Multipath routing in mobile ad hoc networks or how to route in the presence of frequent topology changes

Aristotelis Tsirigos
School of Electrical Engineering
Cornell University
Ithaca, NY 14853
E-mail: artsi@ece.cornell.edu

Zygmunt J. Haas
School of Electrical Engineering
Cornell University
Ithaca, NY 14853
E-mail: haas@ece.cornell.edu
http://www.ee.cornell.edu/~haas/wnl.html

Siamak S. Tabrizi
Air Force Research
IFGC
Rome, NY 13441
E-mail: Siamak.Tabrizi@rl.af.mil

Abstract—

In this paper, we propose a framework for multipath routing in mobile ad hoc networks (MANETs) and provide its analytical evaluation. The instability of the topology (e.g., failure of links) in this type of networks, due to nodal mobility and changes in wireless propagation conditions, makes transmission of time-sensitive information a challenging problem. To combat this inherent unreliability of these networks, we propose a routing scheme that uses multiple paths simultaneously by splitting the information among the multitude of paths, so as to increase the probability that the essential portion of the information is received at the destination without incurring excessive delay.

Our scheme works by adding an overhead to each packet, which is calculated as a linear function of the original packet bits. The resulting packet (information and overhead) is fragmented into smaller blocks and distributed over the available paths. Our goal is, given the failure probabilities of the paths, to find the optimal way to fragment and then distribute the blocks to the paths, so that the probability of reconstructing the original information at the destination is maximized. Our algorithm has low time-complexity, which is crucial since the path failure characteristics vary with time and the optimal block distribution has to be recalculated in real-time.

I. Introduction

In this paper we consider the problem of routing data over multiple disjoint paths in an ad hoc network. A lot of research has been done in the area of multipath routing in wired networks. One of the initial approaches to this problem was the *Dispersity Routing* [1]. In [2], another multipath scheme is proposed, *Diversity Coding*, in order to achieve self-healing and fault tolerance in digital communication networks. In [3], a per-packet allocation granularity for multipath source routing schemes was shown to perform better than a per-connection allocation. An exhaustive simulation of the various tradeoffs associated with dispersity routing was presented in [4].

The application of multipath techniques in mobile ad hoc networks seems natural, as multipath routing allows to diminish the effect of unreliable wireless links and the constantly changing topology. The *On-Demand Multipath Routing* scheme is presented in [5] as a multipath extension of *Dynamic Source Routing* (DSR) [6], in which alternate routes are maintained, so that they can be utilized when the primary one fails. Another extension of DSR, *Multiple Source Routing (MSR)* [7], proposes a weighted-round-robin heuristic-based scheduling strategy among multiple paths in order to distribute load, but provides no analytical modeling of its performance. The *Split Multipath Routing* (SMR), proposed in [8], focuses on building and maintaining maximally disjoint paths, however, the load is distributed only in two roots per session. In [9], the positive effect of *Alternate Path Routing* (APR) on load balancing and end-to-end delay in mobile ad hoc networks has been explored. It was argued however, that the network topology and channel characteristics (e.g., route coupling) can severely limit the gain offered by APR strategies. In an interesting application [10], *Multipath Path Transport* (MPT) is combined with *Multiple Description Coding* (MDC) in order to send video and image information in a multihop mobile radio network.

In our paper, we propose a multipath scheme for mobile ad hoc networks based on *Diversity Coding* [2]. Data load is distributed over multiple paths in order to minimize the packet drop rate, achieve load balancing, and improve end-to-end-delay. The routing paradigm is depicted in figure 1, where three different paths are utilized at the same time in order to send data from a source to a destination node. As we explain in the next section, each data packet is split into multiple pieces, which are distributed among the available paths. We evaluate our scheme by calculating the probability that a transmission from the source results in successful packet reception at the destination. The probability function of successful reception is analytically derived and data is split over multiple paths in such a way, that the function is maximized.

The model we are using in order to evaluate our scheme is developed under the assumption that the mean time of packet transmission is much smaller than the mean time between variations in network topology. If this assumption holds, then we can assume that the probability that one or more path links fail is constant during the transmission of a packet. In other words, one can assume that the topology of the network

This work has been sponsored in part by ONR contract no. N00014-00-1-0564, AFRL contract no. F360602-97-C-0133, and NSF grant no. ANI-9980521.

