Optimizing Redundancy Using MDS Codes and Dynamic Symbol Allocation in Mobile Ad Hoc Networks

Anna Kacewicz
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853
Email: ak387@cornell.edu

Stephen B. Wicker
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853
Email: wicker@ece.cornell.edu

***Abstract*—We consider the design of a network in which packet transmissions between two nodes are to be guaranteed a certain probability of success. There are N independent paths between the two nodes, each path having a fixed probability of success (perhaps representing an a priori estimate of the security of the path). The original message is encoded using a maximum-distance separable code. We find the minimum redundancy and optimal path symbol allocation for the 3-path case based on a desired success probability and the path success probabilities. Two algorithms are presented for determining the minimum required redundancy – an exponential time algorithm (MRAET) and a polynomial time algorithm (MRAPT). We show that MRAET is optimal in several cases, and that both algorithms perform very well.**

***Index Terms*—Error Correction Coding, Networks, Redundancy**

I. Introduction

In this paper we consider a scenario in which there are multiple independent paths between a source-destination pair through which information can be transmitted. We further assume that some nodes along these paths may fail to co-operate, leading to lost messages. One way to correct for lost data is to add redundancy to the original message. As unnecessary redundancy results in lower network efficiency, it follows that some means must be found for estimation of the minimum necessary redundancy. In [2], the authors introduce a method of secure message transmission over a multipath channel. They suggest a technique which assesses the integrity of each available path set between the source and destination. The degree of trustworthiness of a path is associated with the probability of successful transmission. The paths that are determined to be secure, or the active path set (APS), are used to transmit the message.

Mobile Ad Hoc Networks are wireless networks in which topology is constantly changing due to link failures or wireless channel conditions. Changing topology renders pre-determined routing schemes inadequate, since a previously good link might become inconsequential because of severe fading or compromised nodes. Thus, it is important to determine the reliability of each link, and further, how to dynamically distribute data across the network. Our scheme is amenable to these networks, since it considers the integrity of each link and then allocates information accordingly.

The multipath routing literature is extensive. In [6], the authors suggest a protocol for finding maximally disjoint paths. They divide the message among the paths, though do not consider different security levels in the paths. In [3], the authors develop a multipath routing technique for the case where the paths have equal failure probability. Given the number of paths, they try to maximize the probability of success by optimally allocating channel coded packets to the paths. In [4], the authors extend their previous work to the case where the path failure probabilities are all different. They attempt to ameliorate the problem of a complicated probability of success function by deriving an approximation, and then determine the optimal block allocation to each path in order to maximize the approximated function. In all of this work, the encoded data is transmitted amongst the paths assuming that the desired level of redundancy is known. In this paper we attempt to find the minimum redundancy and optimal symbol allocation to meet a minimum required success probability.

In [2] the authors use a redundancy coding technique based on Rabin's algorithm [5], which acts like an erasure code. This adds redundant symbols to the message, allowing proper decoding even if the encoded message is not fully received at the destination node. To alleviate lost data caused by a malevolent node, we consider the use of maximum-distance separable (MDS) codes, which include the well-known Reed-Solomon codes [9]. MDS codes are notable in this context, as any k symbols in an (n, k) codeword can be used to recover the original k- symbol message.

We consider a technique in which the symbols in a message are encoded using an MDS code. Then the symbols in the given codeword are broken up and allocated across the multiple paths to achieve the target success probability at the destination. We determine the minimum amount of redundancy necessary for this probability of success to be achievable.

II. Problem Setup and Background Information

We have a source node that wishes to transmit a message to a specified destination node, within a certain probability of success. Between these two nodes there are N paths that have a security level associated with them. In other words, path i has a value p_i assigned to it which represents the probability of successfully receiving packets sent across this path. Without loss of generality we assume that $p_1 \geq p_2 \geq \ldots \geq p_N$. We want to disperse the message among these paths so that successful reception has at least target success probability p^*. In the case that one or more paths has a vicious node, the destination node cannot recover the original message. We suggest the use of maximum-distance separable codes to mitigate this problem.

