Reliable Packet Transmissions in Multipath Routed Wireless Networks

Petar Djukic and Shahrokh Valaee *

December 21, 2004

Abstract

We study the problem of using path diversification to provide low probability of packet loss (PPL) in wireless networks. Path diversification uses erasure codes and multiple paths in the network to transmit packets. The source uses Forward Error Correction (FEC) to encode each packet into multiple fragments and transmits the fragments to the destination using multiple disjoint paths. The source uses a load balancing algorithm to determine how many fragments should be transmitted on each path. The destination can reconstruct the packet if it receives a number of fragments equal to or higher than the number of fragments in the original packet.

We study the load balancing algorithm in two general cases. In the first case, we assume that no knowledge of the performance along the paths is available at the source. In such a case, the source decomposes traffic uniformly among the paths; we call this case blind load balancing. We show that for low PPL, blind load balancing outperforms single-path transmission. In the second case, we assume that a feedback mechanism periodically provides the source with information about the performance along each path. With that information, the source can optimally distribute the fragments. We show how to distribute the fragments for minimized PPL, and maximized efficiency given a bound on PPL. We evaluate the performance of the scheme through numerical simulations.

Index Terms: *wireless communication, fault tolerance, network monitoring, algorithm/ protocol design and analysis, linear programming*

1 Introduction

In this paper, we study the problem of using *path diversification* to provide probabilistic guarantees on quality-of-service (QoS) in multihop wireless networks. The QoS guarantees

*The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada, e-mail: {djukic,valaee}@comm.utoronto.ca

are bounds on the end-to-end delay and probability of packet loss (PPL). Path diversification has two components: *Forward Error Correction* (FEC) and *load balancing*. The source uses FEC to encode each packet into $M + K$ fragments [1], where M is the number of fragments in the original packet and K is the number of parity fragments. The source then transmits subsets of fragments over multiple disjoint paths. The allocation of fragments on each path is determined with a load balancing algorithm. The destination node attempts to reconstruct the packet with fragments it receives in less than $D_{\max}$ seconds after the original transmission. The reconstruction is possible with the FEC code if the destination receives M or more fragments. Our objective, is to devise a load balancing algorithm, which minimizes the probability that the destination receives less than M fragments (i.e. minimizes PPL), when the delay is fixed.

Examples of wireless networks where path diversification can be used are cellular networks with multihomed mobile hot-spots [2], mesh networks with roaming users [3, 4], sensor networks [5], and intelligent transportation systems [6]. In the sequel, we discuss the first two examples. A multihomed mobile hot-spot is connected to the backbone network through multiple receivers, and each receiver is connected to a different service provider [2, 7]. The home agent on the wired side of the network sends IP packets to different addresses of the multihomed remote agent at the mobile hot-spot. The reason for multiple interfaces is to get better coverage and higher bandwidth when available. The problem with this type of data transmission is that TCP may timeout due to disparate round trip times (RTTs) on the paths or enter fast retransmission mode due to out-of-order packets. In both cases, the TCP congestion algorithm is invoked needlessly [7]. This problem is accentuated in wireless networks where packets may be lost frequently at the link level, requiring retransmissions, which introduce larger path delay than the delay in wired networks.

For multihomed mobile hot-spots, path diversification can be implemented on top of the pre-existing architectures to provide guaranteed delay in the network layer [7, 2]. The delay guarantee improves the performance of TCP. Connection stripping for multihomed mobile hosts was also proposed in the transport layer [8]. Although, our approach can work with either layer, in the subsequent discussion we assume that the implementation is in the network layer. The advantage of implementing path diversification in the network layer is that no modifications are required in the TCP and IP protocols.

Mesh networks [9, 10, 3] are wireless ad hoc networks that can provide broadband wireless access with high data rates. A mesh network is made of many wireless routers interconnected with wireless links. Each router also serves as an access point for users in its vicinity. There are two problems with this type of network. First, the mesh connections may be implemented with IEEE 802.11 protocol, which decreases TCP throughput due to the exposed station problem, and collisions, which cause the link-layer to retransmit packets [11]. Second, it may be difficult to provide a guaranteed QoS to mobile users in mesh networks. For example, mobile IP hand-offs introduce delay that deteriorates TCP performance [12].

We propose path diversification as the solution to both problems in mesh networks. First, we assume that the link layer does not retransmit any packets since path diversification provides end-to-end reliability. Second, we assume that the source uses a multipath routing protocol and sends fragments along multiple disjoint paths to several access points in the user's vicinity.[1] The user listens to the access points and reconstructs the packets as soon as it receives M or more fragments. This scheme can be thought of as "information raining" to mobile users in the mesh network [13, 14]. The erasure code allows the user to reconstruct the packet even if some of the fragments are lost due to channel impairments or because the user is traveling between access points.

In both examples of wireless networks, TCP does not perform well due to high end-to-end PPL. For example, using the results of [15], it can be easily shown that a decrease in reliability from 0.999 to 0.95 (a decrease of 5%), decreases the throughput more than 5 times (80%). Note that reliability of 0.95 corresponds to a PPL of 0.05. So, it is important to achieve low PPL or high reliability in wireless networks for TCP to perform well.

[1]The paths may have to be frequency disjoint to have independent fragment transmissions over different paths.

Path diversification is a difficult problem to solve in the general case, which involves optimizing the routing and load balancing at the same time. For example, finding the optimal set of QoS constrained multiple disjoint paths is computationally hard [16]. The problem is more difficult in wireless networks where the channel conditions and node connectivity change with time. If, in addition, the solution should be implemented on the time scale of packet transmissions the problem becomes even harder. Therefore, we assume that the system is composed of two separate sub-layers. The first sub-layer is responsible for creating and maintaining multiple disjoint paths. Several solutions have been proposed for this problem in the literature [17, 18, 19, 20, 21, 22]. The second sub-layer distributes the fragments over the paths. In this paper, we focus on the latter sub-layer.

We address the problem of achieving low PPL in two steps. First, we assume source cannot collect any information about the path performance. In this case, the source distributes the fragments uniformly over the parallel paths. We call this case *blind load balancing.* We will show that blind load balancing outperforms single-path transmissions for low values of PPL. Second, we assume that some performance metrics about the parallel paths are available at the source. This can be the case if the destination moves slowly. The 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. The source node uses the information about the paths to periodically change the load on each path.

For the second case, where the source has information about the paths, we discuss two different optimization problems: minimization of packet loss and maximization of efficiency subject to a fixed packet loss. The optimum way to allocate the packets to minimize PPL is to use a *greedy* algorithm. The greedy algorithm allocates the maximum possible number of packets possible to the path with the smallest PPL and then the maximum possible number of packets to the path with the second smallest PPL, and so on. We show that in order to make this method robust, we need to limit the number of fragments on each path.

The second optimization maximizes the efficiency of the scheme. The efficiency is defined as the ratio of the size of the original packet to the amount of the transmitted data. We give an algorithm that performs a linear search for the smallest K fragments for which the minimum required reliability is satisfied. The algorithm increases K until it reaches the reliability threshold. We also examine the probability that the network may not be able to provide service at a given guaranteed reliability and efficiency; that is connections are blocked due to lack of resources.

We give a model of path diversification in Section 2 and use it in the subsequent sections to solve the optimization problems. Blind path diversification is analyzed in Section 3. We minimize PPL in Section 4, using two different techniques. The first technique uses an exact algorithm that calculates PPL under ideal conditions. The second technique uses the Poisson cumulative distribution function allowing us to account for non-ideal conditions in our optimization. We maximize efficiency in Section 4.3. Simulation results are given in Section 5 to illustrate the benefits of path diversification. Finally, we conclude the paper in Section 6. Next, we review the related literature.

1.1 Related Work

First, we review the related work in the QoS implementations for wireless networks and multipath routing, and then in path diversification. The work closest to ours is [24, 25] and we review this work in more details. We will also point out the differences between that work and our approach in the subsequent sections.

Generally, it is difficult to provide QoS in wireless networks. [26, 27] use reservation to guarantee QoS. However, the reservations are not effective since the source reserves the resources on a single path. If there is a serious impairment on the path or the path is broken because of mobility, the reservations and the packets on that path are lost. Another approach to improve QoS is multipath routing. The primary use for multipath routing has been to reduce route discovery time in ad hoc networks [17, 18, 19]. For example, in [17, 18] the source node finds multiple paths to the destination, but it only uses a single path for transmissions. The other paths are on standby and are only used if the main path fails. The usage of secondary paths reduces route discovery time, however it does not address the problem of unreliable links.