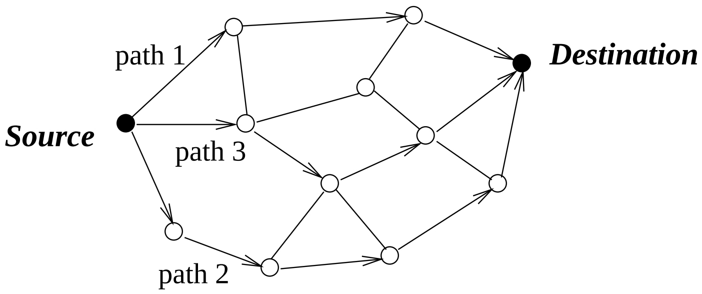

Fig. 1. Using multiple paths in an ad hoc network.

will not change significantly while a packet is being transmitted.

Our paper is organized as follows: Section II provides a description of the proposed scheme and the definition of the successful transmission probability function P_{succ}, the function used for the evaluation of the scheme. In section III we derive an analytical formula for P_{succ} and its approximation. Finally, in section IV we conclude the work and set some goals for our research in the future.

II. Description of our scheme

In this section, we describe how our scheme exploits the multitude of paths, in order to offer increased protection against path failures. Data packets are sent from source to destination over these paths, making use of *Diversity Coding* [2], which we explain later in this section.

In our network model, we assume that n_{max} paths are available for the transmission of data packets from a source to a destination node. All paths are mutually disjoint, i.e., they have no nodes in common. Each path, indexed as i, $i = 1..n_{max}$, is assigned a probability of failure p_i, which is the probability that path i is down at the time that the source attempts to transmit. In addition, each path is treated as a pure erasure channel: either no information reaches the destination through path i (with probability p_i), or all the information is received correctly (with probability $1 - p_i$). Since there are no common nodes among the paths, they are considered independent in the sense that success or failure of one path cannot imply success or failure of another.

Without loss of generality, the failure probabilities of the available paths are organized in the probability vector $\underline{p} = [p_i], i = 1..n_{max}$, in such a way that $p_i \leq p_{i+1}$, i.e., the paths are ordered from the "best" one to the "worst" one. Given $\underline{p}$ we also define $\underline{q} = [q_i], q_i = 1 - p_i, i = 1..n_{max}$, which is the vector of the success probabilities. Throughout the paper we use $\underline{p}$ and $\underline{q}$ interchangeably.

The failure probability vector $\underline{p}$ reflects the network topology and the quality of the available routes, no matter what the node mobility pattern is. Our goal is to develop a method for the fast calculation of the optimal solution defined later in this section, so that our scheme can respond (i.e., recalculate the optimal solution) to rapid changes in $\underline{p}$, i.e. changes in the network topology.

Let's suppose that the proposed scheme has to send a packet of X information bits utilizing the set of available independent paths in such a way as to maximize the probability that these bits are successfully communicated to the destination. This probability is denoted as P_{succ}. In order to achieve this goal, we employ a coding scheme in which Y extra bits are added as overhead. The resulting B bits ($B = X + Y$) are treated as one network-layer packet. The extra bits are calculated as a function of the information bits in such a way that, when splitting the B-bit packet into multiple equal-size non-overlapping blocks, the initial X-bit packet can be reconstructed given any subset of these blocks with a total size of X or more bits. First, we define the overhead factor r:

$$r = \frac{B}{X} = \frac{b}{x}, \tag{1}$$

where b and x take integer values and the fraction b/x cannot be further simplified, i.e., the greatest common divisor of b and x is 1.

The key decision that we have to make is how the B bits will be distributed over the available paths. For this reason, we define the vector $\underline{v} = [v_i]$, where v_i is the number of equal-size blocks that is allocated to path i. Clearly, some of the paths may demonstrate such a poor performance that there is no point in using them at all. This means that we might require to use only some of the available paths. If n is the number of the paths we have to use in order to maximize P_{succ}, it would be preferable to define the block allocation vector $\underline{v}$, as a vector with a variable size n, instead of fixing its size to the number of available paths (i.e., n_{max}). Given the fact that the probability vector is ordered from the best path to the worst one, a decision to use n paths implies that these paths will be the first n ones. Based on these observations, the allocation vector $\underline{v}$ has the following form:

$$\underline{v} = (v_1, v_2, ..., v_n), n \leq n_{max}. \tag{2}$$

If the block size is w then:

$$w \cdot \sum_{i=1}^{n} v_i = B = rX. \tag{3}$$

Therefore the total number of blocks that the B-bit packet is fragmented to is:

$$a = \sum_{i=1}^{n} v_i = \frac{rX}{w}. \tag{4}$$

From $p_i \leq p_{i+1}$ follows that $v_i \geq v_{i+1}$, because a path with higher failure probability cannot be assigned less blocks than a path with a lower failure probability. As a convention, we also set v_n to 1, i.e., the last path we use receives 1 block.