A. Maximum-Distance Separable Codes (MDS)

Before going into the details of MDS codes, we will begin with some basic theorems and definitions [1].

Definition 1: **Hamming Distance**
The *Hamming Distance* between two codewords $\mathbf{u}, \mathbf{v}$ of length n is the number of positions in which they differ or,

$$d_{\text{Hamming}}(\mathbf{u}, \mathbf{v}) = d(\mathbf{u}, \mathbf{v}) = |\{i | u_i \neq v_i,\ i = 0, 1, \ldots, n-1\}|$$

Definition 2: **Minimum Distance of a Code**
The *Minimum Distance* , $d_{\min}$ of a code is the minimum Hamming distance between all distinct codewords in the codebook.
An (n, k) code is one which starts with a message of length k and encodes it to a codeword of length n, or adds $n-k$ redundant symbols. We call the ratio $\frac{n}{k} = \gamma$.

Theorem 1: **Singleton Bound**
The minimum distance $d_{\min}$ for an (n, k) code is bounded by

$$d_{\min} \leq n - k + 1$$

Maximum-distance separable codes are linear block codes that satisfy the Singleton Bound with equality; i.e., for an (n, k) code, $d_{\min} = n - k + 1$. Consider a k-symbol message which is encoded into an n symbol MDS codeword. It is known (see, for example, [1]) that any combination of k out of n symbols in an MDS code can be used to obtain the original message. With this useful property, as long as $\leq n-k$ symbols are lost, we can perfectly reconstruct the primary message (assuming that there are no errors in the received symbols).

B. System Model

The N paths between the source and destination node are independent from one another, and are each assumed to be a block erasure channel [8]. By block erasure channel we mean that either the symbols in a given transmission are received perfectly or they are not received at all. Let $\mathbf{f}$ denote the symbol allocation vector, where f_i represents the number of symbols allocated to path i. We can compose a vector $\mathbf{s}$ of length N, composed of 0's and 1's, where a 0 in spot i represents failure of path i to deliver its symbols, and a 1 in spot i represents a successful transmission at path i. The message is extended from k symbols to n symbols, and the scenario that k or more symbols are received means that the message can be decoded without error. Let S be a matrix of all possible vectors s. Then, the probability of successful reception of the message at the destination can be written as,

$$P_{success}(\mathbf{f}) = \sum_{\mathbf{s} \in S} \prod_{i=1}^{N} p_i^{s_i} (1 - p_i)^{1 - s_i} u(\mathbf{s} \cdot \mathbf{f} - k) \qquad (1)$$

where $u(\cdot)$ is the unit step function.
It can be seen that S is composed of 2^N vectors, and hence, calculating the probability of success for a given vector $\mathbf{f}$ takes exponential time. For each path, the probability that the symbols transmitted across the path are received follows a Bernoulli distribution. Thus, in [4] they find an approximation for the probability of success based on the fact that a Bernoulli distribution can be approximated by a Gaussian distribution. Since the paths are independent, if we model them all using a Gaussian distribution, the joint distribution of all the paths is also a Gaussian distribution, $\sim \mathcal{N}\left(\sum_{i=1}^{N} f_i p_i, \sum_{i=1}^{N} f_i^2 p_i (1 - p_i)\right)$. The approximation of the probability of success is,

$$P_{success}(\mathbf{f}) \approx \frac{1}{2} + \frac{1}{2} erf\left(\frac{\sum_{i=1}^{N} f_i p_i - k + \frac{1}{2}}{\sqrt{2 \sum_{i=1}^{N} f_i^2 p_i (1 - p_i)}}\right) \qquad (2)$$