Other multipath schemes use multiple paths simultaneously. However, even if the multiple paths are used simultaneously, the performance of TCP decreases [20, 21]. [20] used connection splitting in ad hoc networks to transmit parts of a packet simultaneously over multiple links. However, this approach was shown in [21] to be inappropriate for TCP connections due to disparate delays on the disjoint paths. This is because packets get reordered or lost due to the discrepancy in the quality of the paths.

[28] introduced path diversification for wired networks. In that work, path diversification was called *dispersity routing*. However, this method is actually load balancing with FEC coding, so we use the term path diversification. In [28], a single fragment is transmitted on each path. The focus of the work was to analyze the decrease in delay due to load balancing, introduced with path diversification. The approach of [28] may not be appropriate for wireless multihop networks, especially if the wireless network is sparse, so that only a few disjoint paths are available between the source and the destination.

[24, 25] investigate path diversification in wireless multihop networks. In that work, the authors apply the path diversification of [28] to minimize packet loss in highly mobile ad hoc networks. The authors allow for multiple fragment transmissions on each path. The model of the path transmission used in [24, 25] is that if a fragment is lost on a path all of the fragments on that path are also lost. This model may be appropriate in highly mobile networks where path breakage means that all corresponding fragments are lost. However, the model is not appropriate in wireless networks with lossy connections.

Our work is appropriate for wireless networks with lossy connections. First, we assume that the source transmits each fragment individually. Second, we assume that the links are highly unreliable. In wireless networks, with mobile users, this is the case if there is fast fading or collisions in the physical channel, and if the link layer does not retransmit the lost packets. In this type of networks, the fragment losses can be approximated as independent or loosely correlated. In such a scenario, the load balancing of [24, 25] is not appropriate since there it is assumed that the fragment losses are completely correlated. Therefore, a new load balancing algorithm should be designed.

In addition to having a model distinct from [24, 25], we give a load balancing algorithm that maximizes the efficiency when the minimum packet loss is bounded. This algorithm is more practical than the algorithm that minimizes PPL since it allows the connections to specify the QoS, in terms of PPL, beforehand and then it minimizes the cost of the connection.

2 Path Diversification

In this section, we propose our version of path diversification for multihop wireless networks. We first describe path diversification and then we give the details of how information is collected at the source node. Finally, we give the wireless model used in the rest of the paper.

Fig. 1 shows how path diversification works. Assume that the source has multiple paths to the destination and that the paths are independent (i.e. the fragment delay is statistically independent on each of the paths.) This assumption is true if the networks, which carry the paths, are owned by different service providers [7], as in the example of multihomed mobile hot-spots, or if each access point in a mesh network can use multiple frequencies to forward fragments to other nodes. In the latter case, the paths should be arc-disjoint as well as frequency disjoint[2] for the statistical independence. An example of this is when wireless nodes have multiple network interface cards, each operating on different physical channels. For example, in the case of IEEE 802.11a, channels are created with different carrier frequencies and there are a total of 11 channels available.

[2]Two arc-disjoint paths have no nodes or edges in common [16], but in wireless environments they still have the first and the last edges in common. However, if the paths are frequency disjoint, the first and the

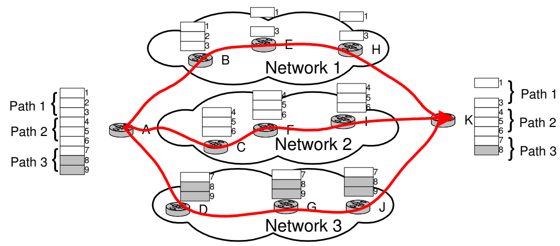


Figure 1: Path diversification.

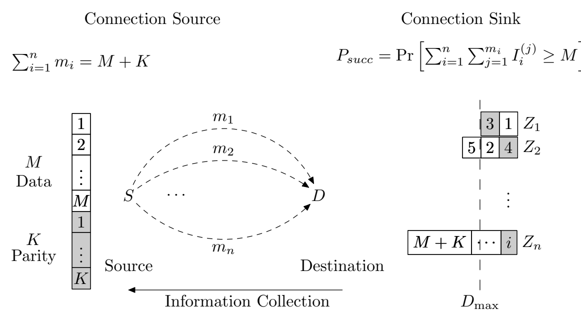


Figure 2: Mathematical model for path diversification

The network layer on the source node receives an Mb bit packet every $D_{\max}$ seconds. The network layer encodes each packet with the FEC code into $M + K$ fragments each of size b.[3] In Fig. 1, $M = 7$ and $K = 2$. The fragments are distributed on the paths and transmitted one-by-one to the destination. The destination listens to the paths and tries to reconstruct the packet from the fragments it receives in $D_{\max}$ seconds; some of the fragments may be lost before they arrive at the destination. For example, in Fig. 1, fragment 2 is lost on node B and fragment 9 is lost on the last hop from node J to the destination. However, the destination can still reconstruct the packet since it has received 7 fragments.

We show the mathematical model of path diversification in Fig. 2. Let us assume that n parallel and independent paths are available between the transmitter and the receiver. The $M + K$ fragments are subdivided into n non-overlapping sets with m_i fragments in each

last hop should also behave independently.

[3] There are many ways to implement an erasure code like this. For example, [1] uses integer modulus algebra while [29] uses the more efficient modulo-2 algebra. The actual implementation details are not important for this paper; we use erasure codes to increase reliability without duplicating transmitted information.

set. m_i is the number of fragments transmitted on path i. PPL is given as the probability that the destination receives less than M fragments; or equivalently that more than K fragments are lost. In the rest of the paper, we will use the probability of success P_{succ} in our optimizations instead of PPL. P_{succ} is the probability that at least M of the fragments are received successfully:

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

where $\mathcal{W}$ is a random variable indicating the total number of lost fragments, and $\mathcal{I}_i(j)$ is an indicator random variable corresponding to an unsuccessful transmission of fragment j on path i; that is, $\mathcal{I}_i(j) = 1$ if fragment j on path i is lost and $\mathcal{I}_i(j) = 0$ if the segment arrived at the destination within $D_{\max}$ seconds.

Path diversification introduces overhead in the network in terms of buffering costs and increased traffic, however this cost can be justified with the benefits of path diversification. The buffer overhead is comparable to the cost of IP fragmentation. The destination needs a buffer of Mb bits to hold the fragments before they are discarded. The traffic is increased both with the increase in the transmitted information and with the increased header cost. We measure the increase in the transmitted information with *efficiency* of path diversification:

$$\eta \triangleq \frac{\text{Effective Throughput}}{\text{Actual Throughput}} = \frac{M}{M+K}. \tag{2}$$

The total header overhead introduced with path diversification is $h(M + K)$, where we assume that header size in the network layer is h. In order to decrease the traffic cost, we maximize η, which decreases the total number of parity fragments K and the total number of fragments transmitted by the source. Nevertheless, the overhead can be justified by the decrease in PPL that we get with our scheme. As we showed in Section 1, a relatively small decrease in PPL substantially increases the TCP throughput.

2.1 Collection of Information

If the network status does not change very often, the source and the destination can collect information about the paths. This information can be used in the two optimization techniques that we will present in later sections. We show this in Fig. 2 as information collection moving in the opposite direction of traffic flow. We assume that the status of the paths is communicated to the source by a periodic feedback mechanism. This information can be transmitted to the source as a part of routing or as a separate probing mechanism. If path diversification is used in conjunction with routing, each node can collect its own statistics about fragment loss [30] and this information can be carried to the source as a part of routing information.

Alternatively, path diversification can be implemented separately from routing with an ingress probing technique similar to [31] or an egress probing technique such as [32] to collect statistical information at the source. Ingress probing is initiated by the source sending a series probing packets to the destination. The destination bounces all of the probes back to the source, allowing the source to find out the delay and packet loss statistics on a path. Egress probing, is a more passive technique in which the destination collects the delay and loss statistics of each data packet and occasionally sends information to the source. The network measurements are updated every T_{w} seconds. Ideally, this period is less than the network variation time and close to the maximum transmission time $D_{\max}$.

