, so at every increment of K the next most optimum solution is explored, until all the solutions are considered. \square

The complexity of this algorithm is $O(K_{\text{max}}M(M + K_{\text{max}}))$. The proof of this statement follows the proof of Theorem 4, so we omit it. Better solution times can be achieved ($O(6 \times K_{\text{max}})$) by using the Poisson approximation to evaluate the solution instead of the exact evaluation, as we discussed in Section 2.3.2.

We can also solve the $\{0, 1\}$ -integer version of this problem (for the non-uniform probabilities of packet loss). Algorithm 3 can be modified in a similar way as Algorithm 2 in Section 3.2. The main idea is to consider the $\sum_{i=1}^n M_i^{\text{th}}$ slots where the algorithm can assign the fragments, and choose the top $M + K$ slots to assign the packet. Each slot corresponds to a probability of successful fragment transmission $q_i(j)$. The algorithm can be modified to add each new parity fragment to the slot with the highest probability of successful frag-

ment transmission, instead to the path with the highest probability of successful fragment transmission.

3.4 Minimization of Energy Consumption (P3)

In this section, we solve problem P3, where we minimize the total energy used by the connection, while the level of reliability and efficiency is maintained for the connection. This optimization is an example for path diversification to achieve more than just the basic QoS in the network. The optimization does not find the global minimum energy solution in the network, which would also minimize the total power spent by the source node. However, path diversification offers a good way to provide load balancing, which increases the overall network lifetime. We use the Poisson approximation for the network reliability bound, so that the optimization can be solved with linear programming.

In order to provide a solution for optimization P3, we assume that the source node has access to the information about energy consumption on each path in the network. We have shown in Section 2.2 how the energy consumption on the paths can be determined by probing the downstream nodes. We use the vector $\mathbf{E}_b = [e_1^{(b)}, e_2^{(b)}, \dots, e_n^{(b)}]^T$ to indicate per-bit energy consumption on each path. So, the total amount of energy spent to transmit the $M + K$ fragments is given by:

$$E_{\text{Total}}(\mathbf{m}, \mathbf{E}_b) = b\mathbf{m}^T \mathbf{E}_b \quad (3.18)$$

the optimization P3 minimizes the objective function $E_{\text{Total}}(\mathbf{m}, \mathbf{E}_b)$. The optimization is

given as the following mathematical program:

$$\begin{aligned}
& \text{Minimize: } E_{\text{Total}}(\mathbf{m}, \mathbf{E}_b) = b\mathbf{m}^T \mathbf{E}_b \\
& \text{Subject to: } P_{\text{succ}}(\mathbf{m}, K) \geq \epsilon \\
& \qquad \eta(\mathbf{m}, K) \geq \delta \\
& \qquad \mathbf{m}^T \mathbf{1} = M + K \\
& \qquad 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}}
\end{aligned} \tag{3.19}$$

where ϵ is specified by the connection and δ can either be specified by the connection or determined with the efficiency of upper layer protocols, as shown in Section 3.3. For $\epsilon = 0$ and $\delta = 0$ the optimization becomes trivial and amounts to a “greedy” algorithm similar to Algorithm 2, except that the paths are sorted in the order of increasing energy.

Note that the non-linearity of P_{succ} makes it possible that we can achieve lower energy consumption with a higher number of transmitted packets. For example, we can have a situation where we have path with high reliability and high energy consumption, and a path with low reliability and low energy consumption. Adding fragments to the path with low reliability, and subtracting fragments from the path with high reliability may decrease energy and still increase overall reliability.

In fact, the minimization in (3.19) is very difficult to solve due to the non-linearity of P_{succ} . We have no way of knowing the convexity of the function, since we have no closed form solution for it. However, the Poisson Approximation (2.18) gives a closed form approximation. One way to approach the problem would be to transform the minimization into a convex problem using the results of Appendix A, however we showed in Section 3.3 that this approach puts an upper bound on efficiency η , which may make most solutions infeasible due to the constraint that $\eta \geq \delta$. We use the properties of the derivative of $Q(\mathbf{m}, K)$ to transform the optimization into a linear program.

We transform the problem into a linear programming problem by using the fact that

$Q(\mathbf{m}, k)$ is a monotonically decreasing function of $\lambda(\mathbf{m})$ for a fixed K . So, for a given K there exists $\alpha(K)$ such that:

$$\lambda(\mathbf{m}) \leq \alpha(K) \rightarrow Q(\mathbf{m}, K) \geq \epsilon \rightarrow P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) \geq \epsilon \quad (3.20)$$

we show the behaviour of $\alpha(K)$ in Fig. 3.7.

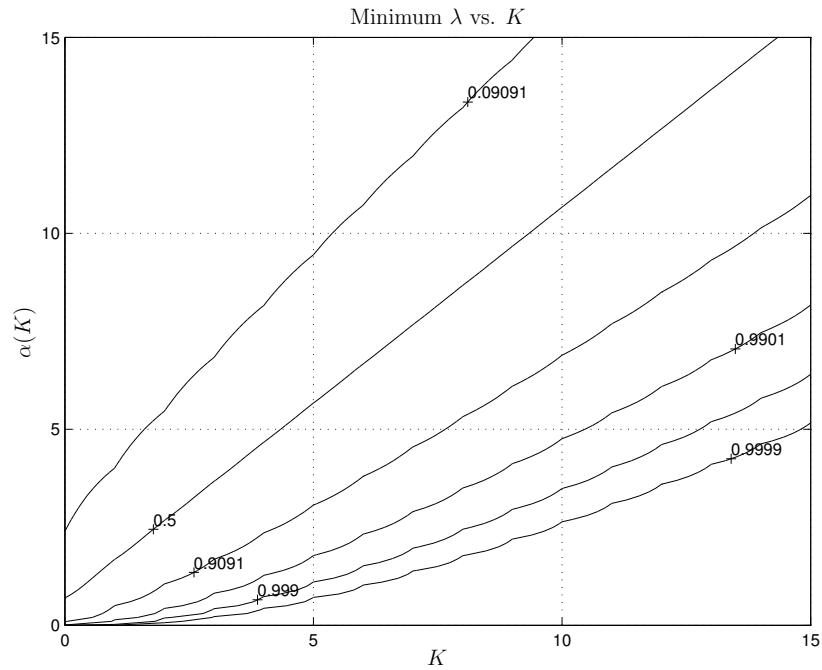


Figure 3.7: $\alpha(k)$ as the function of ϵ

We note two facts about $\alpha(K)$. First, we do not need to know the precise value of \mathbf{q} to calculate $Q(\mathbf{m}, K)$, so we only need to tabulate $\alpha(K)$ for the values of ϵ that we allow the connections to specify. Second, for values of $K > 5$, $\alpha(K)$ is practically a linear function. We will use both of these to find the solution for (3.19). We will use the table in the first method and the linearity in the second method.

Full Search with the Linear Approximation (Emin-1)

We now simplify the problem by using the results of optimizations P1 and P2 and the Poisson approximation. The solution of the optimization is to search for an allocation \mathbf{m} which has the minimum energy consumption and satisfies the constraints of the problem. We approximate the reliability constraint with $\alpha(K)$ and search over all K . We modify the optimization as follows:

$$\min E_{\text{Total}} = \min_{1 \leq k \leq K_{\max}} \{ \min \tilde{E}_{\text{Total}}(\mathbf{m}, \mathbf{E}_b) \} \quad (3.21)$$

where $\tilde{E}_{\text{Total}}(\mathbf{m}, \mathbf{E}_b)$ is the minimum energy for some value of k . We also omit the constraint set in (3.21) to make the equation simpler. So, we can approximate (3.19) by finding the minimum $\tilde{E}_{\text{Total}}(\mathbf{m}, \mathbf{E}_b)$ that solves the following problem for all $1 \leq k \leq K_{\max}$:

$$\begin{aligned} \text{Minimize: } & \tilde{E}_{\text{Total}}(\mathbf{m}, \mathbf{E}_b) = b\mathbf{m}^T \mathbf{E}_b \\ \text{Subject to: } & -\mathbf{m}^T \ln(\mathbf{q}) \leq \alpha(k) \\ & \mathbf{m}^T \mathbf{1} = M + k \\ & 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \end{aligned} \quad (3.22)$$

we use the lower case k to indicate that k is a constant in the optimization.

However, we can improve the running time of this search significantly if we limit the range of K . From optimizations P1 and P2 we can put the bound on K from the P_{succ} and $\eta(\mathbf{m}, K)$ constraints in (3.19). First, we find K_{\min} by solving P2 as described in Section 3.3. Second, we can find K_{\max} with (3.16). However, we can further restrict the range of K by finding the maximum K for which E_{Total} is still smaller than the minimum energy that can be achieved with K_{\min} . We can find this by using an algorithm similar to Algorithm 3, where the stopping criteria can be calculated by comparing E_{Total} with the energy that is obtained for K_{\min} , and the paths are sorted by increasing energy consumption. This assures that we

only take into account values of K in the feasible region of (3.19) and do not search in the region where the solution cannot be found.

Optimization (3.22) is a standard integer linear programming problem that can be solved with well-known techniques. Unfortunately, this is still a difficult problem to solve since it is an integer problem, and the constraint matrix is not unimodular¹ [77]. However, we can solve the integer version of the problem by using linear programming and the *branch and bound* technique, or by using convex programming (to solve the linear program) and the branch and bound technique, or any other integer programming technique. Nevertheless, in practical situations, there will only be a few of variables in the optimization (the number of paths), which should make the running time reasonable.

Full Linear Approximation (Emin-2)

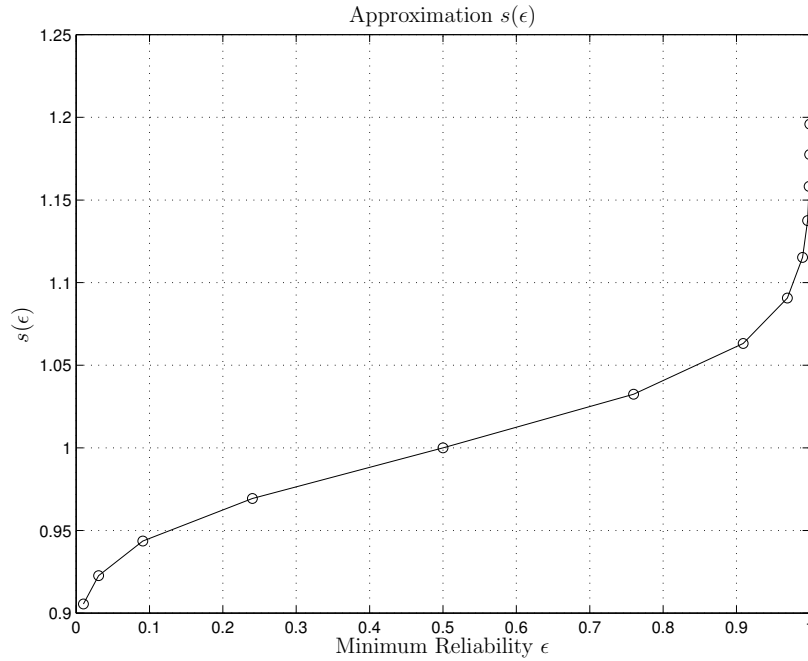
We now improve the performance of the search by a factor of $\Theta(\Delta K)$, where $\Delta K = K_{\max} - K_{\min}$, by using the apparent linearity of $\alpha(K)$ to approximate the problem further. We approximate $\alpha(K)$ as a straight line for a given ϵ :

$$\alpha(K) \approx s(\epsilon)K + c(\epsilon) \quad (3.23)$$

the slope $s(\epsilon)$ and constant $c(\epsilon)$ can be obtained with any number of techniques and stored on the nodes prior to field deployment of the nodes. We plot $s(\epsilon)$ vs. ϵ in Fig. 3.8.