Given a target success probability p^*, the question that needs to be answered is: what is the minimum redundancy, and how do we optimally distribute symbols, to attain success probability of at least p^*? Naturally, the source node will use knowledge of the security of each path to its advantage, most likely sending more information down the most trustworthy path. This implies that $f_1 \geq f_2 \geq \ldots \geq f_N$. Another initial observation that we make is that sending more than k symbols down any path is useless, since we only need k symbols to decode the original message. Hence, for N paths, the highest redundancy one can have is $\gamma = N$. Also, if $p^* \leq p_1$ then the optimal approach is to have $\gamma = 1$ and $f_1 = k, f_2 = 0, \ldots, f_N = 0$. For simplicity, we begin with the case where $N = 3$.

III. Optimal Symbol Allocation and Minimum Redundancy for $N = 3$

Initially we assume that there exist 3 active paths that are ranked based on their "trustworthiness" level. The first step in distributing symbols to the paths is to determine the ratio $\gamma = \frac{n}{k}$. For example, if we have 3 active paths, and $\gamma = 3$, it is clear that the optimal approach is to send $f_1, f_2, f_3 = k$ symbols along paths 1,2, and 3. Based on p^* and the path success probabilities, using brute force, we determine the minimum redundancy and optimal symbol allocation as follows:

- If $p_1 + p_2 - p_1 p_2 < p^* \leq p_1 + p_2 + p_3 - p_1 p_2 - p_2 p_3 - p_1 p_3 + p_1 p_2 p_3$
  $\Rightarrow \gamma_{\min} = 3$ and $f_1, f_2, f_3 = k$
- If $\max\{p_1 p_2 + p_2 p_3 + p_1 p_3 - 2 p_1 p_2 p_3, p_1\} < p^* \leq p_1 + p_2 - p_1 p_2$
  $\Rightarrow \gamma_{\min} = 2$ and $f_1, f_2 = k, \quad f_3 = 0$

- If $p_1 < p_1p_2 + p_2p_3 + p_1p_3 - 2p_1p_2p_3$ and $p_1 < p^* \leq p_1p_2 + p_2p_3 + p_1p_3 - 2p_1p_2p_3$
  $\Rightarrow \gamma_{\min} = \frac{3}{2}$ and $f_1, f_2, f_3 = \frac{k}{2}$
- If $0 < p^* \leq p_1$
  $\Rightarrow \gamma_{\min} = 1$ and $f_1 = k, f_2, f_3 = 0$

Below in Fig. 1 we plot the minimum redundancy versus the target success probability for the case where $p_1 \geq p_1p_2 + p_2p_3 + p_1p_3 - 2p_1p_2p_3$.

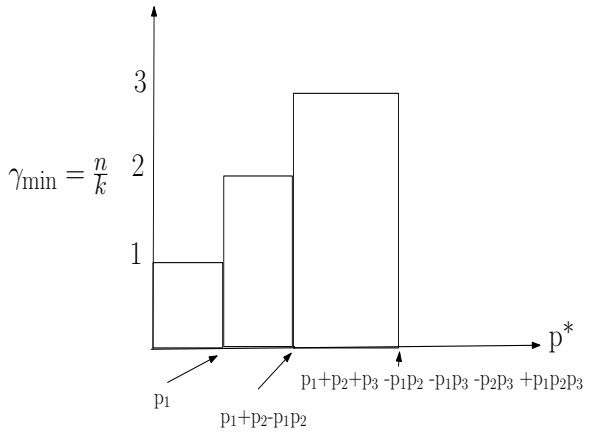

Fig. 1. Minimum Redundancy for $N = 3$ when $p_1 \geq p_1p_2 + p_2p_3 + p_1p_3 - 2p_1p_2p_3$

If $p_1 < p_1p_2 + p_2p_3 + p_1p_3 - 2p_1p_2p_3$ then we see that there is an additional possibility for the redundancy, $\gamma = \frac{3}{2}$. We show this in Fig 2.