2.2 Wireless Model

We model wireless connections with a Markov chain model [33] with multiple "GOOD" and "BAD" states. A GOOD state corresponds to a high probability of successful segment transmission, $q_i(j) \triangleq \Pr[\mathcal{I}_i(j) = 0]$, and a BAD state corresponds to a high probability of failure $p_i(j) \triangleq 1 - q_i(j)$. If only two states are considered, we arrive at the Gilbert-Elliott model [34]. An alternative model uses multiple states and has been proposed in [35] to replace the Gilbert-Elliott model. In each state, we model the fragment losses as independent, meaning that $\mathcal{I}_i(j)$ are independent Bernoulli random variables. Nevertheless, we also consider the case where $\mathcal{I}_i(j)$ are dependent and show in Appendix A how to account for dependence in our optimizations.

This model is different from the model in [24, 25] where it is assumed that the $\mathcal{I}_i(j)$ on a path are fully dependent, i.e.:

$$\sum_{j=1}^{m_i} \mathcal{I}_i(j) = \begin{cases} m_i, & \text{with probability } q_i \\ 0, & \text{with probability } p_i. \end{cases} \tag{3}$$

The assumption (3) may be true if the fragments are lost only due to path disconnections. However, in this paper, we model the paths as highly unreliable wireless connections due to fast-fading and collisions in the physical layer. We also assume that there are no retransmissions in the link layer so that the reliability in the network is provided end-to-end with erasure coding.

3 Blind load balancing

In blind load balancing, no knowledge of the performance of parallel paths is available at the transmitter. Assume that there are M fragments offered by the connection and K parity fragments generated by the source node. In blind load balancing, the source deploys $\frac{M+K}{n}$ fragments on each path; for simplicity, we assume that $\frac{M+K}{n}$ is an integer number. In this section, we show that blind load balancing has a higher probability of success than the single-path transmission as $P_{\text{succ}} \to 1$. However, we also show that blind load balancing is suboptimal and motivate the need for optimum methods in Section 4.

We study blind load balancing in two extreme cases using the Markov chain models. In the first case, we assume that K is very small as compared to M. This case corresponds to a high value of efficiency. For a channel that is in a BAD state, most of the transmitted fragments will be lost. In a multiple-path transmission, since K is small, to reconstruct the original packet, we will need most of the fragments from each path to arrive at the destination.

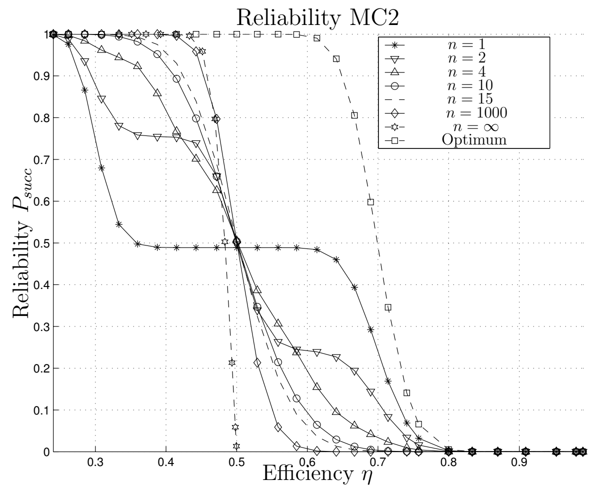


Figure 3: Performance of Blind Diversification

Intuitively, 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 fragments will succeed. Since the probability of having a GOOD state along a single path is higher than the probability of having simultaneous GOOD states 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 a fewer number of 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. Therefore, we can expect that for higher K, multipath transmission will perform better.

We have performed a simulation in which we transmit $N = 5000$ packets over a Gilbert-Elliott channel. The channel has a GOOD and a BAD state. In GOOD state, $q_i = 0.7$, and in BAD state $q_i = 0.3$, giving the average probability of error of $q_i = 0.5$. We have used $M = 100$; the value of K can be found from $K = \frac{1-\eta}{\eta}M$. The transitions from the

GOOD state to the BAD state form a Markov Chain. For simplicity, we assume that the channel only changes states between the transmissions of packets. We distribute the packets uniformly over the n paths. In Fig. 3, we show the likelihood of packet recovery, P_{succ}, as the number of paths increases for the full range of P_{succ} of $[0, 1.0]$. However, if we consider the performance of TCP over this channel, the region of interest is $[0.9, 1.0]$. It is clear that in the region of interest, the performance of the erasure code increases with the number of paths. Note that for a given $P_{\text{succ}} > 0.9$ the efficiency converges to a maximum value with increasing n the number of paths.

We now find maximum efficiency achievable with blind path diversification, as the number of paths n increases. We note that as n becomes large, the maximum number of fragments that will be sent on each path approaches 1 while $M + K \leq n$. So, $j = 1$ for all $\mathcal{I}_i(j)$ in (1) when $M + K \leq n$, and

$$P_{\text{succ}} \underset{n \to \infty}{=} \Pr\left[\sum_{i=1}^{M+K} \mathcal{I}_i \leq K\right], \tag{4}$$

where $\mathcal{I}_i$ are i.i.d. Bernoulli random variables. Here, we have assumed that the paths are independent and have identical statistical properties. For large $M + K \leq n$, we can use the Central Limit Theorem (C.L.T.) to further approximate P_{succ} as:

$$\begin{aligned} P_{\text{succ}} &\underset{M+K \to \infty}{\approx} \Phi\left(\frac{K - (M+K)p}{\sqrt{(M+K)pq}}\right) \\ &= \Phi\left(\sqrt{\frac{M}{\eta}} \frac{1 - \eta - p}{\sqrt{pq}}\right), \end{aligned} \tag{5}$$

where $\Phi(x) \triangleq 1/2\pi \int_{-\infty}^{x} e^{-t^2/2} dt$ is the cummulative distribution function of the standardized normal random variable and $\Pr[\mathcal{I}_i = 1] = p = 1 - q$. So, for a given P_{succ} the maximum η that can be achieved is given by the roots of:

$$\eta + \sqrt{\frac{pq}{M}} \Phi^{-1}(P_{\text{succ}}) \sqrt{\eta} + p - 1 = 0. \tag{6}$$

We show the performance of blind load balancing as $n \to \infty$ in Fig. 3. As a check, we also show the performance of path diversification for $n = 1000$ paths. It is clear from the figure that the C.L.T. approximates P_{succ} closely for large values of n and relatively large P_{succ}. From the C.L.T. approximation, we see that blind load balancing cannot achieve efficiency higher than $\eta = 0.5$. For $P_{\text{succ}} > 0.9$ and for single path transmission, the efficiency is $\eta = 0.28$. Increasing the number of paths to $n = 10$, increases the efficiency to $\eta = 0.42$, and increasing the number of paths to $n = \infty$ increases the efficiency to 0.47. Clearly, blind load balancing is better than single path transmission, however simply increasing the number of paths is not the best way to increase reliability in the network.

In Fig. 3, we also show a case in which at each time instant all fragments are transmitted on the path with the lowest probability of fragment loss p_i. This case has been denoted by "Optimum" in the graph. Note that transmitting the fragments in this way substantially increases the performance. The notable gain obtained for the optimum curve in Fig. 3 motivates us to investigate the problem for cases where a performance metric along each path is available at the source.

We also show the performance of multipath diversity in Fig. 4 for a five-state Markov Chain channel. In this model, the channel changes gradually between the GOOD and BAD states, and the states form a Markov Chain. Note that for this channel, we will also need to have multiple paths to achieve both high probability of success and acceptable efficiency.

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 that with this approach it is not possible to increase the efficiency for large values of reliability. We saw in the figure that if we could collect performance indexes along each path and use them to distribute fragments over multiple paths, we could have achieved much higher efficiency. In the rest of the paper, we assume that some performance metrics are available at the source node and use these metrics to devise optimum schemes for the distribution of fragments along parallel paths.