With this modification we can optimize $E_{\text{Total}}(\mathbf{m}, K, \mathbf{E}_b)$ directly, by approximating

¹A unimodular matrix consists of sub-matrices whose determinants can only be $\{-1, 0, +1\}$

Figure 3.8: $s(\epsilon)$ vs. ϵ

(3.19) with a linear program:

$$\text{Minimize: } E_{\text{Total}}(\mathbf{m}, K, \mathbf{E}_b) = b\mathbf{m}^T \mathbf{E}_b$$

$$\text{Subject to: } -\mathbf{m}^T \ln(\mathbf{q}) - s(\epsilon)K \leq c(\epsilon)$$

$$0 \leq K \leq \frac{1 - \delta}{\delta} \tag{3.24}$$

$$\mathbf{m}^T \mathbf{1} - K = M$$

$$0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}}$$

We can also solve this approximation with a linear programming techniques or convex programming techniques, similar to minimization (3.22).

3.5 Maximization of Network Lifetime (P4)

In this section, we solve optimization P4, maximization of network lifetime. We have defined the network lifetime earlier as the time until one of the nodes spends all of its energy. The loss of a node results in higher load on the rest of the network and in sensor networks it may result in loss of data. In a remote ad hoc network, lifetime is a key design issue since there may be no way to recharge the node with no energy and put it back into service. This is also a key issue in deploying ad hoc networks were the nodes can be recharged, since a part of the deployment must be a recharging plan. This optimization is a second application in which we show how path diversification can be used to achieve more than just simple QoS in the network.

Optimization P4 maximizes the network lifetime by maximizing the lifetime of the path with the minimum lifetime. When there are no QoS restrictions this optimization can be solved with “water-filling”. However, just as in the optimization P3 with the QoS constraints the problem becomes difficult, so we use the Poisson approximation to solve the optimization with a linear approximation.

We have shown in Section 2.2 how to calculate the lifetime for a path using (2.8). The network lifetime, from the source node point of view, is the time at which one of the paths will break because the energy on that path has run out. So, the lifetime of the network from the source node point of view is given by:

$$T_{\text{net}}(\mathbf{m}) = \min_{1 \leq i \leq n} \{T_i\} = \min_{1 \leq i \leq n} \left\{ \frac{D_M M_i^{(e)}}{M_i} \right\}, \quad 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \quad (3.25)$$

we use the condition $0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}}$ to indicate the domain of the function $T_{\text{net}}(\mathbf{m})$. We note that $M_i^{(e)}$ is also used in the domain part of the equation, as the maximum number of fragments that can be transmitted before the energy on the path runs out. We also note that in practice the paths where $M_i^{(d)} > M_i^{(e)}$ should not be used in the optimization P4

since the use of such paths may in fact result in the drainage of power in one of the nodes, which is opposite to the result P4 strives to avoid.

We use the network lifetime time $T_{\text{net}}(\mathbf{m})$ as the objective function for optimization P4, so the optimization P4 becomes:

$$\begin{aligned}
\text{Maximize: } T_{\text{net}}(\mathbf{m}) &= \min_{1 \leq i \leq n} \left\{ \frac{D_M M_i^{(e)}}{M_i} \right\} \\
\text{Subject to: } P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) &\geq \epsilon \\
\eta(\mathbf{m}, K) &\geq \delta \\
\mathbf{m}^T \mathbf{1} - K &= M \\
0 &\preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}}
\end{aligned} \tag{3.26}$$

We add two QoS constrains, consistent to the requirements that

- P4 should achieve guaranteed level of reliability
- P4 should limit the number of parity fragments, K .

Maximization (3.26) is similar to the optimization of network lifetime defined in [53]. In fact, when there are no QoS constraints, it is easy to show that this optimization yields the same result as the optimization of [53], when paths are selected to increase the network lifetime. However, the use of path diversification allows us to optimize the network lifetime at the source node and not have to resort to an approximate flow redirection algorithm on each node in the network.

We could also add a constraint for the total energy spent in the network:

$$E_{\text{Total}}(\mathbf{m}, \mathbf{E}_b) = b\mathbf{m}^T \mathbf{E}_b \leq E_{\text{max}}. \tag{3.27}$$

There are two reasons to add this constraint. First, the energy spent by the connection can be used for pricing at the source node. For example, total energy consumption will depend

on the number of parity fragments used to increase the reliability, as well as the number of actual data fragments. Both of these reflect on energy consumption in the network. Second, even if the lifetime of the network is maximized, from the source node point of view, path diversification does not assume anything about the way the paths were picked. So, it is possible that the network load picked to maximize “network lifetime” may consume more than the network load that minimizes total energy consumption in the network, thus decreasing the actual network lifetime. The source node can use the energy restriction to prevent this from happening. Nevertheless, the addition of this constraint does not change the solution of the optimization significantly, so we ignore it.

In order to solve (3.26) we convert it to a minimization problem with a simpler objective function. We note that the objective function $T_{\text{net}}(\mathbf{m})$ is concave and that

$$T_{\text{net}}^{\text{cvx}}(\mathbf{m}) = \max_{1 \leq i \leq n} \left\{ \frac{M_i}{D_M M_i^{(e)}} \right\} \quad (3.28)$$

is a convex function with the same optimum point since the variables M_i are positive. So we rewrite (3.26) in a more convenient form as:

$$\begin{aligned} \text{Minimize: } & \max_{1 \leq i \leq n} \left\{ \frac{1}{M_i^{(e)}} M_i \right\} \\ \text{Subject to: } & P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) \geq \epsilon \\ & 0 \leq K \leq \frac{1 - \delta}{\delta} \\ & \mathbf{m}^T \mathbf{1} - K = M \\ & 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \end{aligned} \quad (3.29)$$

We drop the factor of D_M since the constant does not make a difference for the optimal allocation of fragments.

We are able to solve the optimization with both approximations we used to solve the

optimization P3. We only present the full linear approximation with (3.23) instead of the reliability constraint in (3.29). We convert the problem by introducing a new variable t [75] to arrive at:

$$\begin{aligned}
& \text{Minimize: } t \\
& \text{Subject to: } \frac{1}{M_i^{(\epsilon)}} M_i \leq t, \quad 1 \leq i \leq n \\
& \quad \quad \quad -\mathbf{m}^T \ln(\mathbf{q}) - s(\epsilon)K \leq c(\epsilon) \\
& \quad \quad \quad 0 \leq K \leq \frac{1-\delta}{\delta} \\
& \quad \quad \quad \mathbf{m}^T \mathbf{1} - K = M \\
& \quad \quad \quad 0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}}
\end{aligned} \tag{3.30}$$

We can solve this problem with either integer programming combined with the branch and bound technique, or with convex programming combined with the branch and bound technique.

Chapter 4

Numerical Results

In this chapter, we give numerical results to justify the use of path diversification in ad hoc network. We use Monte-Carlo simulations to generate sets of possible network scenarios and then apply the optimization from Chapter 3 to show the performance of path diversification in the scenarios. We show the results for optimization P1, the maximization of reliability, in Section 4.1. We show the results for optimization P2, the maximization of efficiency, in Section 4.2. We show the results for optimization P3, the minimization of energy consumption, in Section 4.3. We show the results for optimization P4, the maximization of network lifetime, in Section 4.4.

4.1 Results for P1, Maximization of P_{succ}

In this section, we present the numerical results for the reliability of path diversification. We have shown in Section 2.3 that network reliability is the QoS parameter that needs to be improved in ad hoc networks in order to make them accessible to regular network traffic. First, we show the performance of path diversification in networks where the full knowledge of future path status is known to the source node, and there are no restrictions on the delay and energy consumption on the paths in Section 4.1.1. In fact, this is the best performance

we can ever hope to achieve with path diversification. Second, we give a numerical results for a more realistic scenario in which there are restrictions on the number of fragments that can be transmitted on the paths in Section 4.1.2. The limit on the number of fragments limits the performance of path diversification, however it gives a more realistic performance of the scheme.

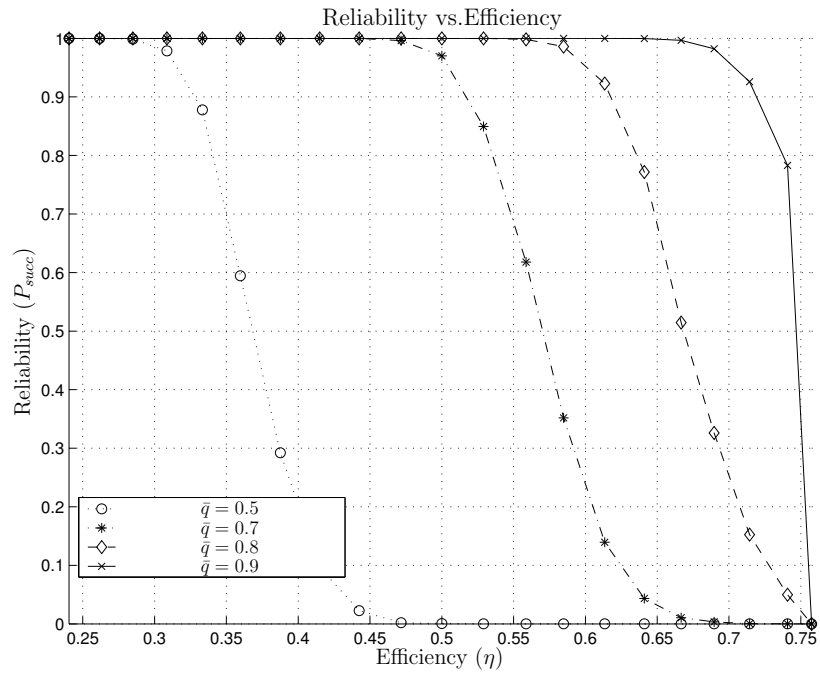
4.1.1 Resource-Unconstrained optimization of P_{succ}

We use the Monte-Carlo method to generate 200 different sets of q_i for $n = 5$ paths. The path success probabilities q_i are drawn from a uniform distribution with the mean \bar{q} . Fig. 4.1a illustrates the maximum achievable reliability for several values of \bar{q} . The maximum P_{succ} can be achieved only for resource-unconstrained paths (i.e. $\mathbf{M}_{\text{th}} = \infty$). We plot η on the horizontal axis and the average value for P_{succ} on the vertical axis.