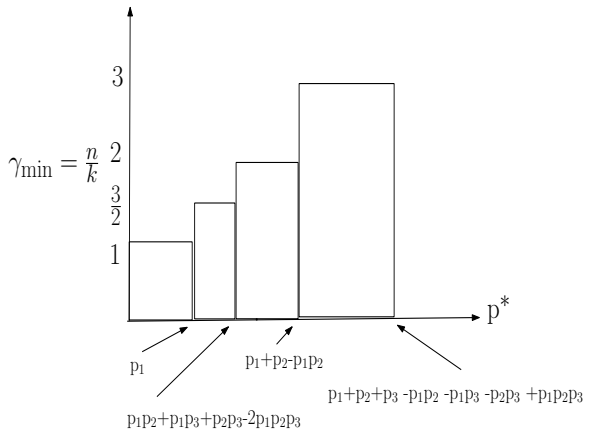

Fig. 2. Minimum Redundancy for $N = 3$ when $p_1 < p_1p_2 + p_2p_3 + p_1p_3 - 2p_1p_2p_3$

This is obtained by considering all possible logical symbol distributions among the paths. We know that $\sum_{i=1}^{3} f_i \geq k$, and the probability of success is dependent on how many of the paths have symbol allocations that sum to k. Since we are assuming that as the path number increases the probability of success decreases, this greatly diminishes the number of possible symbol allocations. So for example, for $\gamma_{\min} = 1$ we have three choices for the allocation which are: $f_1 = f_2 = f_3 = \frac{k}{3}$, $f_1 = f_2 = \frac{k}{2}, f_3 = 0$ or $f_1 = k, f_2 = f_3 = 0$. These choices are associated with the success probabilities $p_1p_2p_3$, p_1p_2, p_1 respectively, and clearly p_1 is the greatest. We proceed like this for all scenarios, and obtain the results.

IV. Algorithm for Arbitrary Number of Paths

Generally, the size of the active path set is known prior to transmission, though it is dynamic in the sense that the cardinality of this set changes every time the security of each path is assessed. Due to the structure of the success probability function, it is difficult to obtain a closed form expression for the redundancy and symbol allocation. We take another approach, and develop heuristic algorithms Minimum Redundancy Algorithm Exponential Time (MRAET) and Polynomial Time (MRAPT) which determine these mentioned parameters to achieve the desired probability of success. The algorithms are composed of two parts, where the first part is primarily to reduce the dimensionality of the search space and to give us a starting point. We assume that the table of the set S is computed offline, and is not considered in the running time analysis.

The first part of the algorithm considers the simple case where either we assign 0 or k symbols to path i, $i \in \{1, \ldots, N\}$. It finds the first path j such that if all the paths $i \in \{1, \ldots, j\}$ are assigned k symbols, and the rest of the paths have 0 symbols, then the probability of success is at least p^*. It is clear that if only the first $j-1$ paths are assigned k symbols, and the rest 0, then the success probability is strictly less than the target probability of success.

In the simple case of three paths it can be seen that the only way we can diminish the redundancy of the code is by starting at the point where the first $j-2$ paths have k symbols and the rest having 0 symbols. We will first define some notation in the algorithm for simplicity:

$$P^A_{success} = \sum_{\mathbf{s} \in A} \prod_{i=1}^{N} p_i^{s_i} (1 - p_i)^{1 - s_i}$$

Where A is some subset of S. Let,

$$Z_{(i,s)} = \{z \in \{1, \ldots, 2^{N-(j-2)}\} \mid \sum_{l=j-1}^{j-2+i} S_{z,l} \geq s\}$$

for some integers s, i. Where $S_{z,l}$ represents the element of S in the z^{th} row and l^{th} column.