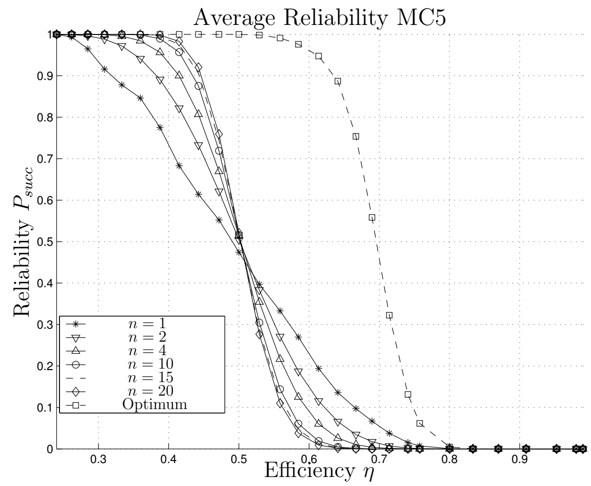


Figure 4: Performance of Blind Diversification

4 Optimum Load Balancing

We characterize QoS in the network in terms of delay and packet loss. We assume that the maximum delay in the network is fixed to $D_{\max}$ and that all fragments received after this time are considered to be lost. This allows us to characterize path's QoS behaviour in terms of just one parameter for each path in the network, namely fragment loss. We assume that the network has a mechanism that allows us to collect statistical information, such as path failure and delay statistics about the paths. The information allows us to improve on blind load balancing and minimize PPL.

The optimization is performed every T_{w} seconds, when a new vector of estimated packet loss statistics $\mathbf{q}$ becomes available. Ideally, T_{w} should be updated before every packet transmission. First, we show how P_{succ} can be calculated with an exact algorithm. The algorithm is valid when subsequent fragment losses are independent. Second, we use an approximation of P_{succ}, with the Poisson cumulative distribution function (c.d.f), which also allows us to account for the possible dependency between successive fragment losses. Third, we show that P_{succ} can be maximized with a "greedy" algorithm. In the ideal case, when the fragment losses are independent, both the Poisson approximation and the exact algorithm have the same solution.

We also give the maximization of efficiency of the scheme subject to minimum reliability. This optimization takes into account how the network operates. First, QoS of service parameters must be satisfied, this is the minimum reliability constraint. Second, the network optimizes the usage of resources with maximum efficiency.

In order to simplify the calculations in subsequent sections, we will assume that $p_i(j) = p_i$ and $q_i(j) = 1 - p_i$. We will use $\mathbf{q} = [q_1, q_2, \ldots, q_n]^T$ to denote the vector of q_i's. This assumption will allow us to express the optimization problems 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.

4.1 Calculation of P_{succ}

The calculation of the exact value for P_{succ} is directly related to the calculation of the reliability of algebraic structures, which in general is computationally hard [24]. However, we use the special properties of k-out-of-n structures to calculate P_{succ} [36]. The algorithm is based on the use of the moment generating function for the sum of $M + K$ independent Bernoulli random variables. The technique is similar to the direct calculation of the probability of failure of k-out-of-n structures with independent component reliabilities [37]. We model the transmission of fragments by the independent Bernoulli random variables $\mathcal{I}_i(j)$. By (1), the moment generating function for the number of lost packets, $\mathcal{W}$, is given by:

$$G_{\mathcal{W}}(z) = \prod_{i=1}^{n} (q_i + p_i z)^{M_i} = \sum_{i=0}^{M+K} c_i z^i. \tag{7}$$

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

$$P_{\text{succ}} = \Pr[\mathcal{W} \leq K] = \sum_{i=0}^{K} c_i \tag{8}$$

where each c_i is given by (7). Therefore, if the $q_i, i = 1, \ldots, n$ are known, the reliability can be calculated from (8) using $M + K$ recursions, one for every transmitted fragment. We use the following recursion to find the coefficients at each iteration, [37]:

$$c_j^{(k+1)} = q_{k+1} c_j^{(k)} + p_{k+1} c_{j-1}^{(k)}. \tag{9}$$

We now use the cumulative distribution function of the Poisson random variable to bound the reliability. P_{succ} can be approximated with the Poisson distribution, [38], as:

$$Q(\lambda(\mathbf{m}), K) \leq P_{\text{succ}} \leq Q(\lambda(\mathbf{m}), K) + \frac{1}{2} \sum_{i=1}^{n} m_i \ln^2(q_i) \tag{10}$$

where,

$$\begin{aligned} Q(\lambda(\mathbf{m}), K) &= \sum_{j=0}^{K} \frac{e^{-\lambda(\mathbf{m})}[\lambda(\mathbf{m})]^j}{j!} \\ \lambda(\mathbf{m}) &= \sum_{i=1}^{n} \ln(q_i^{-m_i}) = -\mathbf{m}^T \ln(\mathbf{q}) \end{aligned} \tag{11}$$

and $\ln(\mathbf{q}) = [\ln(q_1), \ln(q_2), ..., \ln(q_n)]^T$ is the natural logarithm of the vector of probabilities of success. The Poisson approximation is a good replacement for the exact algorithm 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 $10! > 3 \times 10^6$).

The other advantage of the Poisson approximation is that we can model the dependence of successive fragment losses. So far, we have approximated the consecutive packet losses in the time period T_{w} as independent in order to calculate the probability of successful packet transmission P_{succ}. Using the Poisson approximation (11), it is possible to find the estimates for q_i that minimize the error of approximating the dependent variables $\mathcal{I}_i(j)$ as independent ([38] gives one such estimate). However, we use a different approach by expressing the error in dependance on each path with:

$$Q(\lambda(\mathbf{m}), K) - 2\sum_{i=1}^{n}\sum_{j=2}^{m_i} p_i \leq P_{\text{succ}}. \tag{12}$$

We derive the relation in Appendix A.

We make two observations about (12). First, the error in approximation depends on the total number of transmitted fragments $M + K$, meaning that the load balancing algorithm should be robust if $M + K$ is relatively small. Second, the error is 0 if each path only carries a single fragment. This means that the load balancing is more robust to the dependance of fragment transmissions if the number of fragments transmitted on each path is small. The second observation can be used in the two optimizations in this paper by limiting the number of fragments transmitted on each path.

4.2 Minimum PPL (Maximum P_{succ})

In this section, we give an algorithm that finds the allocation vector $\mathbf{m} = [m_1, m_2, .., m_n]^T$ for which P_{succ} is maximized. We note that P_{succ} is a monotonically increasing function of K. This is clear from (1) since P_{succ} is defined as the c.d.f of the random variable $\mathcal{W}$. So, 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 fragments that can be transmitted on each path. The optimization is given as:

$$\underset{\mathbf{m}}{\text{Maximize:}} \; P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) \tag{13a}$$

$$\text{Subject to:} \; \mathbf{m}^T\mathbf{1} = K + M \tag{13b}$$

$$\mathbf{0} \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \tag{13c}$$

where $\mathbf{1}$ is a vector of all 1's, $\mathbf{0}$ is a vector of all 0's, $\mathbf{M}_{\text{th}} = [M_1^{(th)}, M_2^{(th)}, \ldots, M_n^{(th)}]$ is the maximum number of fragments that can be transmitted on each path, and $\preceq$ indicates a memberwise comparison.

Constraint (13b) assures that the total number of fragments on the paths is $M+K$. The second constraint (13c) limits the number of fragments on each path. Note that increasing the number of fragments transmitted on a path also increases the delay and congestion on that path. Therefore, we assume that the maximum number of fragments transmitted on each path is limited, where the limit is determined either through statistical properties of the end-to-end delay determined with our probing mechanism or by letting each node locally determine the number of fragments it can transmit and then taking the minimum of these numbers on every path. We also assume that $\mathbf{M}_{\text{th}}$ is kept small to make the algorithm more robust to the dependence of fragment losses on each path. The optimization (13) can be solved with a "greedy" algorithm. We give a proof for the correctness of this algorithm in [39].

In comparison to the solution with the greedy algorithm, the Poisson approximation has the same optimum allocation vector. If we use the Poisson approximation for P_{succ}, optimization (13) becomes:

$$\underset{\mathbf{m}}{\text{Maximize:}}\ Q(\mathbf{m}, K) \tag{14a}$$

$$\text{Subject to: } \mathbf{m}^T\mathbf{1} = K + M \tag{14b}$$

$$0 \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}}. \tag{14c}$$

It can easily be shown that $Q(\lambda(\mathbf{m}), K)$ is a decreasing function of $\lambda(\mathbf{m})$ for a fixed K.[4] This means that we can maximize the probability of success by minimizing $\lambda(\mathbf{m})$. The

[4]Taking the derivative of $Q(\lambda(\mathbf{m}), K)$ with respect to each m_i gives:

$$\frac{\partial Q}{\partial m_i} = -\frac{\partial \lambda(\mathbf{m})}{\partial m_i} e^{-\lambda(\mathbf{m})} \frac{[\lambda(\mathbf{m})]^K}{K!} < 0, \tag{15}$$

since $\lambda(\mathbf{m})$ is a positive function.

optimization then becomes:

$$\underset{\mathbf{m}}{\text{Maximize:}} \ \mathbf{m}^T \ln(\mathbf{q}) \tag{16a}$$

$$\text{Subject to: } \mathbf{m}^T \mathbf{1} = K + M \tag{16b}$$

$$\mathbf{0} \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \tag{16c}$$

Optimization (16) can be solved exactly for $\mathbf{m} \in \mathbb{R}^n$ with the use of greedy algorithm 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.