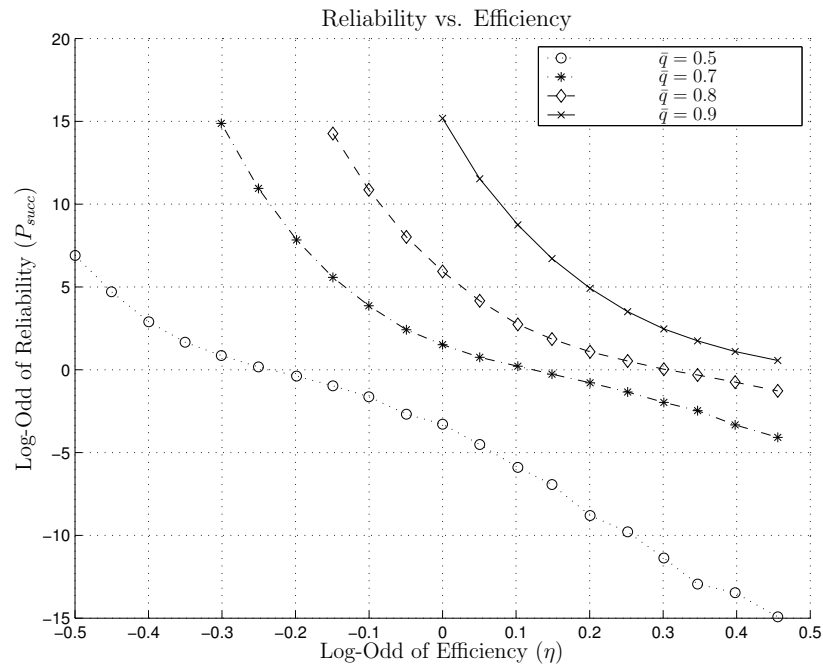
We also give a log-normal plot of the same graph in Fig. 4.1b. The values in log-odd scale are plotted as $\log_{10}\left(\frac{\alpha}{1-\alpha}\right)$ in place of α . The log-odd scale allows us to map the set $[0, 1]$ uniformly to the set $[-\infty, \infty]$, so that we can see the asymptotic effect when $P_{\text{succ}} \rightarrow 0$ or $P_{\text{succ}} \rightarrow 1$. We give some sample values for $\alpha \geq 0.5$ in Table 4.1. We note that the value in the log-odd table also corresponds to how close the value of α is to 1. The log-odd value of l (l positive integer) corresponds to the actual value of $\alpha = 1 - 10^{-l}$. We will use log-odd scale for all other plots in the thesis in order to make the plots easier to analyze.

Table 4.1: The Log-Normal Transformation

$\log_{10}\left(\frac{\alpha}{1-\alpha}\right)$	α	$\log_{10}\left(\frac{\alpha}{1-\alpha}\right)$	α
0.0	0.500	0.7	0.834
0.1	0.557	0.8	0.863
0.2	0.613	0.9	0.888
0.3	0.666	1.0	0.909
0.4	0.715	2.0	0.990
0.5	0.759	3.0	0.999
0.6	0.799	4.0	0.9999



(a) Reliability vs. Efficiency (Regular Plot)



(b) Reliability vs. Efficiency (Log-Odd Plot)

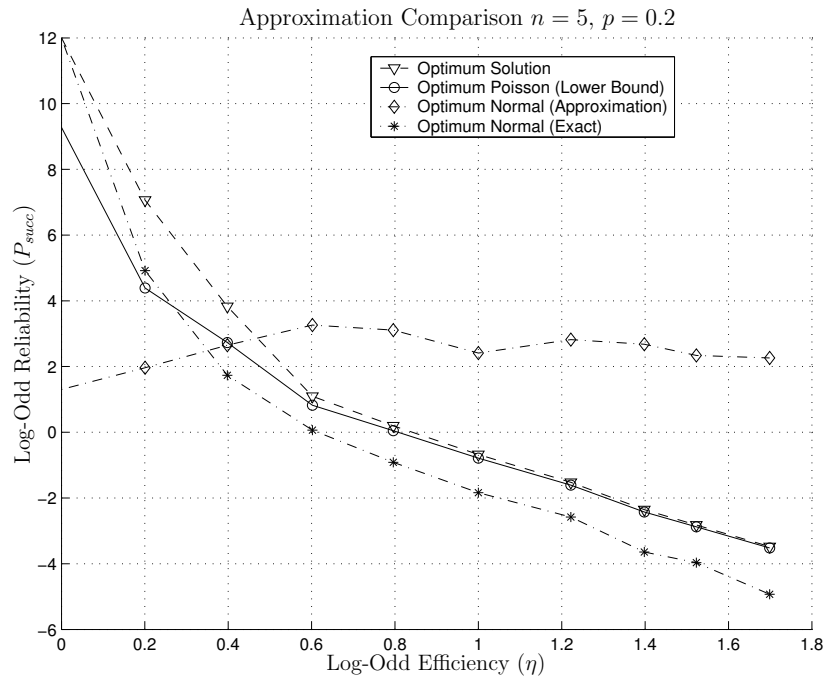
Figure 4.1: Reliability of *Path Diversification*

Path diversification is a very efficient way to ensure network reliability. For example, for $\bar{q} = 0.8$, we can achieve the reliability of 0.999 with the overhead of 35%. We can achieve the same level of reliability for $\bar{q} = 0.6$ with the overhead of 89%.

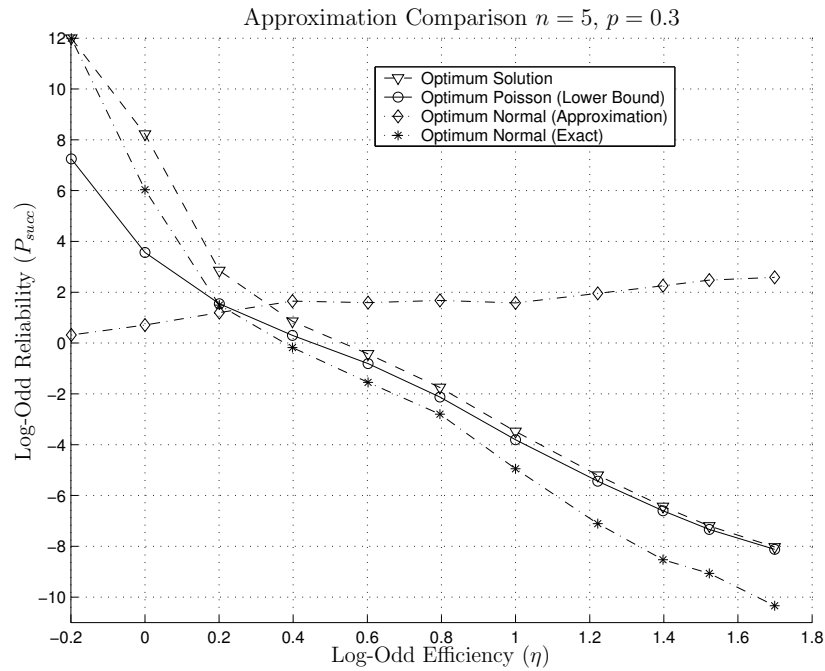
We compare the Normal and Poisson approximations with the Optimal solution in Fig. 4.2. The results are shown for the resource-unconstrained solution. We use a Monte-Carlo simulation to generate 200 scenarios in which the average probability of failure on a path is 0.2 and 0.3. In this example the number of fragments $M = 100$. The plot of “Optimum Poisson (lower bound)” is the plot of $Q(\mathbf{m}, K)$ with the optimum allocation vector. The plot of “Optimum Normal (Approximation)” is the plot we obtained by using the optimization of P_{succ} given in [64] and we plotted the estimate of P_{succ} using the Normal approximation. The plot of “Optimum Normal (Exact)” is the plot of the the exact value of P_{succ} we obtain by using the allocation method of [64].

We can see from Fig. 4.2 that the lower bound of the Poisson approximation of P_{succ} gives results that are very close to the actual value of P_{succ} . The difference between the Poisson approximation and the exact value of P_{succ} differ by less than 1 in the log-odd scale. From Table 4.1, we see that such a difference is really only in the last digit of the two values. So, the Poisson approximation is a good candidate to replace Algorithm 1 since very good values of P_{succ} can be obtained in relatively small number of steps. For example, with just 10 iterations, we can calculate $Q(\mathbf{m}, K)$ with precision of less than 10^{-6} , which is less than the error of the Poisson approximation for log-odd efficiency higher than 0.2.

It is also clear from Fig. 4.2 that the Normal approximation gives an incorrect estimate of P_{succ} , compared to the Poisson approximation, especially for low values of $M + K$. We also see that exact evaluation of P_{succ} for the packet allocation through the Normal optimization gives results which are suboptimal.



(a) Normal vs. Poisson Approximations $n = 5, \bar{p} = 0.2$



(b) Normal vs. Poisson Approximations $n = 5, \bar{p} = 0.3$

Figure 4.2: Comparison to the Normal and Poisson Approximation

4.1.2 Constrained optimization of P_{succ}

We demonstrate the effect of the QoS constraints given by \mathbf{M}_{th} , on P_{succ} in Fig. 4.3. We use the Monte-Carlo simulation to generate 1000 sets of \mathbf{M}_{th} constraint vectors from a Poisson distribution. The vector of path success rates was fixed to $\mathbf{q} = [0.85, 0.7, 0.7, 0.7, 0.7]^T$. We plot P_{succ} for different values of the Poisson parameter μ in the log-odd scale. As the value of the number of fragments we can transmit on each path increases, the value for P_{succ} increases as well. This is because we can transmit more packets on the path with the higher reliability, so the overall network reliability increases.

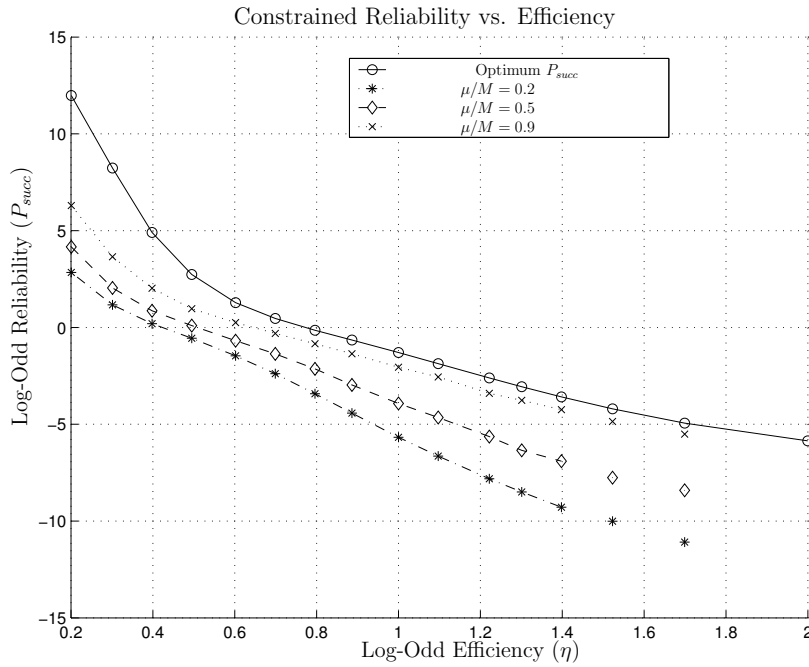


Figure 4.3: QoS Constrained Reliability vs. Efficiency

We define the blocking probability for the connection as:

$$P_{\text{block}} = \Pr \left[\sum_{i=1}^n M_i^{(\text{th})} < M + K \right]. \quad (4.1)$$

In Fig. 4.4 we show the theoretical probability of blocking as the function of the number of

paths n in the network. We use the Poisson distribution to set the number of constrained packets on each path \mathbf{M}_{th} . We can see that as the number of fragments that can be transmitted on a path decreases, a connection may still be admitted if the number of paths between the source and the destination is large.