Minimum Redundancy Algorithm in Exponential Time Part 1:

Step 1: Assign $j = 1$ and go to step 2.
Step 2: Let $A = S_{\left((2^j+1,\ldots,2^N),(1,\ldots,N)\right)}$ and go to step 3.
Step 3: Calculate $P^A_{success}$.
If $P^A_{success}(\mathbf{f}) \geq p^*$ then save j, let
$f_1, \ldots, f_j = k, f_{j+1}, \ldots, f_N = 0$
and $\gamma_{\min} = j$, $P_{temp} = P^A_{success}$.
Then go to part 2 of the algorithm.

Otherwise let $j = j + 1$ and if $j > N$ move on to Part 2, else if $j \leq N$ return to Step 2.

If $j < 2$ then we have an optimal allocation and we are done. Otherwise:

Part 2:
Let $i = 2$

Step 1: Let $i = i+1$ and if either $i > N$ or $j-2+i > N$ then terminate Part 2, otherwise go to step 2.

Step 2: Let $s = 2$

Step 3: If $j - 2 + \frac{i}{s} \leq \gamma_{\min}$ then
let A denote the subset of matrix S composed of rows whose indices are in $Z_{(i,s)}$ and go to step 4. Otherwise, go to step 6.

Step 4: Calculate $P^A_{success}$
If $P^A_{success} \geq p^*$ with $j - 2 + \frac{i}{s} < \gamma_{\min}$
or
$j - 2 + \frac{i}{s} = \gamma_{\min}$ and $P_{temp} < P^A_{success}$
then go to step 5, otherwise go to step 6.

Step 5: Let $P_{temp} = P^A_{success}$, $\gamma_{\min} = j - 2 + \frac{i}{s}$, and
$f_1, \ldots, f_{j-2} = k$,
$f_{j-1}, \ldots f_{j-2+i} = \frac{k}{s}$,
$f_{j-1+i}, \ldots, f_N = 0$.
Go to step 6.

Step 6: Let $s = s + 1$. If $s > i$ then go to step 1, else go to step 3.

This algorithm is optimal for several of cases. One case is when we have $N = 3$ paths.

Theorem 2: **MRAET is optimal when** $N = 3$
Proof: After Part 1 of the algorithm, we have 3 options for j, $j = 1, 2, 3$.
If $j = 1$, then the algorithm terminates after part 1 since $j < 2$. We are left with $\gamma_{min} = 1 \Rightarrow n = k$

Thus $f_1 = k, f_2 = f_3 = 0$ which is optimal.
If $j = 2$, then

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

and $P_{temp} = p_1 + p_2 - p_1 p_2$
The algorithm then steps into part 2. It starts and ends with the scenario $i = 3, s = 2$ since $N = 3$. It first checks is $j - 2 + \frac{i}{s} \leq \gamma_{\min} = j$. If so, then it searches through

$$S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

rows $(1, \ldots, 2^{N-(j-2)}) = (1, \ldots, 8)$ such that the columns $(j-1, \ldots, j-2+i) = (1, \ldots, 3)$ sum to greater than or equal to $s = 2$. Then

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Then since $j - 2 + \frac{i}{s} = \frac{3}{2} < j = 2$, MRAET checks if $P_{success} = \sum_{\mathbf{s} \in A} \prod_{l=1}^{3} p_i^{s_i} (1-p_i)^{1-s_i} = p_1 p_2 + p_2 p_1 + p_1 p_3 - 2 p_1 p_2 p_3 \geq p^*$. If so, then $\gamma_{\min} = \frac{3}{2}$ and $f_1, f_2, f_3 = \frac{k}{s} = \frac{k}{2}$, otherwise $\gamma_{\min} = 2$ and $f_1 = f_2 = k, f_3 = 0$.

Lastly, if $j = 3$ the algorithm stops before part 2 because $j - 2 + 3 > 3$. Thus, $f_1 = f_2 = f_3 = k$, which is the same allocation as we had above. □

Next, we prove a theorem to help us show another optimal case.