In [25], the optimization of P_{succ} was performed by approximating P_{succ} as the normal c.d.f with CLT. This approximation is more appropriate with the complete dependence of fragment losses on each path. With the Normal approximation, one may also interpret the objective function of [25] in terms of path failure. However, there are still two problems with this approach. First, CLT approximates P_{succ} only when there are a large number of paths. Since P_{succ} in [25] is the c.d.f of a sum of n integer random variables the approximation becomes increasingly better with larger n. Second, it is difficult to know how good the normal approximation is since the best known bound for it is the Berry-Essen bound; this bound is loose for small n [40].

We use an example to further illustrate the differences between our scheme and that of [24, 25]. Suppose, there are three paths available to transmit fragments between the source and the destination with $\mathbf{q} = [0.9, 0.8, 0.8]$ and $M = 2$ and $K = 1$. With the assumption of independent fragment loss, we assign all three fragments to path 1 to get $P_{\text{succ}} = 0.95$. The algorithm in [25] assigns each fragment on a separate path to get $P_{\text{succ}} = 0.928$. On the other hand, if the fragment losses on each path are fully dependent, as assumed in [25], our allocation results in $P_{\text{succ}} = 0.9$ and the allocation with the algorithm from [25] results in $P_{\text{succ}} = 0.928$. The normal approximation evaluates P_{succ} for the allocation with the algorithm in [25] as 0.999, for both the independent fragment allocation and the dependent fragment allocation. The normal approximation calculates that our allocation evaluates P_{succ} as 0.9927, which is also incorrect.

4.3 Maximization of Efficiency (η)

In this section, we show how the efficiency η can be maximized. As we explained earlier in Section 2, η is important since it is directly related to the amount of overhead introduced by the scheme. We perform the optimization with a constraint on the minimum network reliability $P_{\text{succ}} \geq \epsilon$ where ϵ is a QoS parameter supplied by the connection. The efficiency should be maximized to decrease the cost of the connection, however the packet loss should be bounded to guarantee QoS to the connection.

The optimization problem is given by:

$$\underset{\mathbf{m},K}{\text{Maximize:}} \ \eta(\mathbf{m}, K) = \frac{M}{M+K} \tag{17a}$$

$$\text{Subject to: } P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) \geq \epsilon \tag{17b}$$

$$\mathbf{m}^T \mathbf{1} = M + K \tag{17c}$$

$$\mathbf{0} \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \tag{17d}$$

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:

$$\underset{K}{\text{Minimize:}} \ K \tag{18a}$$

$$\text{Subject to: } P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) \geq \epsilon \tag{18b}$$

$$\mathbf{m}^T \mathbf{1} = M + K \tag{18c}$$

$$\mathbf{0} \preceq \mathbf{m} \preceq \mathbf{M}_{\text{th}} \tag{18d}$$

We assume that $\eta(\mathbf{m}, K) \geq \delta$, where δ is a parameter used to put a bound on the complexity of the optimization. So, the maximum number of parity checks that can be used is

$$K_{\max} = \frac{1-\delta}{\delta} M. \tag{19}$$

We can perform the optimization (18) with a simple linear search over at most $K_{\max}$ items. Algorithm 1 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 way. The algorithm uses the function MAXIMIZE-PSUCC (note shown here in the interest of space) to allocate the fragments for maximum P_{succ} with the optimization (13). The algorithm may not find a solution for K in $0 \le K \le K_{\max}$. In such a case, we declare that the packet cannot be transmitted in the network. This case corresponds to the probability of blocking.

The algorithm checks the validity of the solution by evaluating P_{succ} using the methods proposed in Section 4. Here, we note that more efficient implementations of Algorithm 1 can be obtained if we use the recursive relationship (9). We will not discuss these techniques here. We show the correctness of the algorithm in [39].

5 Numerical Results

We created several simulation scenarios to examine the performance of path diversification. First, we show the results for unconstrained optimization, which gives the optimal performance for independent fragment losses. Second, we limit the maximum number of fragments that can be transmitted on each path $\mathbf{M}_{\text{th}}$ to account for the delay and dependence between fragments. We examine the impact of the constraints on the efficiency of the load balancing algorithm. Third, we show that the robustness of the algorithm when q_i are not accurate.

Algorithm 1 Optimize-Utilization$(\mathbf{M}_{th}, \mathbf{q}, \epsilon, \delta)$

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

$M_{\text{total}} = 0$, $\mathbf{m} = \mathbf{0}$
$\mathbf{m} \leftarrow$ Maximize-Psucc$(\mathbf{q}, M, 0, \mathbf{M}_{th})$
$i \leftarrow \min\{\forall j : 1 \leq j \leq n, M_j \neq 0\} - 1$
$K_{\max} \leftarrow \lfloor \frac{1-\delta}{\delta} M \rfloor$
repeat
    if $i < n \wedge K \leq K_{\max} \wedge M_i < M_i^{(th)}$ **then**
        $K \leftarrow K + 1$
        $M_i \leftarrow M_i + 1$
    else
        $i \leftarrow i + 1$
    end if
until $P_{\text{succ}}(\mathbf{m}, K, \mathbf{q}) < \epsilon$
if $i > n \vee K > K_{\max}$ **then**
    return $\mathbf{m} \leftarrow \emptyset$
else
    return $\mathbf{m} \leftarrow \{M_1, M_2, .., M_n\}$
end if

Fourth, we give results for the maximization of efficiency.

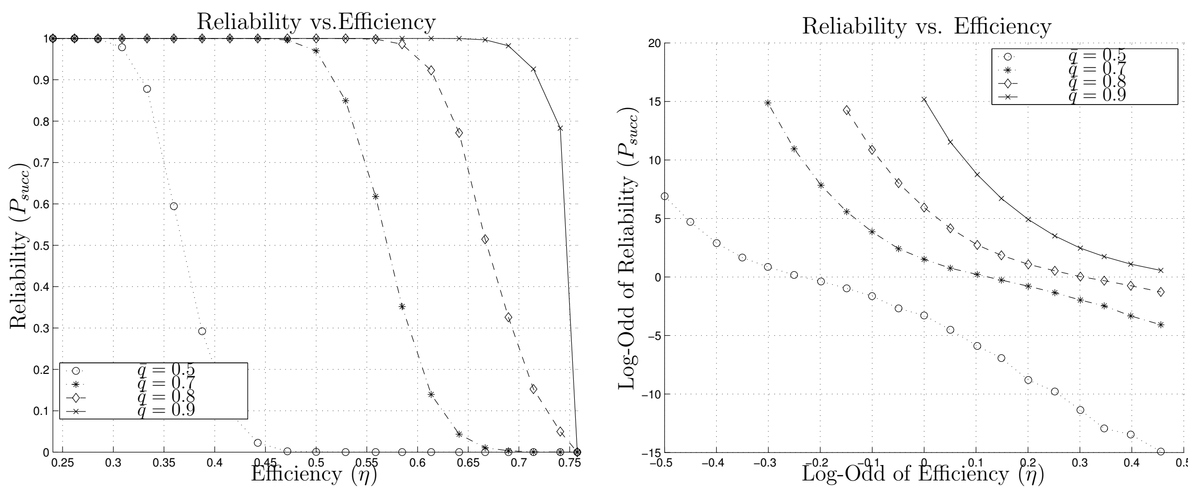
Table 1: The Log-Normal Transformation

$\log_{10}\left(\frac{\alpha}{1-\alpha}\right)$	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

5.1 Unconstrained Maximization of P_{succ}

We evaluate the unconstrained maximization of P_{succ}. In the unconstrained optimization (13), constraint (13c) is not taken into account (i.e., $\mathbf{M}_{\text{th}} = \infty$). We simulate 200 packet transmissions for $n = 5$ paths. The q_i's are selected before each transmission from the uniform distribution $[\bar{q}-0.1, \bar{q}+0.1]$ where the average probability of fragment success (PFS) is $\bar{q} = 0.5, 0.7, 0.8, 0.9$. Fig. 5a shows η on the horizontal axis and the average value for P_{succ} on the vertical axis. Path diversification is an efficient way to ensure network reliability. For example, for $\bar{q} = 0.8$, we can achieve the reliability of 0.999 with the efficiency of 57%.