In figure 2 we can see the B-bit packet and its relation to the original X-bit packet (gray area). We also show how the B-bit packet is fragmented into equal-size non-overlapping blocks of size w. The original X-bit packet is fragmented into N w-size blocks, $d_1, ..., d_N$, and the Y-bit overhead packet into M w-size blocks, $c_1, ..., c_M$. Path 1 will be assigned the first v_1 blocks of the B-bit sequence, path two will receive the next v_2 blocks and so on. Thus path i will be assigned v_i blocks, each block of size w.

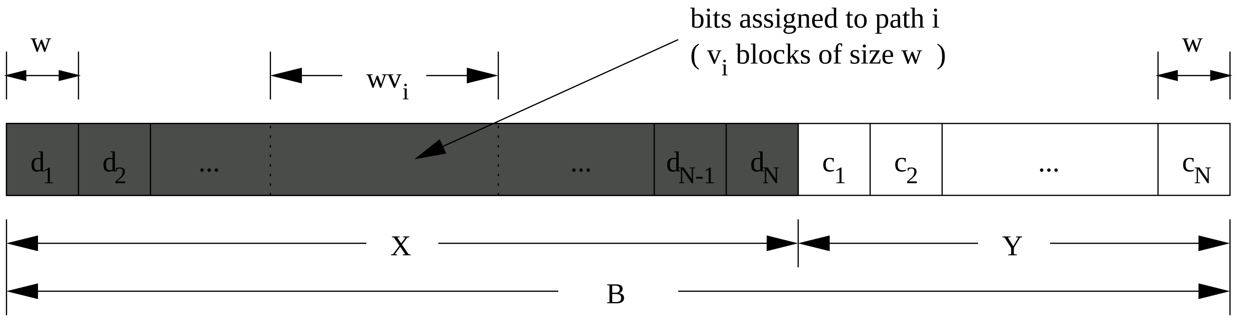

Fig. 2. Information and overhead packet fragmentation.

We can derive the expressions for N and M from figure 2:

$$N = \frac{X}{w} = \frac{a}{r}, \tag{5}$$

$$M = \frac{Y}{w} = (r-1)N = \frac{r-1}{r}a. \tag{6}$$

This is a typical case where *M-for-N Diversity Coding* can be applied. In [2], Ayanoglu et al. have proved that if M or less blocks are lost, out of the $N+M$ total data and overhead blocks, the original N information blocks can be recovered using appropriate linear transformations. The overhead blocks $c_i, i = 1..M$, are also calculated as a linear transformation of the information blocks $d_i, i = 1..N$. The block size has to satisfy the following inequality, so that the original information can be recovered:

$$w \geq \lceil \log_2(N+M+1) \rceil \geq \log_2(a+1), \tag{7}$$

where we used equations (5) and (6). By incorporating (4) into (7), we give an inequality for the number of blocks a, into which we can split the B-bit packet:

$$B \geq a \log_2(a+1). \tag{8}$$

The optimization algorithm we developed is used to determine the optimal number of paths and the optimal allocation vector, given the path probability vector $\underline{p}$ and the overhead factor r. The details of this algorithm are explained in our paper [11] and will be omitted here because of space limitations. The optimization process involves the maximization of P_{succ}, the definition of which we give shortly.

If v_i is the number of blocks we send over path i, and z_i the number of blocks that actually reach the destination through path i, then:

- $Pr\{z_i = v_i\} = q_i$
- $Pr\{z_i = 0\} = p_i$

because we assume that if a path fails, then all the blocks sent over the path are lost (recall the pure erasure channel assumption). *M-for-N Diversity Coding* can reconstruct the original X-bit information packet, provided that at least N blocks reach the destination. Therefore, we can define P_{succ} in terms of the number of paths that are actually used and the corresponding allocation vector:

$$P_{succ}(n, \underline{v}) = Pr\left\{\sum_{i=1}^{n} z_i \geq \frac{a}{r}\right\}, \tag{9}$$

where we expressed N as a function of a and r, using equation (5).