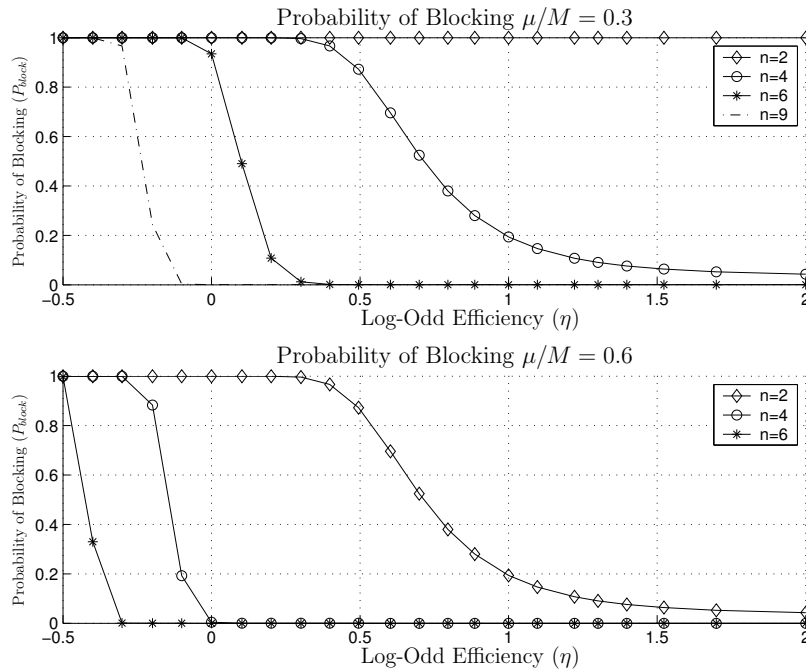
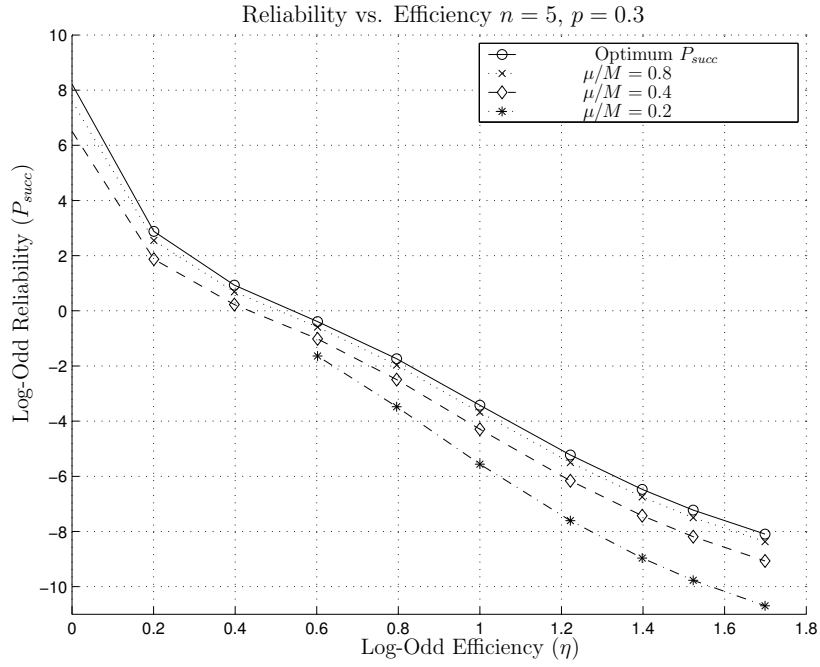


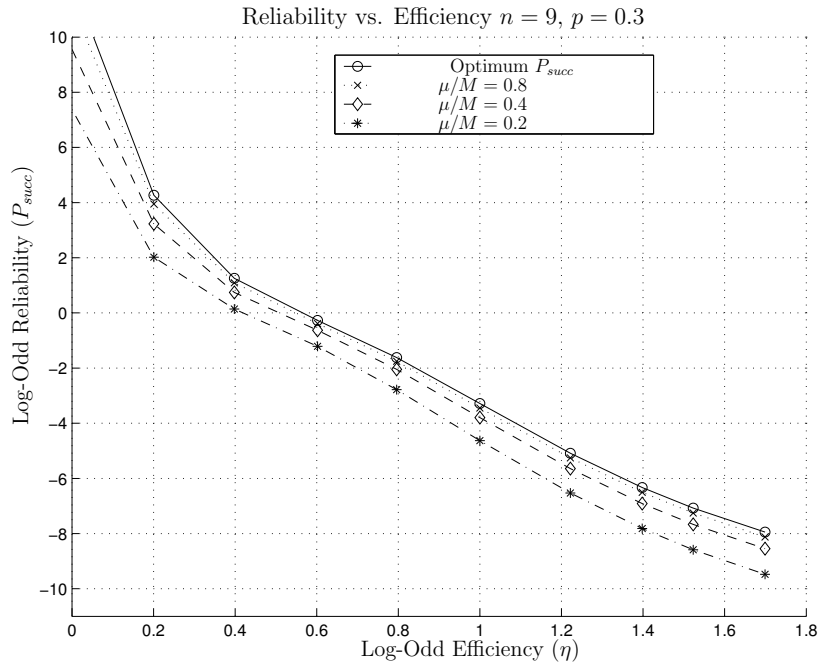
Figure 4.4: Blocking Probability vs. Efficiency

We perform another simulation with QoS constrained paths ($\mathbf{M}_{\text{th}} < \infty$). We create path scenarios with the Monte-Carlo method similar to the scenario in Fig. 4.1. We generate 200 independent sets of paths with average probability of failure $\bar{p} = 0.7$ as well as 1000 sets of \mathbf{M}_{th} vectors from the poisson distribution. Fig. 4.5 shows the behaviour of P_{succ} for different values of the Poisson parameter μ for $n = 5$ and $n = 9$ paths. The increase in the number of paths increases the performance of path diversification when the number of fragments that can be transmitted on the path is low. We can use this fact about path diversification when the estimates \hat{p}_i are not reliable. In that case the limit on the number of transmissions for which the number is valid is relatively small, so we need more paths to guarantee the

reliability.



(a) QoS Constrained Reliability vs. $n = 5, \hat{p} = 0.3$



(b) QoS Constrained Reliability vs. Efficiency $n = 9, \hat{p} = 0.3$

Figure 4.5: QoS Constrained Reliability of *Path Diversification*

4.2 Results for P2, Maximization of Efficiency

In this section, we give numerical results for optimization P2 where we optimize the efficiency of path diversification. We use the minimum achievable efficiency to show that path diversification improves the efficiency of the upper layer protocols even though it introduces overhead by using parity packets.

In Fig. 4.6 we show the maximum efficiency that can be achieved for different values of minimum reliability ϵ . We have performed a Monte-Carlo simulation for the scenario with $n = 5$ paths where we have set $M = 100$, and the threshold on each path to $M_i^{(\text{th})} = 50$. We generated a set of 1000 set of paths with average probability of failure of \bar{q} . In Fig. 4.6, we show the results for $\bar{q} = 0.5, 0.6, 0.7, 0.8$.

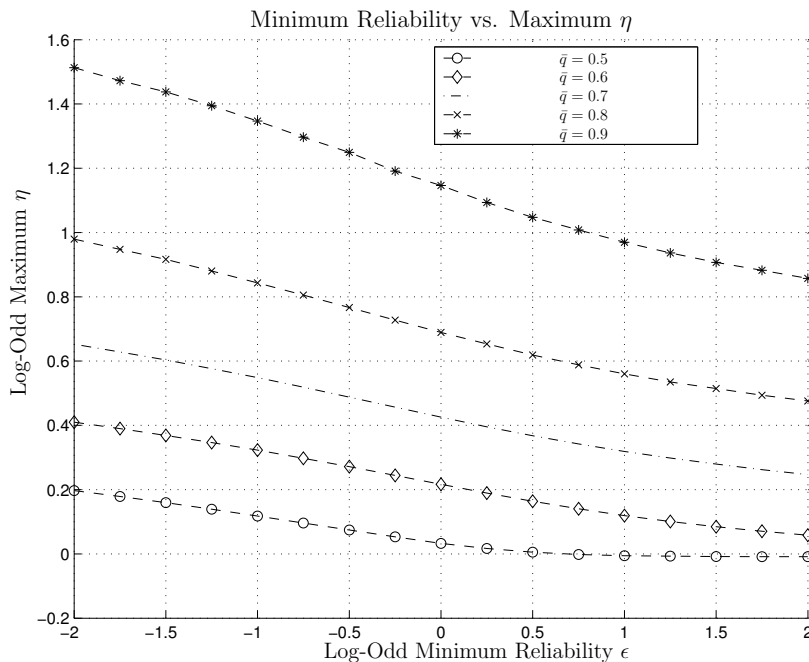


Figure 4.6: Minimum Reliability vs. Maximum Efficiency

In Fig. 4.7 we show the maximum efficiency that can be achieved for different values of minimum reliability ϵ . We performed a Monte-Carlo simulation for the scenario with $n = 5$ paths where we have set $M = 200$. We generated 200 sets of path scenarios with average

probability of failure of $\bar{p} = 0.3$, and 200 sets of thresholds from a Poisson distribution for each of $\mu = 20, 40, 60, 70, 100$. We indicate the ration of μ/M in the figure.

We show the blocking probability for the scenario in Fig. 4.8. Fig. 4.8a shows how the blocking probability changes when the average threshold μ increases compared to the number of packets M for $n = 5$ paths. Fig. 4.8b shows how the blocking probability changes with the number of paths when the ratio $\mu = 40$ and $M = 100$.