In Fig. 5b, we show the same plot of P_{succ} as in Fig. 5a in the log-odd scale. The values in the log-odd scale are plotted as $\log(\alpha/(1-\alpha))$ 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 observe the asymptotic effect of $P_{\text{succ}} \to 0$ or $P_{\text{succ}} \to 1$. For example, a value of $P_{\text{succ}} = 0.999$, would translate to a value of 3 in the log-odd scale. We show some other mappings to the log-odd scale in Table 1.



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

Figure 5: Reliability vs. Efficiency

Our work differs from [25] in two aspects. First, our load balancing algorithm is different from the load balancing algorithm in [25] and second we calculate P_{succ} differently from [25], giving us a more accurate way to validate the performance of either load balancing algorithm. Here, we compare our work with [25]. In our simulations, the source transmits 200 packets each decomposed into $M = 100$ fragments over $n = 5$ paths, with the average PFS of $\bar{q} = 0.8$.

We show the results of the simulations in Fig. 6. The "Optimum Solution" and the

"Optimum Poisson" curves, show the performance of our load balancing algorithm. We used the exact algorithm to calculate P_{succ} for the "Optimum Solution" curve and the Poisson approximation (10) to calculate P_{succ} for the "Optimum Poisson" curve. The Poisson approximation gives a very good estimate of P_{succ}. So, the Poisson approximation is a good candidate to replace the exact algorithm 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 $Q(\lambda(\mathbf{m}), K)$ with precision of less than 10^{-6}.

In Fig. 6, the "Normal Approximation" curve shows the performance of the load balancing scheme in [25] evaluated with the normal c.d.f., which was used in [25] to evaluate P_{succ}. We can see that the approximated values of P_{succ} in the "Normal Approximation" curve are inaccurate. The "Optimum Normal" curve shows the performance of the load balancing scheme in [25] evaluated with the exact algorithm for the evaluation of P_{succ} that was proposed in this paper. We see that the transmission of packets using our load balancing scheme outperforms the load balancing in [25].

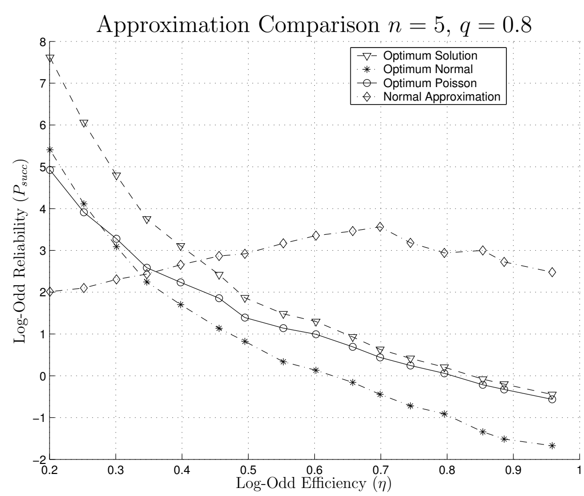


Figure 6: Comparison to the Normal and Poisson Approximation $n = 5$, $\bar{q} = 0.8$

5.2 Constrained Optimization of $P_{\mathbf{succ}}$

Ideally, the source should distribute all the fragments on the path with the highest PFS. However, this is only possible without the constraint (13c) in optimization (13), that is $\mathbf{M}_{\text{th}} = \infty$. In this section, we assume that $\mathbf{M}_{\text{th}}$ on each path is limited.

We perform a simulation with 1000 packets transmitted on $n = 5$ paths with a fixed PFS on each path and $\mathbf{M}_{\text{th}}$ given by a Poisson distribution with the parameter μ. The PFS was fixed to $\mathbf{q} = [0.85, 0.7, 0.7, 0.7, 0.7]$; these are the same values used in [23]. Without the upper bound on $\mathbf{M}_{\text{th}}$, the optimum strategy is to distribute all the fragments on path 1 with $q_i = 0.85$. However, the upper bound on $\mathbf{M}_{\text{th}}$ forces the source to distribute the fragments on more than one path.

Fig. 7 demonstrates the effect of the upper bound $\mathbf{M}_{\text{th}}$ on P_{succ}. We plot P_{succ} for different values of the Poisson parameter μ. We see, from the figure, that limiting the number of fragments on each path makes the load balancing algorithm less efficient. For example, for $P_{\text{succ}} = 0.999$, the efficiency decreases 18% when the source uses 3 paths on average, $\mu/M = 0.3$, and 15% when the source uses 2 paths on average, $\mu/M = 0.5$. However, this also means that if the transmission must be distributed among many paths due to constraints on each path, the source can still increase reliability by increasing K.

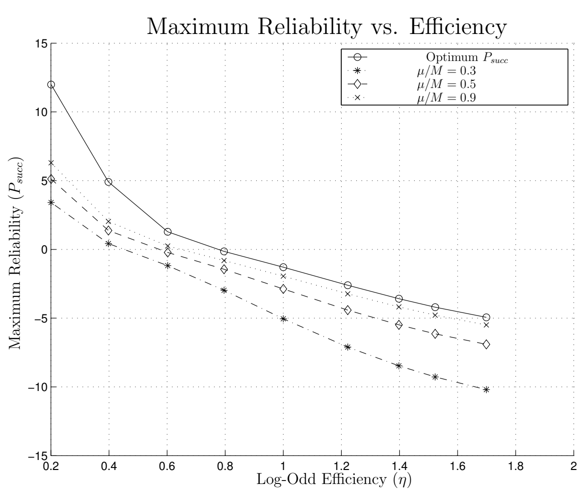


Figure 7: QoS Constrained Reliability vs. Efficiency

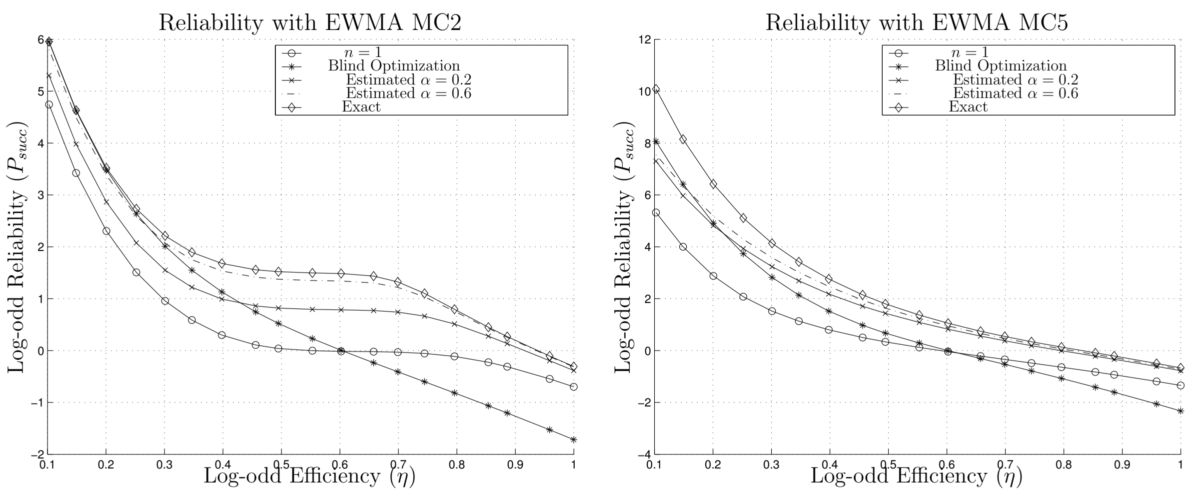
5.3 Robustness of the Optimization of P_{succ}

Previously, we assumed that the system can accurately estimate the current PFS on each path. In order to test the effect of inaccurate PFS on the optimization (13), we implemented an estimator for the fragment loss on each path using the Exponentially Weighted Moving Average (EWMA) estimator [41]. The EWMA estimator uses the following relation to calculate the current estimate $\hat{q}_i^{\text{new}}$, from the previous estimate $\hat{q}_i^{\text{old}}$:

$$\hat{q}_i^{\text{new}} = \hat{q}_i^{\text{old}} + \alpha(q_i - \hat{q}_i^{\text{old}}) \tag{20}$$

where q_i is the latest estimate received from the network. Coefficient α dictates the influence of old samples on the current estimate; a smaller α means that the old samples have a larger impact on the current estimate.