III. Evaluation of the function P_{succ}

In section III-A below, we use the definition in equation (9) in order to provide an analytical formula for P_{succ} and to estimate its complexity. In section III-B, we present a formula that approximates P_{succ}, and based on that formula, we explain how an optimal allocation of blocks to the paths can be obtained. In section III-C, we present the evaluation results.

A. Formula and complexity of P_{succ}

In this section we present a formula for the calculation of the probability of success P_{succ}, given the probability vector $\underline{p}$, the overhead factor r, and the allocation vector $\underline{v}$. We also give an estimation of the complexity of this function in terms of the number of the multiplications involved in its calculations.

According to our network model, each one of the n paths used by our scheme is subject to two distinct events:

- the event of failure to transmit the assigned packets (probability p_i)
- the event of successful attempt to transmit the packets (probability q_i)

We define an n-dimensional vector $\underline{s}$, which reflects the state of the n paths:

- $s_i = 0$, if path i failed
- $s_i = 1$, if path i succeeded

The associated probabilities are:

- $Pr\{s_i = 0\} = p_i$
- $Pr\{s_i = 1\} = q_i = 1 - p_i$

The probability $t(\underline{s})$ of the n paths being in state $\underline{s}$ is easily calculated as:

$$t(\underline{s}) = \prod_{i=1}^{n} {p_i}^{1-s_i} {q_i}^{s_i} \tag{10}$$

The cost of calculating $t(\underline{s})$, defined as the number of multiplication operations, is:

$$C\{t(\underline{s})\} = n \tag{11}$$

Each different state corresponds to a different set of paths succeeding in transmitting packets. All possible states describe all combinations of such sets, thus covering the whole space of events (2^n events in total). Since a transmission is successful, when at least N blocks arrive at the destination, each term $t(\underline{s})$, defined in equation (10), can contribute to P_{succ} only if the number of blocks sent over the set of paths described by $\underline{s}$ is more than or equal to N. By making this observation and by replacing N using equation (5), we can write P_{succ} as [1]:

$$P_{succ}(n, \underline{v}) = \sum_{\underline{s}} t(\underline{s}) \cdot u(\underline{s} \cdot \underline{v} - \frac{a}{r}), \quad (12)$$

where $\underline{s} \cdot \underline{v}$ is the inner product of vectors $\underline{s}$ and $\underline{v}$, and equals the total number of successfully received blocks allocated to the subset of paths described by the state vector $\underline{s}$. The function $u(\cdot)$ in equation (12) is the unit step function defined as:

$$u(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}$$

The cost in multiplication operations of calculating P_{succ} using equation (12) is:

$$C\{P_{succ}(n, \underline{v})\} = n \cdot 2^n, \quad (13)$$

because the total number of states is 2^n and each term has a cost of n (see equation (11)). Given the probability vector $\underline{p}$ and the overhead factor r as parameters, we are looking for the optimal number n^* of paths to use (out of the n_{max} available ones) as well as the optimal allocation vector b^* over the n^* paths, so that P_{succ} is maximized.

Equation (12) is a complex formula, which makes it impossible to apply analytical maximizing techniques such as Lagrange multipliers, primarily because of the presence of the unit step function. Let us estimate the cost of exhaustively testing all combinations of the allocation vector $\underline{v}$, in order to find the maximum of P_{succ}. The number of possible allocations of B bits to n paths is $\binom{B+n-1}{n-1}$. As we have explained before, for a given n and $\underline{v}$, computing P_{succ} involves $n \cdot 2^n$ multiplication operations. Therefore the total cost is:

$$C = \sum_{n=1}^{n_{max}} n 2^n \binom{B+n-1}{n-1}. \quad (14)$$

It is clear that the cost of brute force optimization of P_{succ} is exponential. Even for a small number of paths (e.g., 10 paths), testing all the valid combinations will take unacceptably long time, making the optimization of P_{succ} impossible to implement in real-time. Even if programming techniques such as dynamic programming are employed, the size of the search space remains exponential with respect to n. Moreover, the computation of P_{succ} itself requires exponential time, as shown in equation (13), so even if we knew the optimal solution, it would take a long time to calculate the value of P_{succ} for that solution. The reader is reminded that in a mobile ad hoc network the probability vector $\underline{p}$ will not be constant with time and therefore the optimization process of P_{succ} must be repeated when the network topology changes.