Theorem 3: **Suppose we have k symbols allocated to paths $1, \ldots, i-1$ and 0 symbols allocated to the remaining of the N paths, resulting with probability of success $\hat{p}_{i-1}$. Then, if we let path i have k symbols, the probability of success is**

$$\hat{p}_i = \hat{p}_{i-1} + p_i - \hat{p}_{i-1} p_i \tag{3}$$

Proof: By induction on the integer i.

Base Case: $i = 2$
We have $\hat{p}_{i-1} = \hat{p}_1 = p_1$, since we only have success if path 1 succeeds. If we let path 2 have k symbols, then we have a success solely if path 1 succeeds, if path 2 is the only successful one, or if they both succeed. This is equivalent to:

$$\hat{p}_2 = \hat{p}_1 p_2 + \hat{p}_1 (1 - p_2) + (1 - \hat{p}_1) p_2 = \hat{p}_1 + p_2 - \hat{p}_1 p_2$$

Inductive Hypothesis: Suppose Eq. 3 holds $\forall i \leq m - 1$

Inductive Step: Let $i = m$. Then by inductive hypothesis we know that $\hat{p}_{m-1}$ is the probability of success for the first $m-1$ paths having k symbols and the rest having 0. We can think of $\hat{p}_{m-1}$ as being the probability of success for one super path. Thus, if we let the m^{th} path have k symbols, then we have success if only the super path is successful, the m^{th} path is the only successful one, or if they are both successful. That is:

$$\begin{aligned} \hat{p}_m &= \hat{p}_{m-1} p_m + \hat{p}_{m-1}(1 - p_m) + (1 - \hat{p}_{m-1}) p_m \\ &= \hat{p}_{m-1} + p_m - \hat{p}_{m-1} p_m \end{aligned}$$

Hence the result holds $\forall i \in 2, \ldots N$. □

Theorem 4: **MRAET is optimal when** $j = N$
Proof: If j=N, then we know that A is equal to S, excluding the all zero first row, $P^A_{success} \geq p^*$.
By Thm. 3 we know that the probability of success for the first $N - 2$ paths having k symbols and the rest having 0 is:

$$\hat{p}_{N-2} = \hat{p}_{N-3} + p_{N-2} - \hat{p}_{N-3} p_{N-2} < p^*$$

Thus, if we treat the first $N - 2$ paths as one super path with probability $p_{super} = \hat{p}_{N-2}$, then the current problem can

be mapped to the case where $N = j = 3$ since super path is path 1, $N-1$ is path 2, and N is our third path. □

Corollary 1: **MRAET is optimal when** $j = N-1$
This results follows from the theorem above, since $j = N-1$ can be mapped to the case where $N = 3$ and $j = 2$.

Minimum Redundancy Algorithm in Polynomial Time (MRAPT)

Proceed with Part 1 as in MRAET, though using the approximation for probability of success found in Eq. 2. Then proceed similarly as in Part 2 though we do not have to search through the matrix S, and do not need to go through the exponential time calculation of the probability of success. After part 2, we are done since we already have the symbol allocation vector $\mathbf{f}$.

A. Running Time Analysis

We use a logarithmic time search algorithm such as binary search, [7], for the search through S , so the search takes $O(log_2(2^N)) = O(N)$ time (since we search at most all rows of S). The calculation of the probability of success requires $O(2^N)$ steps worse case, if $A = S$. So the inner portion of part 1 for MRAET takes $O(N + 2^N)$ time and the inner time is performed at most N times, thus part 1 has worst time complexity $O(N(N + 2^N))$. Part 2 is similar though there are two loops, and hence the worst case running time is $O(N^2(N + 2^N))$. Hence it is clear that MRAET has exponential running time. It can be seen that if the search inside the S matrix is not included, as well as calculating the true probability of success, the running time is reduced drastically.