Here, we simulate the fragment losses as a Markov chain, where the states correspond to the probability of successful fragment transmission q_i. For simplicity, we assume that the state changes at every packet transmission. We ran the simulation for $N = 5000$ packets.



(a) Two State Markov Channel

(b) Five State Markov Channel

Figure 8: Robustness of the Load Balancing Algorithm

Fig. 8a shows P_{succ} when an EWMA estimator is used over a two state Markov chain, where q_i goes through the states $[0.7, 0.9]$ to give an average PFS of $\bar{q} = 0.8$. We perform the experiment for $\alpha = 0.2$ and $\alpha = 0.6$. The performance of the estimator for $\alpha = 0.6$ is very close to that of the optimum since the estimates of $q_i(j)$ are more accurate. However, the estimator does not perform as well with $\alpha = 0.2$ since it emphasizes the old values of $q_i(j)$ on the fast varying channel. Fig. 8b shows P_{succ} for a five state Markov chain where q_i goes through the states $[0.7, 0.75, 0.8, 0.85, 0.9]$. In this case, both estimators have similar performance, but they are not as accurate as in the case of the two state Markov chain.

The performance of the load balancing is not sensitive to inaccurate estimates of q_i. For example, for the two state Markov chain scenario the estimate with $\alpha = 0.2$ is only 2% less efficient than the optimum solution for $P_{\text{succ}} = 0.999$. The reduction in efficiency for the same value of P_{succ} in the five state Markov chain scenario scenario is 5%.

Fig. 8 also shows the performance of the blind load balancing algorithm and the single path solution. We note that the load balancing performs better than the single path solution, regardless of the accuracy of q_is. The load balancing algorithm also performs better than the blind load balancing algorithm for moderate values of P_{succ}. For example, for the five state Markov chain scenario blind path diversification performs better than the load balancing algorithm for $P_{\text{succ}} > 0.99999$. However, for more practical values of reliability $0.99 \leq P_{\text{succ}} \leq 0.999$ the load balancing algorithm outperforms the blind load balancing algorithm, even with inaccurate values of q_i.

5.4 Optimization of Efficiency

In this section, we examine the efficiency of the load balancing algorithm when the source uses optimization (17). We perform simulations for a scenario with $n = 5$ paths where we have set $M = 100$. The source transmits $40,000$ packets in which we fix the PFS of each path for 200 transmissions and then we choose another set of PFSs for the next 200 transmissions and so on; the average PFS for all transmissions is $\bar{q} = 0.8$. We perform 200 sets of 200 packet transmissions. For each set of 200 packet transmissions, we select the upper bound

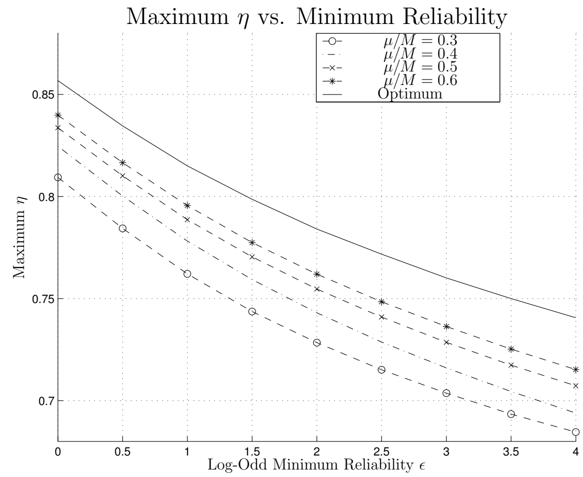


Figure 9: Minimum Reliability vs. Maximum Efficiency

on the number of fragments on each path from a Poisson distribution with the parameter μ.

Fig. 9 shows the maximum efficiency that can be achieved for different values of minimum reliability ϵ. The "Optimum" curve is obtained by examining the unconstrained optimization (17), i.e. $\mathbf{M}_{\text{th}} = \infty$. This case corresponds to the maximum efficiency that can be achieved with path diversification. We note that for the minimum reliability $\epsilon = 0.999$ the efficiency is $\eta = 0.76$. This means that the overhead of path diversification with optimization (17) is 59% lower than the overhead of the optimization (13).[5]

We also see that the effect of using multiple paths is mitigated with the optimization. For example, if the source is forced to use 2 paths on average ($\mu/M = 0.2$) the efficiency decreases 4%; this is better than the optimization (13) where the reduction in efficiency was 15%. We also see from the figure that it will be reasonable to expect the efficiency in the network to be $\eta > 0.70$ if the source does not need to use more than 3 paths.

[5]We have shown in Section 5.1 that $\eta = 0.57$ for $P_{\text{succ}} = 0.999$, when the source uses optimization (13).

6 Conclusions

In this paper, we have proposed a new approach to increase reliability in wireless networks. The proposed technique uses multipath routing complemented with erasure codes. The technique can achieve high reliability when no information is known about the network performance. However, we have shown that by collecting information about network behaviour, we can achieve high reliability efficiently and without using a high number of paths in the network.

First, we have given a polynomial time algorithm to find the path allocation that minimizes PPL. We have used numerical simulations to illustrate the effectiveness of this technique. The simulations have shown that the algorithm is robust to delay constraints on the paths and inaccurate information about the network performance. We have also compared the performance of our algorithm with the load balancing algorithm in [25]. We have shown that our algorithm performs better than the algorithm in [25].

Second, we have given a polynomial time algorithm to find the maximum efficiency of the scheme for a given maximum allowed PPL. This algorithm is necessary to allow for provisioning of QoS in the network. We have shown through simulations that this algorithm achieves better efficiency than the first algorithm.

Appendices

A Lower Bound with Dependent Packet Losses (Proof of (12))

We examine the impact of fragment loss dependence on the lower bound for reliability in (10). We use the indicator random variables $\mathcal{I}_i^*(j)$ to represent the true loss on the path including the arbitrary dependence between consecutive losses. Note that $\mathcal{I}_i^*(j)$ is a random variable indicating the loss of segment j on path i and $\Pr[\mathcal{I}_i^*(j) = 1] = p_i^*(j) = 1 - q_i^*(j)$. This is analogous to the random variables $\mathcal{I}_i(j)$ in (1). The difference is that $\mathcal{I}_i(j)$ random variables approximate the losses as independent. In the rest of this section, we will find the error in approximating the sum of $\mathcal{I}_i^*(j)$'s with $\mathcal{I}_i(j)$'s and show the change that this approximation introduces in (10).

We define the dependence of fragment loss on path i with a sequence of random variables $\theta_i(j)$:

$$\begin{aligned} \theta_i(1) &= \Pr[\mathcal{I}_i^*(1) = 1] = p_i^*(1) \\ \theta_i(j) &= \Pr[\mathcal{I}_i^*(j) = 1 | \mathcal{F}_i(j-1)] \end{aligned} \tag{21}$$

where $\mathcal{F}_i(j-1)$ is a σ-algebra on the set $\Omega_i(j-1) = \{\mathcal{I}_i^*(1), .., \mathcal{I}_i^*(j-1)\}$. This makes $\theta_i(j)$ a random variable on the probability space $(\Omega_i(j-1), \mathcal{F}_i(j-1), P_{\theta_i(j)})$, where $P_{\theta_i(j)}$ gives the probability that a specific sequence of losses and successes precedes the jth transmission.

The random variables $\theta_i(j)$ allows us to use the results of [38] and change the lower bound on P_{succ} in (10) to include the dependence of fragment losses, as follows:

$$Q(\lambda(\mathbf{m}), K) - \sum_{i=1}^{n} \sum_{j=1}^{m_i} \mathbf{E}|\theta_i(j) - p_i(j)| \leq P_{\text{succ}}. \tag{22}$$

The extra term is the error from approximating the dependent random variables $\mathcal{I}_i^*(j)$ with arbitrary independent random variables $\mathcal{I}_i(j)$. It is shown in [38] that $p_i(j) = \Pr[\mathcal{I}_i(j) = 1]$ can be chosen arbitrarily. We approximated these values with the average value of $p_i^*(j)$ over all transmissions, p_i, as follows:

$$p_i = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathbf{E} p_i^*(j). \tag{23}$$

The optimum value in terms of dependence would be to use the median of each $\theta_i(j)$. However, this is not important for this discussion.