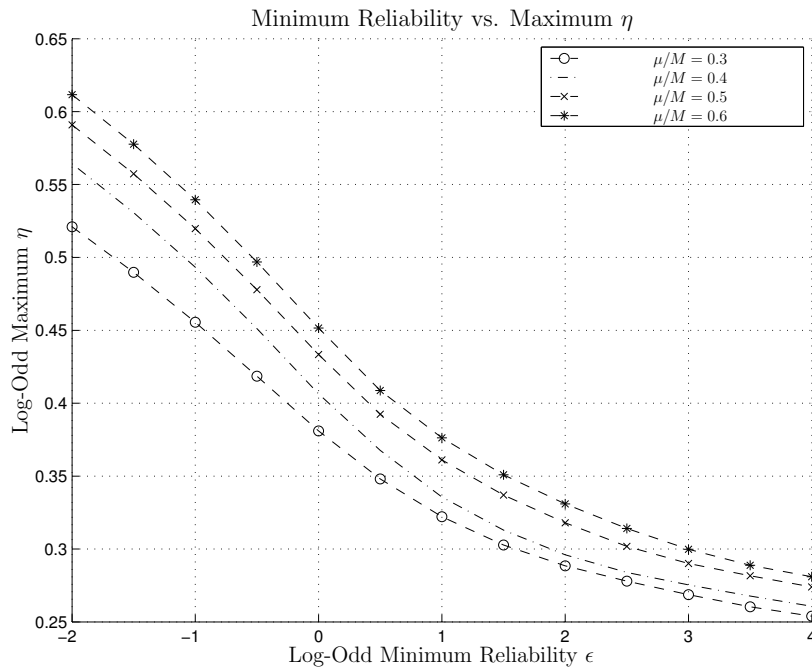
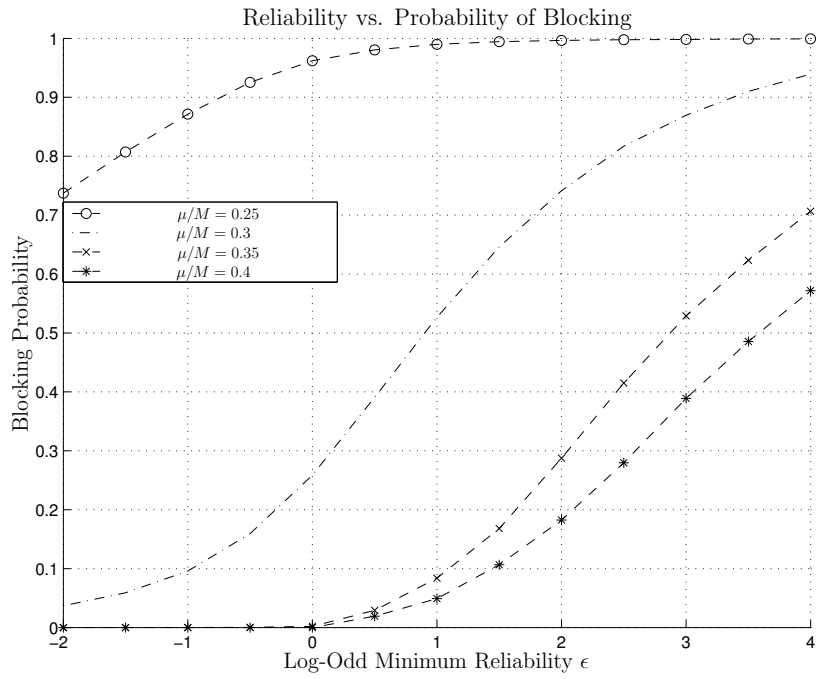
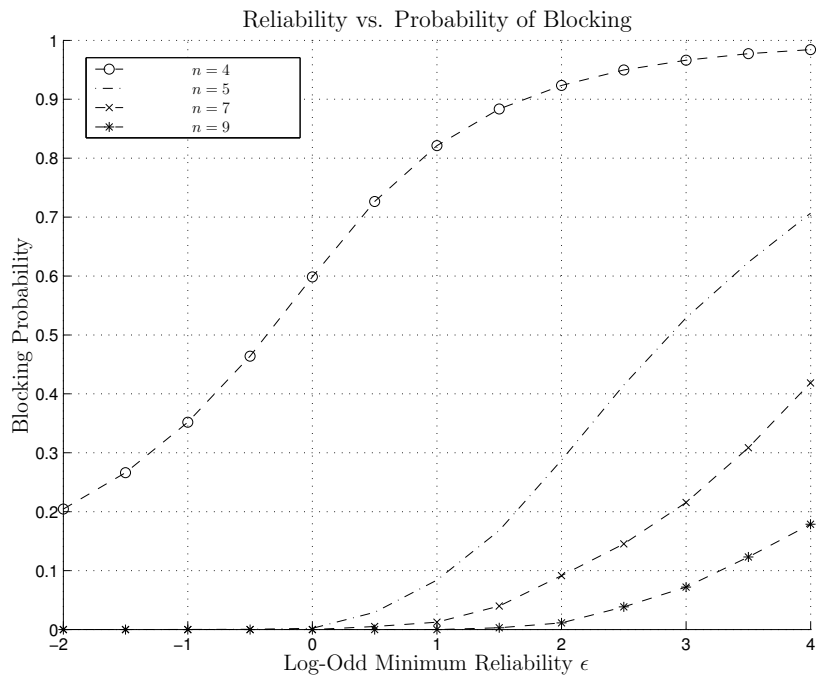


Figure 4.7: Minimum Reliability vs. Maximum Efficiency



(a) Changing μ



(b) Changing n

Figure 4.8: Minimum Reliability vs. Blocking Probability

4.3 Results for P3, Minimization of Energy Consumption

In this section, we give results for optimization P3, minimization of energy consumption. Energy is the limited resource in ad hoc networks, so improving the energy consumption on the node is important. We use the optimization approximations we derived in Section 3.4 to calculate the fragment allocation for the minimum energy at the source node.

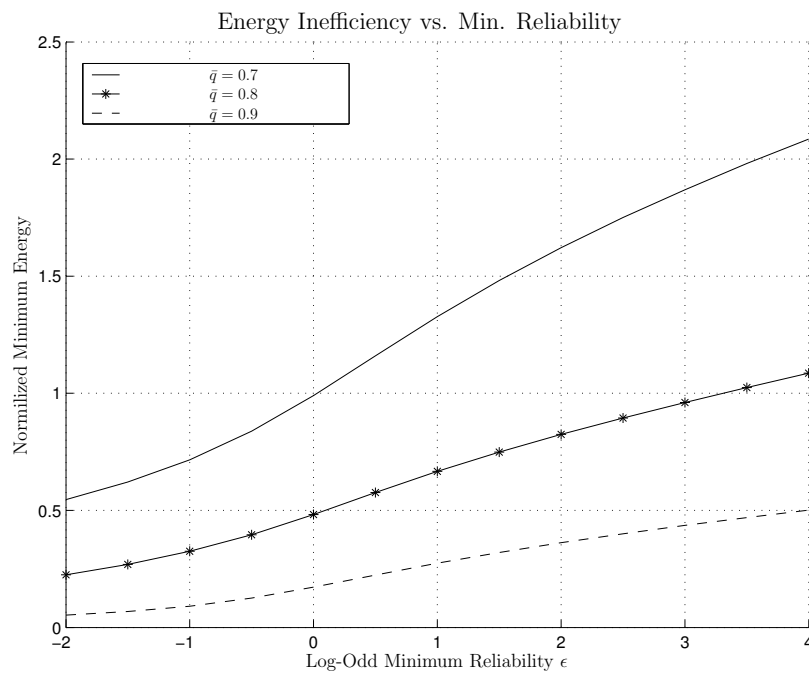


Figure 4.9: Minimum Energy vs. Minimum Reliability ϵ

We show the effect of improving network reliability on the energy consumption in Fig. 4.9. We ran a Monte-Carlo simulation where we generated 100 random energy consumption vectors from a uniform distribution with the mean 0.5. We also generated 100 vectors for path probabilities with $\bar{q} = 0.7, 0.8, 0.9$. We show the percentage increase in energy consumption as the function of required reliability in the network in Fig. 3.7. The minimum energy is achieved when the problem is optimized with no reliability constraints. In that

case, we can use a “greedy” algorithm to assign a maximum number of fragments allowed on the path with minimum energy consumption, and to the next path with minimum energy consumption, and so on until all the fragments are assigned.

In fact, if the path with minimum energy consumption is also the path with the globally minimum energy consumption¹, the figure shows the effect of network reliability on actual minimum energy consumption. In fact, we can see from Fig. 4.9 that with less than twice the increase in total energy consumption we can make the minimum reliability $\epsilon > 0.999$ even for channels where the probability of fragment loss is 20% on average. We showed how the increase in the TCP throughput is tenfold for this kind of reliability increase in Section 2.3.

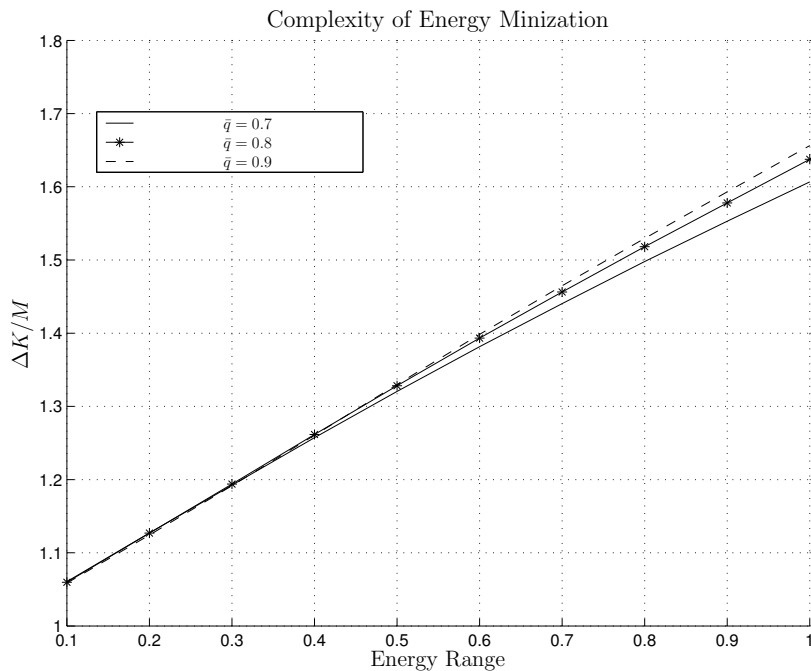


Figure 4.10: ΔK as a Function of ϵ and Energy Range

We use the same Monte-Carlo simulation to show the behaviour of $\Delta K = \Theta(K_{\max} - K_{\min})$ in Fig. 4.10. We showed in Section 3.4 that ΔK is the factor of increase in the number of steps the linear approximation, EMIN-1, has to perform, compared to the number of steps

¹A globally minimum path can be found by using one of the minimum energy routing methods we described in Section 1.6. (e.g. [10])

the full linear optimization, EMIN-2, has to perform. The factor is quite large and increases with the increase in the variance of energy consumption on the paths. Clearly, the full linear optimization of energy consumption decreases the complexity of the energy minimization considerably. We show the error in the full optimization in Fig. 4.11. The error is calculated as the amount of extra energy the connection spends if the full algorithm is used to allocate the fragments.

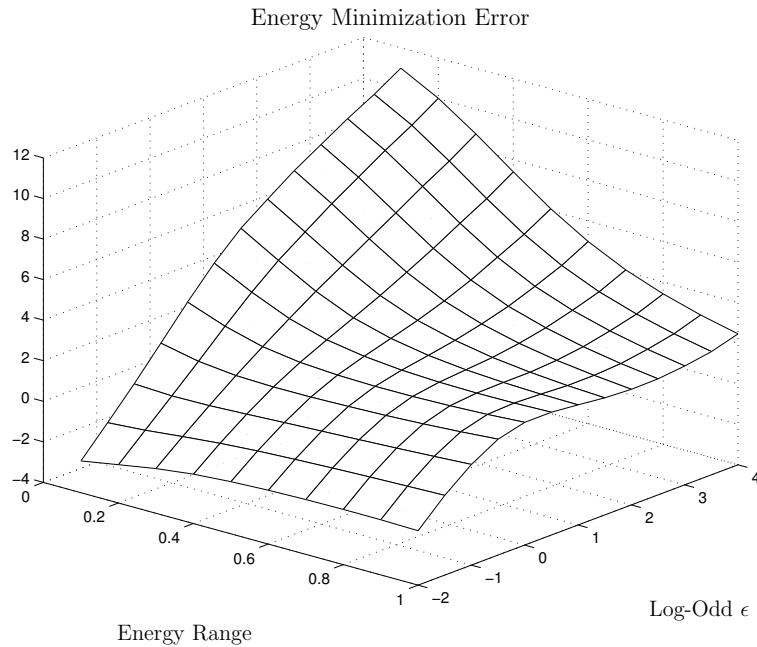


Figure 4.11: Error in Energy Minization

We examine the effect of energy diversity on the total energy consumption in Fig. 4.12. We created a Monte-Carlo simulation where we randomly generated 100 vectors of path reliability \mathbf{q} with a mean 0.7 and 100 vectors of energy consumption \mathbf{E}_b from a uniform distribution with the mean 0.5. We can see that the minimum energy increases with the increase of variance of \mathbf{E}_b . This is expected, since we are more likely to find a situation in which paths with higher reliability have also lower energy cost.

We show the effect of variance of energy consumption on the efficiency in Fig. 4.13. We used the same Monte-Carlo situation we used in Fig. 4.12. Efficiency is plotted for the

packet allocation that has the minimum energy consumption. We can see that the efficiency does not depend on the minimum reliability alone, anymore. In fact, the variance of energy consumption has a great impact, on the traffic efficiency we can expect in the network. Fig. 4.13 shows that of the variance in energy consumption is large we can decrease energy by adding more parity fragments.