In MRAPT, since we are using approximation Eq.2, we first calculate the mean and variance of the Gaussian distribution which takes $O(2N), O(4N)$ respectively. If we save $p_i(1-p_i)\ \forall i$ prior to proceeding with MRAPT then the variance will only take $O(2N)$ time. Thus, part 1 has worst case running time of $O(N(4N)) = O(N^2)$. Similarly, part 2 has a worst case running time $O(N^3)$. It can be seen that this is polynomial in N, the number of paths in the APS.

V. Simulations

We run both the algorithms over numerous Monte Carlo runs in MATLAB, and compare their true probability of success versus the approximated probability of success and the target success probability in Fig.3. We let $N = 7$, $k = 4$, and $\mathbf{p} = [0.8000, 0.5901, 0.5338, 0.5261, 0.5203, 0.5107, 0.5000]^T$.
It can be seen that in Fig. 3 that the probability of success for MRAPT is very close to the desired success probability, and is below it for a very short time. The exponential time algorithm, MRAET, always achieves greater than or equal to the target probability. Also, the approximation for the probability of success, 2 seems to be an overestimate for the simulated success probability for smaller desired success probabilities, but as they get larger, the approximation of success seems to be a better estimate.

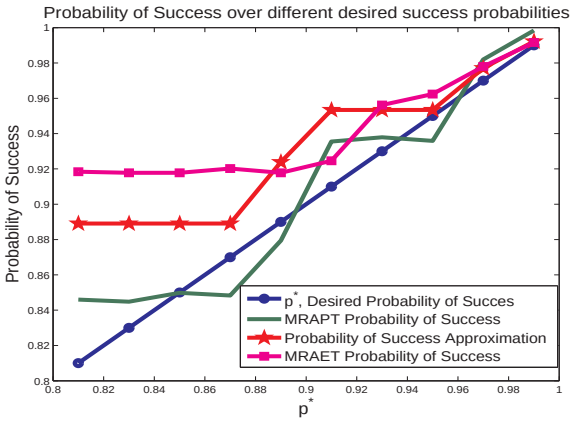

Fig. 3. Probability of Success of MRAPT and MRAET

Next in Fig. 4, we compare the minimum redundancy ratio picked by the two algorithms versus the target success probability. MRAET has a lower redundancy for all the success probabilities, though the polynomial algorithm is fairly close for lower success probabilities. When the success probability becomes large, the polynomial algorithms' redundancy gets much larger than that of the exponential time algorithm.

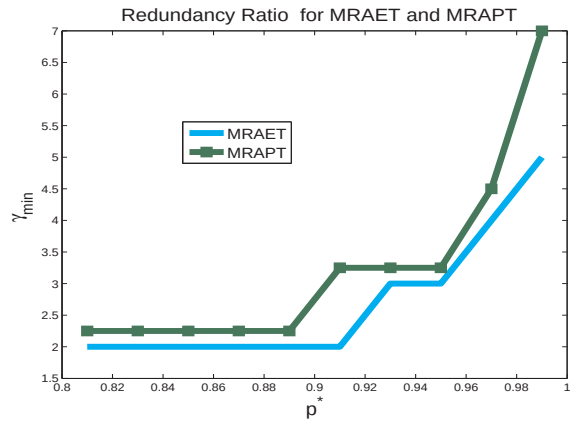

Fig. 4. Redundancy Ratio for MRAPT and MRAET

Lastly in Figures 5 and 6, we compare the actual symbol allocation in the two algorithms versus the desired success probability. MRAET gives more packets to the lower indexed paths, while MRAPT seems to allocate them more evenly. It seems that for the high probability of success, the MRAPT overshoots and gives the paths many more symbols then necessary.