In order, to give a better idea of what the lower bound may actually be, we now calculate the expected value on the summation in (22), conditional on a fixed value of $p_i^*(j)$. Recall that $p_i^*(j)$ is a random process, so this means that we are only looking at a single realization of the process. We can calculate the expected value of $\mathbf{E}|\theta_i(j) - p_i|$, for $m_i > 1$ with:

$$\begin{aligned}\mathbf{E}|\theta_i(j) - p_i| &= \int\limits_{\Omega_i(j-1)} |\theta_i(j) - p_i| \, dP_{\theta_i(j)} \\ &\leq \int\limits_{\Omega_i(j-1)} \theta_i(j) dP_{\theta_i(j)} + \int\limits_{\Omega_i(j-1)} p_i dP_{\theta_i(j)} \\ &= \Pr[\mathcal{I}_i^*(j) = 1] + p_i \\ &= p_i^*(j) + p_i.\end{aligned} \tag{24}$$

Note that for $m_i = 1$, $\mathbf{E}|\theta_i(j) - p_i| = 0$.

If we take the expected value of $\mathbf{E}|\theta_i(j) - p_i(j)|$ over $p_i^*(j)$, sum up the results for all the fragments on all the paths, and combine this with (23), (22) becomes:

$$Q(\lambda(\mathbf{m}), K) - 2\sum_{i=1}^{n}\sum_{j=2}^{m_i} p_i \leq P_{\text{succ}}. \tag{25}$$

References

[1] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *Journal of the Association for Computing Machinery*, vol. 36, no. 2, pp. 335–348, April 1989.

[2] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee, "MAR: a commuter router infrastructure for the mobile internet," in *MobiSys*, 2004, pp. 217–230.

[3] Nortel Networks, "Wireless mesh network - extending the reach of wireless lan, securely and cost-effectively," http://www.nortelnetworks.com/solutions/wlan/, November, 2003.

[4] Mesh Networks, "Putting hot spots in motion mesh-enabled solutions mobile 802.11," http://www.meshnetworks.com/pages/applications/mobile_wifi.htm, Retrieved September 5, 2004.

[5] P. Djukic and S. Valaee, "Minimum energy fault tolerant sensor networks," in *Globecom*, 2004.

[6] Mesh Networks, "White paper: Mesh-enabled solutions for intelligent transportation systems," Tech. Rep., 2004.

[7] T. Goff and D. S. Phatak, "Unified transport layer support for data striping and host mobility," *IEEE J. Select. Areas Commun.*, vol. 22, no. 4, pp. 737–746, May 2004.

[8] 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.

[9] T. Fowler, "Mesh networks for broadband access," *IEE Review*, vol. 47, no. 1, pp. 17–22, January 2001.

[10] B. Schrick and M. J. Riezenman, "Wireless broadband in a box," *IEEE Spectr.*, vol. 39, no. 6, pp. 39–43, June 2001.

[11] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks," *IEEE Commun. Mag.*, vol. 39, no. 6, pp. 130–137, June 2001.

[12] R. Hsieh, A. Seneviratne, H. Soliman, and K. El-Malki, "Performance analysis on hierarchical mobile ipv6 with fast-handoff over end-to-end TCP," in *In Proceedings of GLOBECOM*, Taipei, Taiwan, 2002, pp. 2488–2492.

[13] S. Valaee, "Information raining for mobile hotspots in 4G wireless networks," in *22nd Biennial Symposium on Communications*, Kingston, Ontario, May 31 - June 3 2004.

[14] D. Ho and S. Valaee, "Mobile hotspot in railway system," in *22nd Biennial Symposium on Communications*, Kingston, Ontario, May 31 - June 3 2004.

[15] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of SIGCOMM'98*, 1998, pp. 303–314.

[16] M. Stoer, *Design of Survivable Networks*, ser. Lecture Notes in Mathematics. Heidelberg: Springer Verlag, 1992, vol. 1531.

[17] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proceedings of the International Conference for Network Procotols (ICNP)*, Riverside, November 2001.

[18] A. Nasipuri, R. Castaneda, and S. R. Das, "Performance of multipath routing for on-demand protocols in ad hoc networks," *ACM/Kluwer Mobile Networks and Applications (MONET) Journal*, vol. 6, no. 4, pp. 339–349, 2001.

[19] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path selection in mobile ad hoc networks," in *Proceedings of Mobihoc 2002.* ACM, 2002.

[20] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *ICC*, vol. 10, 2001, pp. 3201 – 3205.

[21] H. Lim, K. Xu, and M. Gerla, "TCP performance over multipath routing in mobile ad hoc networks," in *ICC*, vol. 2, 2003, pp. 1064 – 1068.

[22] S. Vutukury, "Multipath routing mechanisms for traffic engineering and quality of service in the internet," Ph.D. dissertation, Univesity of California, Santa Cruz, May 2001.

[23] A. Tsirigos and Z. J. Haas, "Multipath routing in the presence of frequent topological changes," *IEEE Commun. Mag.*, vol. 39, no. 11, pp. 132–138, November 2001.

[24] ——, "Analysis of multipath routing-part I: The effect on the packet delivery ratio," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 138–146, January 2004.

[25] ——, "Analysis of multipath routing, part 2: Mitigation of the effects of frequently changing network topologies," *IEEE Trans. Wireless Commun.*, vol. 3, no. 2, pp. 500–511, March 2004.

[26] G.-S. Ahn, A. T. Campbell, A. Veres, and L.-H. Sun, "SWAN: Service differentiation in stateless wireless ad hoc networks," in *INFOCOM 2002*, 2002.

[27] S.-B. Lee, G.-S. Ahn, and A. T. Campbell, "Improving UDP and TCP performance in mobile ad hoc netoworks with INSIGNIA," *IEEE Commun. Mag.*, vol. 39, no. 6, pp. 156–165, June 2001.

[28] N. F. Maxemchuk, "Dispersity routing," in *Proceedings of IEEE Internationl Communications Conferenece ICC'75*, Sna Francisco, CA, June 1975, pp. 41.10–41.13.

[29] J. Blomer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman, "An XOR-based erasure-resilient coding scheme," Berkley, CA, Technical Report TR-95-048, 1995.

[30] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges for reliable multihop routing in sensor networks," in *SenSys'03*. ACM Press, 2003.

[31] S. Valaee and B. Li, "Distributed call admission control for ad hoc networks," in *Proceedings of Vehicular Technology Conference (VTC)*. IEEE, September 2002.

[32] C. Cetinkaya and E. 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*, March 2000, pp. 1471–1481.

[33] C. Tang and P. K. McKinley, "Modeling multicast packet losses in wireless lans," Computer Science and Engineering Department, Michigan State University, Technical Report MSU-CSE-02-12, 2002.

[34] M. Mushkin and I. Bar-David, "Capacity and coding for the gilbert-elliott channels," *IEEE Trans. Inform. Theory*, vol. 35, no. 6, pp. 1277–1290, November 1989.

[35] H. S. Wang and N. Moayeri, "Finite-state Markov channel - a useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–171, February 1995.

[36] D. R. Shier, *Network Reliability and Algebraic Structures.* Oxford: Claredon Press, 1991.

[37] R. E. Barlow and K. D. Heidtmann, "Computing k-out-of-n system reliability," *IEEE Trans. Rel.*, vol. R-33, no. 4, pp. 322–323, October 1984.

[38] R. J. Serfling, "Some elementary results on Poisson approximation in a sequence of Bernoulli trials," *SIAM Review*, vol. 20, no. 3, pp. 567–579, July 1978.

[39] P. Djukic, "Optimum resource allocation in multipath ad hoc networks," M.A.Sc. Thesis, University of Toronto, 2003.

[40] Y. V. Prokhorov and V. Satulevicius, Eds., *Limit Theorems of Probability Theory.* New York: Springer-Verlag, 2000.

[41] J. F. Muth, "Optimal properties of exponentially weighted forecasts," *Journal of the American Statistical Association*, vol. 55, no. 290, pp. 299–306, June 1960.