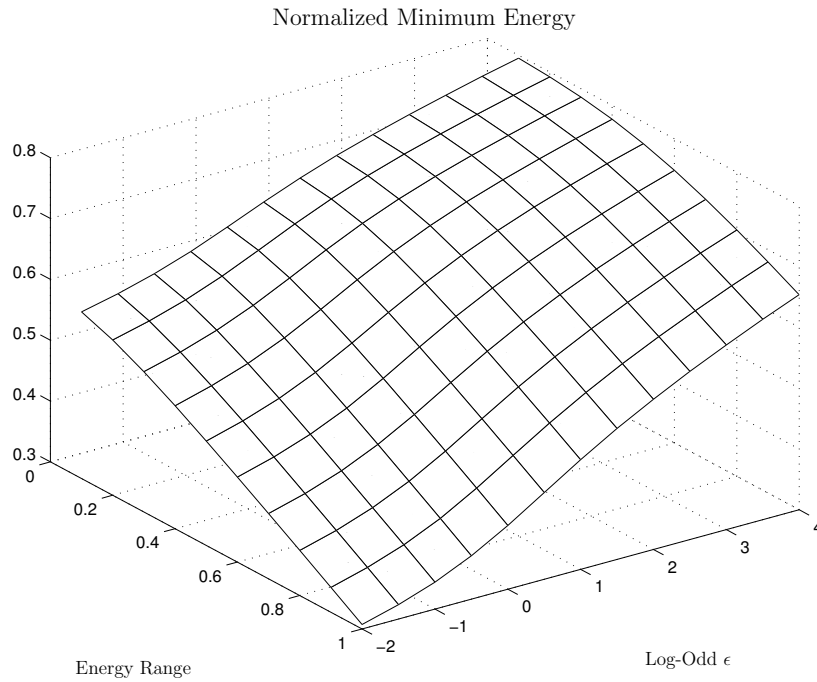


Figure 4.12: Normalized Minimum Energy

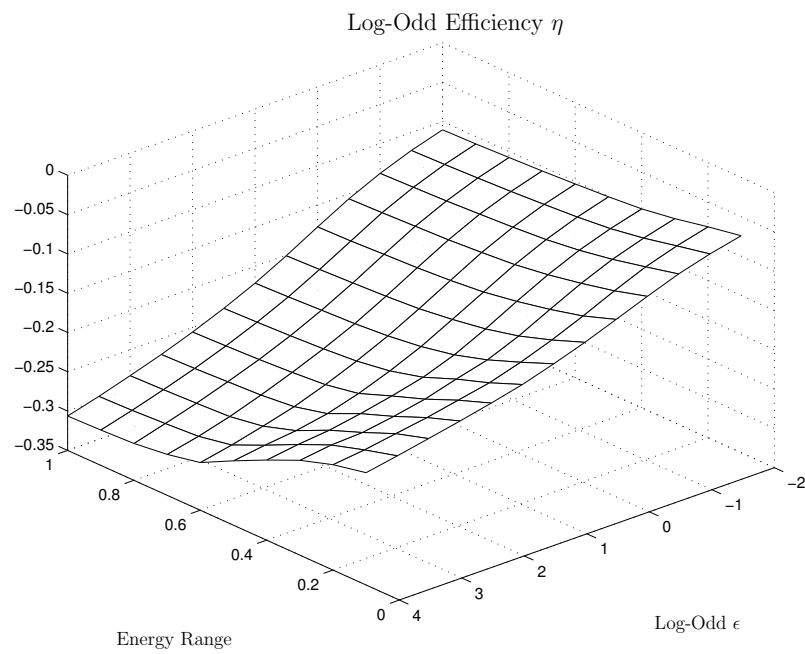


Figure 4.13: Efficiency at Minimum Energy

4.4 Results for P4, Maximization of Network Lifetime

In this section we give results for optimization P4, maximization of network lifetime. Network lifetime is an important parameter in ad hoc network, since the loss of a node results in outages and causes the network to reroute itself. We use the optimization approximation we derived at in Section 3.5 to calculate the fragment allocation for the maximum network lifetime at the source node.

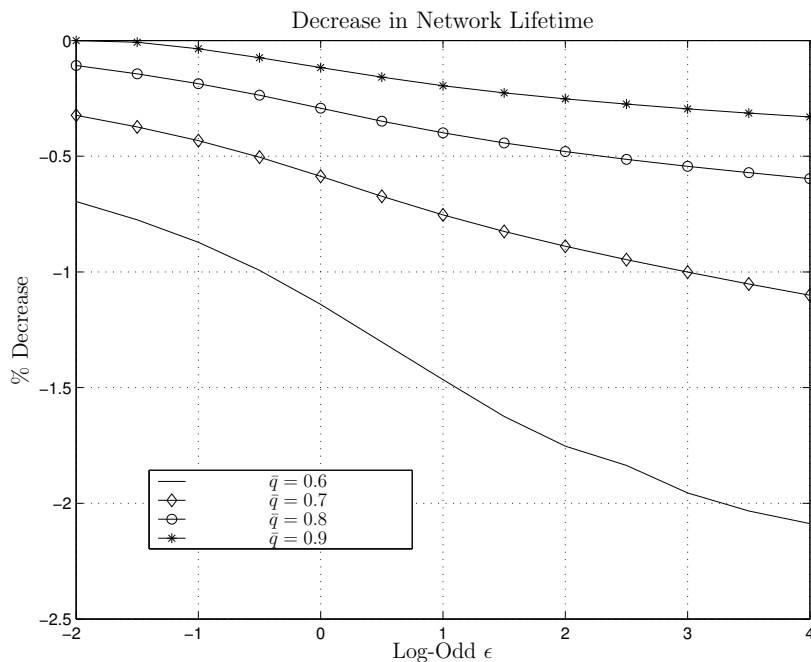
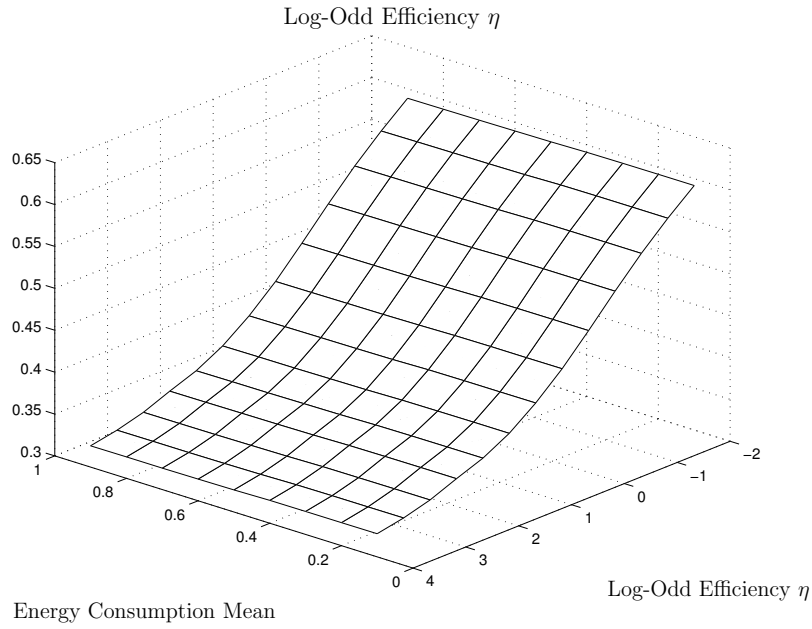


Figure 4.14: Decrease in Network Lifetime with Network Reliability

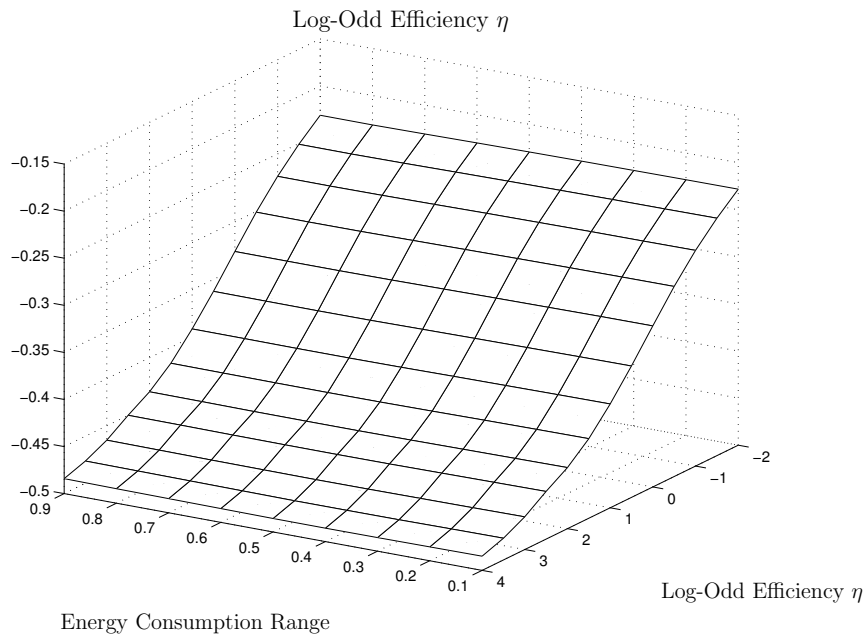
We show the effect the minimum reliability has on the network lifetime in Fig. 4.14. We ran a Monte-Carlo simulation in which we have generated 100 vectors of path reliability q with the mean $\bar{q} = 0.6, 0.7, 0.8, 0.9$, and 100 vectors of \mathbf{M}^e with the mean 0.5. Fig. 4.14 shows the percentage decrease in the network lifetime as the reliability increases. In fact, we can argue that the lifetime is not affected by increased level of reliability. For example, even when the network reliability is $\bar{q} = 0.6$ the lifetime does not decrease more than 2.5%. This is due to the load balancing quality of path diversification.

We show the effect of the energy consumption on each path on the efficiency in Fig. 4.15. We used a Monte-Carlo simulation in which we used 100 vectors \mathbf{q} with mean 0.6 to simulate the probability of fragment loss on the paths. We generated the energy consumption as in the Monte-Carlo simulation for Fig. 4.14. We can see that the energy consumption has very little impact on the efficiency, and that the bulk of inefficiency comes from the reliability bound.

We examine the effect of energy consumption on the paths, on the network lifetime in Fig. 4.16. We use the same Monte-Carlo simulation as in Fig. 4.15. We can see from the figure that for a fixed value of reliability the increase in the possible range of energy consumption decreases the network lifetime. This is an expected result since load balancing is more effective when the paths have are equivalent to each other.