VI. Conclusion

We considered the problem of information dispersal in a multipath mobile ad hoc network. There are a set of paths between the source and destination nodes, with each path assigned a certain trustworthiness level. We use maximum-distance separable error-correction codes for channel coding to

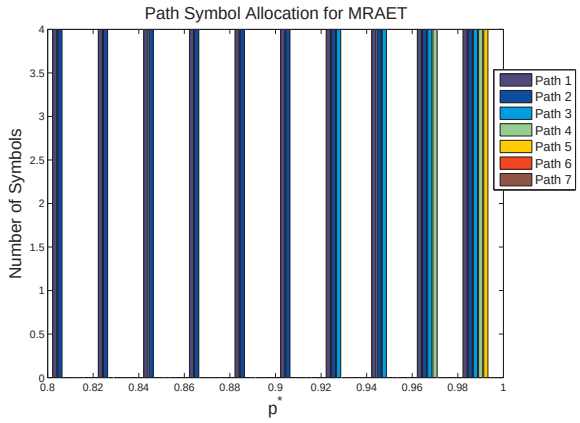

Fig. 5. Symbol Allocation for MRAET

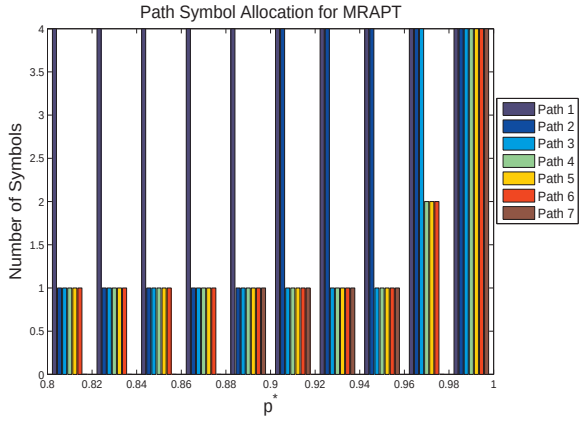

Fig. 6. Symbol Allocation for MRAPT

correct for possible link failure. Based on the measure of trustworthiness and the desired minimum success probability, we have determined the minimum redundancy and optimal path symbol allocation for the 3-path case. Due to the complexity of the probability of success function, we developed two algorithms (MRAET, MRAPT) for finding these parameters in the case of arbitrary number of paths. MRAET is an exponential time algorithm and it has been shown to be optimal for several cases. Using an approximation for the probability of success, we have also developed a polynomial time algorithm, MRAPT, which performs closely to the target success probability .

REFERENCES

[1] Stephen B. Wicker. *Error Control Systems for Digital Communication and Storage*. Upper Saddle River,NJ. Prentice Hall, 1995.

[2] Papadimitratos, P., and Haas, Z.J., "'Secure Message Transmission in Mobile Ad Hoc Networks,"'Ad Hoc Networks,pp.193-209,2003.

[3] Tsirigos, A., and Haas, Z.J., "'Analysis of Multipath Routing, Part 1: The Effect on the Packet Delivery Ratio,"'IEEE Trans. on Wireless Communication ,Vol. 3, No. 1, January 2004.

[4] Tsirigos, A., and Haas, Z.J., "'Analysis of Multipath Routing, Part 2: Mitigation of the Effects of Frequently Changing Network Topologies,"'IEEE Trans. on Wireless Communication ,Vol. 3, No. 2, March 2004.

[5] M.O. Rabin, Efficient dispersal of information for security, load balancing, and fault tolerance, J. ACM 36 (2) (1989) 335348.

[6] S. J. Lee and M. Gerla, Split multipath routing with maximally disjoint paths in ad hoc networks, in Proc. Int. Conf. Communications (ICC 2001), 2001, pp. 32013205.

[7] Cormen, T.H, Leiserson, C.E, Rivest, R.L, Stein, L. *Introduction to Algorithms*. MIT Press; second edition , 2001.

[8] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*.John-Wiley and Sons, Inc. 2nd edition, 2006.

[9] Wicker, S. B. and Bhargava, V. K. (editors), *ReedSolomon Codes and Their Applications*, Piscataway: IEEE Press, 1994.