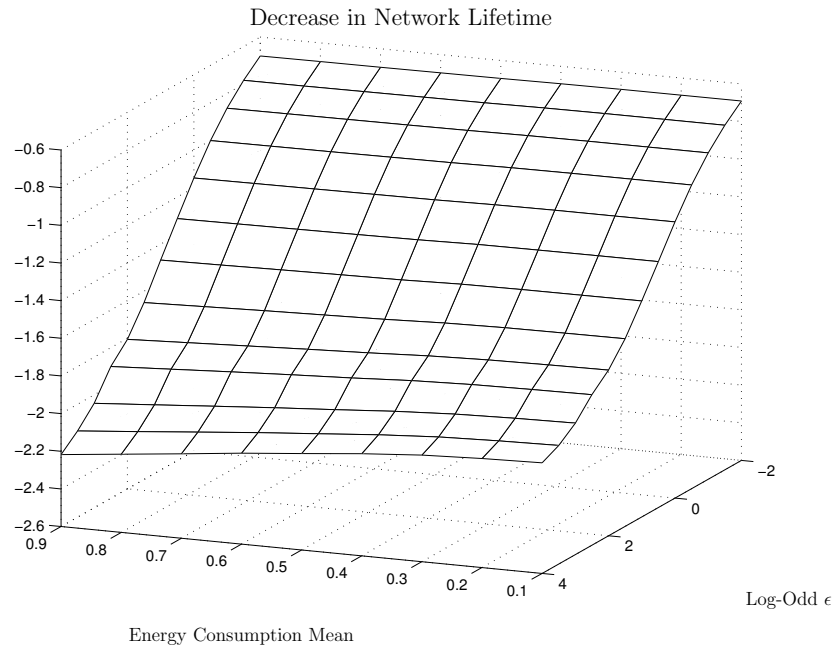


(a) Changing Mean of Network Lifetime Rate

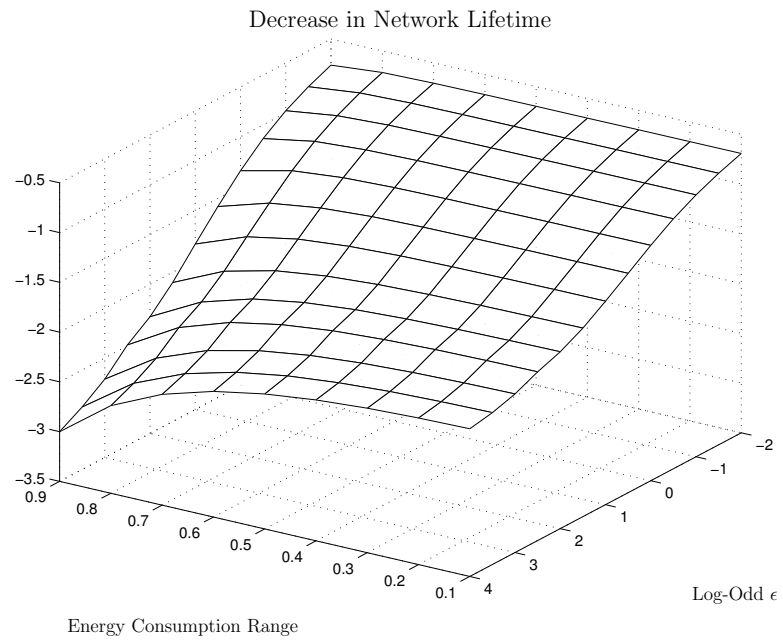


(b) Changing Variance of Network Lifetime Rate

Figure 4.15: Effect of P4 on Efficiency



(a) Changing Mean of Network Lifetime Rate



(b) Changing Variance of Network Lifetime Rate

Figure 4.16: The Effect of Energy Consumption on Network Lifetime

Chapter 5

Conclusion

This thesis proposes a new approach to increase reliability in wireless ad hoc networks called *path diversification*. The proposed technique uses multipath routing complemented with erasure codes to achieve arbitrarily high level of network reliability in ad hoc networks. We have shown that network reliability has a direct impact on the long term throughput of TCP connections in the network and is the most important Quality of Service (QoS) parameter in ad hoc networks.

Path diversification breaks each packet at the source node into fragments and sends the fragments over multiple parallel paths in the network. Path diversification also generates parity fragments from the original fragments to increase the likelihood that packets can be recovered at the destination node. We have shown that when no information is available about the network a good way to use path diversification is to evenly distribute the packet fragments on all the paths available in the network, we call this *blind* path diversification.

We have also shown how to collect network statistics which can improve the performance of path diversification. With the network statistics the source node can optimize the performance of path diversification by assigning a different number of fragments on each path. We have given a polynomial-time algorithm to find the path allocation that maximizes network reliability. We have used numerical simulations to illustrate the effectiveness of this

technique. We have compared our results with Normal and Poisson approximations. We have also given a polynomial-time algorithm to find the maximum efficiency of the scheme for a given reliability. We have shown how linear programming can be used to allocate the number of fragments on each path, so that the total consumed energy is minimized. We have also shown how linear programming can be used to allocate the fragments, so that the network lifetime from the source node point of view is maximized.

Path Diversification provides an arbitrarily high level of reliability with an acceptable amount of overhead. We have shown that the number of paths in the network does not necessarily need to be large to have reliable transmission. We have also shown how to apply path diversification to achieve QoS goals in ad hoc network other just increase in network reliability, such as minimization of energy cost, and maximization of network lifetime.

Appendix A

Convexity of $Q(\mathbf{m}, K)$

We derive the conditions for the concavity and convexity of $Q(\mathbf{m}, K)$. We defined $Q(\mathbf{m}, K)$ in Section 2.3.2 as:

$$Q(\mathbf{m}, K) \leq P_{\text{succ}} \leq Q(\mathbf{m}, K) + \frac{1}{2} \sum_{i=1}^M M_i \ln^2(q_i) \quad (\text{A.1})$$

where,

$$Q(\mathbf{m}, K) = \sum_{j=0}^K \frac{e^{-\lambda(\mathbf{m})} [\lambda(\mathbf{m})]^j}{j!} \text{ and } \lambda(\mathbf{m}) = \sum_{i=1}^n \ln(q_i^{-M_i}) = -\mathbf{m}^T \ln(\mathbf{q}) \quad (\text{A.2})$$

and $\ln(\mathbf{q}) = [\ln(q_1), \ln(q_2), \dots, \ln(q_n)]^T$ is the natural logarithm of the vector of probabilities of success.

We use the results for the convexity of $Q(\mathbf{m}, K)$ in the derivation of optimization techniques that use the $Q(\mathbf{m}, K)$ function. We note that $\lambda(\mathbf{m})$ is a linear function, so that $\nabla \lambda(\mathbf{m}) \succeq 0$, and $\nabla^2 \lambda(\mathbf{m}) = 0$.

We now use the following relations to determine the convexity of $Q(\mathbf{m}, K)$:

$$\begin{aligned}
Q(\mathbf{m}, K) &= e^{-\lambda(\mathbf{m})} + \sum_{j=1}^K \frac{e^{-\lambda(\mathbf{m})} [\lambda(\mathbf{m})]^j}{j!} \\
\frac{\partial Q}{\partial M_i} &= -\frac{\partial \lambda}{\partial M_i} e^{-\lambda(\mathbf{m})} \frac{[\lambda(\mathbf{m})]^K}{K!} < 0 \\
\frac{\partial^2 Q}{\partial M_i \partial M_i} &= \left(\frac{\partial \lambda}{\partial M_i} \right)^2 e^{-\lambda(\mathbf{m})} \frac{[\lambda(\mathbf{m})]^{K-1}}{(K-1)!} \left[\frac{\lambda(\mathbf{m})}{K} - 1 \right] \\
\frac{\partial^2 Q}{\partial M_i \partial M_j} &= \frac{\partial \lambda}{\partial M_i} \frac{\partial \lambda}{\partial M_j} e^{-\lambda(\mathbf{m})} \frac{[\lambda(\mathbf{m})]^{K-1}}{(K-1)!} \left[\frac{\lambda(\mathbf{m})}{K} - 1 \right]
\end{aligned} \tag{A.3}$$

We can see that $\nabla Q(\mathbf{m}, K) \prec 0$, and

$$\nabla^2 Q(\mathbf{m}, K) \begin{cases} \preceq 0, & \text{when, } \lambda(\mathbf{m}) \leq K \\ \succeq 0, & \text{when, } \lambda(\mathbf{m}) \geq K \end{cases} \tag{A.4}$$

So, $Q(\mathbf{m}, K)$ is concave when $K - \lambda(\mathbf{m}) \geq 0$, and:

$$\mathcal{S} = \{\mathbf{m} \in \mathbb{R}^n | \epsilon - Q(\mathbf{m}, K) \leq 0, \lambda(\mathbf{m}) - K \leq 0\} \tag{A.5}$$

is a convex subset set of $\mathcal{S}' = \{\mathbf{m} \in \mathbb{R}^n | Q(\mathbf{m}, K) > \epsilon\}$.

Appendix B

Efficiency of Selective-Repeat ARQ

Protocol

We derive the expression for the efficiency of the Selective ARQ protocol using the definition of efficiency given by (2.2). The efficiency is an important parameter to analyze since the increase in efficiency decreases the total amount of energy used in the network. We analyze the efficiency for the Selective-Repeat ARQ protocols as defined in [74]. In many ways this derivation is very similar to the discussion in [74] and is given here for completeness. We assume a very simple transmission model in which packet losses incur only in the forward direction with probability p independently of one another. The goal of this discussion is to compare the efficiency of Selective-Repeat ARQ scheme with, and without the path diversification, in terms of efficiency. Throughout the discussion we assume that the sender transmits a group of $M + K$ fragments infinitely many times. We ignore the size of the acknowledgment packets.

We analyze selective-repeat ARQ because it is the basic mechanism used by the TCP protocol [74], so the performance of selective-repeat ARQ gives us insights into the performance of TCP over lossy connections. The source in selective-repeat ARQ uses window control for

throughput control and it keeps timers for each packet in the window that was transmitted and not acknowledged. Packets are retransmitted if the timeout expires before the packet is acknowledged, or if a negative acknowledgment is received from the destination. The selective-repeat ARQ destination keeps the track of the successfully received packets and uses negative acknowledgments to request the retransmission of the lost packets.

We are interested in the average number of transmissions \mathcal{T} needed to transfer a packet successfully. The probability that a single packet is transmitted in i transmissions is given by:

$$\Pr[\mathcal{N} = i] = (1 - p)p^{i-1} \quad (\text{B.1})$$

Where \mathcal{N} is the random variable indicating the number of actual transmissions needed for a successful transmission. So the average number of actual transmissions needed to successfully transmit a packet is given by:

$$\mathbf{E}\mathcal{T} = \sum_{i=1}^{\infty} i \Pr[\mathcal{N} = i] = \sum_{i=1}^{\infty} i(1 - p)p^{i-1} = \frac{1}{1 - p} \quad (\text{B.2})$$

The efficiency of the selective-repeat ARQ protocol over a channel with probability of packet loss p is given by:

$$\eta_{\text{SR}} = \frac{M + K}{\frac{1}{1-p}(M + K)} = 1 - p. \quad (\text{B.3})$$

Efficiency over a path diversified channel is given by:

$$\eta_{\text{SR-PD}} = \frac{M}{\frac{1}{P_{\text{succ}}}(M + K)} = P_{\text{succ}} \frac{M}{M + K} = \eta P_{\text{succ}} \quad (\text{B.4})$$

where η is defined in (2.2) as the efficiency of path diversification.

Bibliography

- [1] Chai-Keong Toh. *Ad Hoc Mobile Wireless Networks*. Prentice Hall PTR, Upper Saddle River, New Jersey, 2002.
- [2] Jay Nemeroff, Luis Garcia, Dan Hampel, and Stef DiPierro. Application of sensor network communications. In *Proceedings of MILCOM*, pages 336–341, 2001.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 38(4):393–422, 2002.
- [4] Jean-Pierre Hubaux, Thomas Gross, Jean-Yves Le Boudec, and Martin Vetterli. Towards self-organized mobile ad hoc networks: The terminodes project. *IEEE Communications Magazine*, 39(1):118–124, January 2001.
- [5] Chieh-Yih Wan, Andrew T. Campbell, and Lakshman Krishnamurthy. PSFQ: a reliable transport protocol for wireless sensor networks. In *Proceedings of the first ACM international workshop on Wireless sensor networks and applications*, pages 1–11. ACM Press, 2002.
- [6] Seuong-Bum Lee, Gahng-Seop Ahn, and Andrew T. Campbell. Improving UDP and TCP performance in mobile ad hoc networks with INSIGNIA. *IEEE Communications Magazine*, 39(6):156–165, June 2001.

- [7] Gahng-Seop Ahn, Andrew T. Campbell, Andras Veres, and Li-Hsiang Sun. Swan: Service differentiation in stateless wireless ad hoc networks. In *INFOCOM 2002*, 2002.
- [8] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 64(2):388–404, March 2000.
- [9] Dan Chalmers and Morris Sloman. A survey of quality of service in mobile computing environments. *IEEE Communication Surveys*, Second Quater 1999.
- [10] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the fourth annual ACM/IEEE international conference on Mobile computing and networking*, pages 181–190. ACM Press, 1998.
- [11] Chai-Keong Toh. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *IEEE Communications Magazine*, 39(1):2–11, January 2001.
- [12] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hos networks. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications.*, volume 1, pages 22–31. IEEE, 2000.
- [13] C. Brendan S. Traw and Jonathan M. Smith. Striping within the network subsystem. *IEEE Network*, 9(4):22–32, July/August 1995.
- [14] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1996.
- [15] Asis Nasipuri, Robert Castaneda, and Samir R. Das. Performance of multipath routing for on-demand protocols in ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET) Journal*, 6(4):339–349, 2001.

- [16] Mahesh K. Marina and Samir R. Das. On-demand multipath distance vector routing in ad hoc networks. In *Proceedings of the International Conference for Network Protocols (ICNP)*, Riverside, November 2001.
- [17] Shahrokh Valaee and Bauchun Li. Distributed call admission control for ad hoc networks. In *Proceedings of Vehicular Technology Conference (VTC)*. IEEE, September 2002.
- [18] Coskun Cetinkaya and Edward W. Knightly. Egress admission control. In *Proceedings IEEE INFOCOM 2000, Conference on Computer Communications, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1471–1481, March 2000.
- [19] Coskun Cetinkaya, Vikram Kanodia, and Edward W. Knightly. Scalable services via egress admission control. *IEEE Transactions on Multimedia*, 3(1):69–81, March 2001.
- [20] Michael O. Rabin. Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. *Journal of the Association for Computing Machinery*, 36(2):335–348, April 1989.
- [21] Nicholas F. Maxemchuk. Diversity routing. In *Proceedings of IEEE International Communications Conference ICC'75*, pages 41.10–41.13, San Francisco, CA, June 1975.
- [22] Ender Ayanoglu, Chih-Lin I, Richard D. Gitlin, and J.E. Mazo. Diversity coding for transparent self-healing and fault-tolerant communication networks. *IEEE Transactions on Communications*, 41(11):1677–1686, November 1993.
- [23] Aristotelis Tsirigos and Zygmunt J. Haas. Multipath routing in the presence of frequent topological changes. *IEEE Communications Magazine*, 39(11):132–138, November 2001.
- [24] Haitao Wu, Yong Peng, Keping Long, Shiduan Cheng, and Jian Ma. Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement.

- In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM-02)*, volume 2, pages 599–607, Piscataway, NJ, USA, June 2002. IEEE Computer Society.
- [25] V. C. Zandy and B. P. Miller. Reliable network connections. In *Proceedings of MOBI-HOC 2002*, September 2002.
- [26] Karthikeyan Sundaresan, Vaidyanathan Anantharaman, Hung-Yun Hsieh, and Raghupathy Sivakumar. ATP: a reliable transport protocol for ad-hoc networks. In *Proceedings of the fourth ACM international symposium on Mobile ad hoc networking and computing*, pages 64–75. ACM Press, 2003.
- [27] Anindo Banerjea. Fault recovery for guaranteed performance communications connections. *IEEE/ACM Transactions on Networking*, 7(5):653–667, October 1999.
- [28] Srinivas Vutukury. *Multipath Routing Mechanisms for Traffic Engineering and Quality of Service in the Internet*. PhD thesis, University of California, Santa Cruz, May 2001.
- [29] Panagiotis Papadimitratos, Zygmunt J. Haas, and Emin Gun Sirer. Path selection in mobile ad hoc networks. In *Proceedings of Mobihoc 2002*. ACM, 2002.
- [30] Mechthild Stoer. *Design of Survivable Networks*, volume 1531 of *Lecture Notes in Mathematics*. Springer Verlag, Heidelberg, 1992.
- [31] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.
- [32] Shigang Chen and Klara Nahrstedt. Distributed quality of service routing in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1488–1505, August 1999.

- [33] J. W. Byers, M. Luby, and M. Mitzenmacher. Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads. In *IEEE Infocom 1999*, 1999.
- [34] Siddhartha Goel, Manish Singh, Dongyan Xu, and Baochun Li. Efficient peer-to-peer data dissemination in mobile ad hoc networks. In *Proceedings of the International Workshop on Ad Hoc Networking (IWAHN 2002)*, August 2002.
- [35] Philip K. McKinley, Chiping Tang, and Arun P. Mani. A study of adaptive error correction for wireless collaborative computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):936–947, September 2002.
- [36] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In *Proceedings of ACM SIGCOMM 1998*, 1998.
- [37] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 150–158, 1997.
- [38] Pablo Rodriguez and Ernst W. Biersack. Dynamic parallel access to replicated content in the internet. *IEEE/ACM Transactions on Networking*, 10(4):455–465, August 2002.
- [39] Anirban Mahanti, Derek L. Eager, Mary K. Vernon, and David J. Sundaram-Stukel. Scalable on-demand media streaming with packet loss recovery. *IEEE/ACM Transactions on Networking*, 11(2):195–209, 2003.
- [40] Takahiro Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-01)*, pages 1568–1576, Los Alamitos, CA, April 2001. IEEE Computer Society.

- [41] Hong V. Leong and Antonio Si. Data broadcasting strategies over multiple unreliable wireless channels. In *4th International Conference on Information and Knowledge Management*, pages 96–104, November 1995.
- [42] Jay Duncason. Inverse multiplexing. *IEEE Communications Magazine*, 32(4):34–41, April 1994.
- [43] Paul H. Fredette. The past, present, and future of inverse multiplexing. *IEEE Communications Magazine*, 32(4):42–46, April 1994.
- [44] Fabio M. Chiussi, Denis A. Khotimsky, and Santosh Krishman. Generalized inverse multiplexing of switched ATM connections. In *Proceedings of IEEE GLOBECOM'98: The Bridge to Global Intergration*, volume 5, pages 3134–3140, November 1998.
- [45] Thierry Van Landegem, Patrick Vankwikelberge, and Hans Vanderstraeten. A self-healing ATM network based on multiling principles. *IEEE Journal on Selected Areas in Communications*, 12(1):139–148, January 1994.
- [46] Subir K. Biswas, Rauf Izmailov, and Bhaskar Sengupta. Connection splitting: an efficient way of reducing call blocking in ATM. *IEEE/ACM Transactions on Networking*, 8(5):655–666, 2000.
- [47] H.-Y. Hsieh and R. Sivakumar. A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts. In *Proceedings of MOBIHOC 2002*, September 2002.
- [48] Akis Spyropoulos and C. S. Raghavendra. Energy efficient communications in ad hoc networks using directional antennas. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM-02)*, volume 1 of *Proceedings IEEE INFOCOM 2002*, pages 220–228, Piscataway, NJ, USA, June 23–27 2002. IEEE Computer Society.

- [49] Suresh Singh and C. S. Raghavendra. Power efficient MAC protocol for multihop radio networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, volume 1, pages 153–157. IEEE, 1998.
- [50] Elvino S. Sousa and John A. Silvester. Optimum transmission ranges in a direct-sequence spread-spectrum multihop packet radio network. *IEEE Journal on Selected Areas in Communications*, 8(5):762–771, June 1990.
- [51] Mung Chiang, Daniel O’Neill, David Julian, and Stephen Boyd. Resource allocation for qos provisioning in wireless ad hoc networks. In *Global Telecommunications Conference, 2001. GLOBECOM ’01*, volume 5, pages 2991–2915. IEEE, 2001.
- [52] David Julian, Mung Chiang, Daniel O’Neill, and Stephen Boyd. Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. In *INFOCOM*, pages 477–485. IEEE, 2002.
- [53] Jae-Hwan Chang and Leandros Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 1999.
- [54] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii Internation Conference on System Sciences*, 2000.
- [55] Chao Gui and Prasant Mohapatra. SHORT: self-healing and optimizing routing techniques for mobile ad hoc networks. In *Proceedings of the fourth ACM international symposium on Mobile ad hoc networking and computing*, pages 279–290. ACM Press, 2003.
- [56] Kyle Jamieson. Implementation of a power-saving protocol for ad hoc wireless networks. Master’s thesis, Massachusetts Institue of Technology, February 2002.

- [57] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, August 1999.
- [58] Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-01)*, pages 1388–1397, Los Alamitos, CA, April 2001. IEEE Computer Society.
- [59] Charles J. Cobourn. *The Combinatorics of Network Reliability*. Number 4 in The International Series of Monographs on Computer Science. Oxford University Press, New York, 1987.
- [60] Abhay K. Parekh and Robert G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.
- [61] Richard E. Barlow and Frank Proschan. *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart and Winston, Inc., New York, 1975.
- [62] Robert J. Serfling. A general poisson approximation theorem. *The Annals of Probability*, 3(4):726–731, 1975.
- [63] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of SIGCOMM'98*, pages 303–314, 1998.
- [64] Aristotelis Tsirigos. The effect of multipath routing on the packet delivery ratio. Master's thesis, Cornell University, April 2001.
- [65] Robert J. Serfling. The role of the poisson distribution in approximating system reliability of k-out-of-n structures. In Chris P. Tsokos and I.N. Shimi, editors, *Proceedings*

- on the The Theory and Applications of Reliability with Emphasis on Bayesian and Non-parametric Methods*, volume 1, pages 243–259, University of South Florida, December 1975. Academic Press, Inc.
- [66] Robert J. Serfling. Some elementary results on poisson approximation in a sequence of bernoulli trials. *SIAM Review*, 20(3):567–579, July 1978.
- [67] Douglas R. Shier. *Network Reliability and Algebraic Structures*. Claredon Press, Oxford, 1991.
- [68] Howard Frank and Ivan T. Frisch. Analysis and design of survivable networks. *IEEE Transactions on Communication Technology*, COM-18(5):501–519, October 1970.
- [69] Richard E. Barlow and Klaus D. Heidtmann. Computing k -out-of- n system reliability. *IEEE Transactions on Reliability*, R-33(4):322–323, October 1984.
- [70] Cecilio Pimentel and Ian F. Blake. Enumeration of markov chains and burst error statistisc for finite state channel models. *IEEE Transactions on Information Theory*, 48(2):415–428, March 1999.
- [71] Mordechai Mushkin and Israel Bar-David. Capacity and coding for the gilbert-elliott channels. *IEEE Transactions on Information Theory*, 35(6):1277–1290, November 1989.
- [72] Chiping Tang and Philip K. McKinley. Modeling multicast packet losses in wireless lans. Technical Report MSU-CSE-02-12, Computer Science and Engineering Department, Michigan State University, 2002.
- [73] Hong Shen Wang and Nader Moayeri. Finite-state markov channel - a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, February 1995.

- [74] Alberto Leon-Garcia and Indra Widjaja. *Communication Networks: Fundamental Concepts and Key Architectures*. McGraw-Hill Series in Electrical and Computer Engineering. McGraw-Hill, Boston, 2000.
- [75] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Number 1. <http://www.stanford.edu/boyd/cvxbook.html>, University of Stanford, December 2002.
- [76] Bruce D. Craven. *Fractional Programming*, volume 4 of *Sigma Series in Mathematics*. Heldermann Verlag, Berlin, 1988.
- [77] Laurence A. Wolsey. *Integer Programming*. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1998.