[1]The reader is reminded that $a = \sum_{i=1}^{n} v_i$

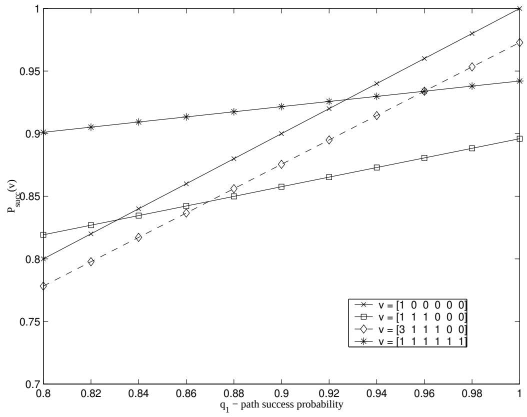

Fig. 3. Comparing different allocation vectors.

In figure 3 we give a numerical example in the case where $r = 3/2$ and $n_{max} = 6$ paths. The probability vector is $\underline{q} = [q_1, 0.8, 0.8, 0.8, 0.8, 0.8]$, where $0.8 \leq q_1 \leq 1$. We plot the probability of success for four allocation vectors, as shown in the legend of the graph. We can see that for $q_1 < 0.92$, the optimal allocation is $\underline{v} = [1, 1, 1, 1, 1, 1]$, whereas for $q_1 > 0.92$ it is $\underline{v} = [1]$. If only 3 paths are available, then for $q_1 < 0.83$, the optimal allocation is $\underline{v} = [1, 1, 1]$, whereas for $q_1 > 0.83$ it is $\underline{v} = [1]$. We also verify that the vector $\underline{v} = [3, 1, 1, 1, 1]$ can never be the optimal.

B. Approximation of P_{succ}

From the analysis presented in the previous section it is evident that neither the optimal number of paths, nor the optimal allocation vector can be calculated in the general case where the probability vector is non-uniform. The main problem is the complexity of P_{succ} in terms of continuity and the required computation time. Since P_{succ} is not continuous because of the presence of unit-step functions in its formula, its derivative is not defined everywhere. Moreover, the time required to calculate P_{succ} is exponential (see equation (13)), which simply means that real-time computation is impossible to achieve.

To address the above problem of P_{succ} evaluation, we will present an approximation of P_{succ} based on the following observations:

- the binomial distribution can be approximated by the normal distribution, and
- the sum of n independent normally distributed random variables follows the normal distribution.

We assume n paths with the path probability vector $\underline{p}$ and the block allocation vector $\underline{v} = [v_i]$. Vector $\underline{p}$ follows an ascending order and, therefore, $v_i \geq v_{i+1}$, because a path with higher failure probability ($p_{i+1} \geq p_i$) cannot receive more blocks than the blocks sent to path with a lower failure probability. Also, without loss of generality, we assume that $v_n = 1$. P_{succ} can be approximated by the following equation, as shown in [11]:

$$P_a(\underline{v}, n) = \frac{1}{2} + \frac{1}{2} \cdot erf\left(\frac{\mu(\underline{v}) - \lceil \frac{1}{r} \sum_{i=1}^{n} v_i \rceil + 1/2}{\sigma(\underline{v})\sqrt{2}}\right), \quad (15)$$

where:

$$\mu(\underline{v}) = \sum_{i=1}^{n} v_i q_i, \sigma(\underline{v}) = \sqrt{\sum_{i=1}^{n} v_i^2 p_i q_i}. \tag{16}$$

For a discussion on the validity of the approximation see [11]. Also, for more details on the normal approximation to the binomial distribution and the condition under which this approximation is satisfactory see [12].

Our goal is to maximize P_a with respect to $\underline{v}$ and n. In this paper, due to limited space, we only present the calculation of the optimal number of paths n^*, assuming a uniform allocation of one block per path, i.e., $v_i = 1$, for $i = 1..n^*$. An optimization technique for both $\underline{v}$ and n is developed in [11]. If one block is sent per path, P_a shows local maxima when the number of paths used is a multiple of b (defined in equation (1)). Thus, the optimal number of paths is found among these local maxima values:

$$n^* = \max_{n=k\cdot b\leq n_{max}}{}^{-1} \left\{ \frac{\mu(n) - n/r + 1/2}{\sigma\sqrt{2}} \right\}. \tag{17}$$

As it can be observed in the previous equation, in order to maximize P_a, it is sufficient to maximize the expression inside the function $erf(\cdot)$ (i.e., the Error Function) in equation (15), since this function is monotonically increasing. In the next section, we show some interesting evaluation results of the derivations presented here.

C. Results and graphs

In this section, we present some results for the case in which one block is allocated to each path. Interesting results are obtained when the probability vector is uniform, i.e., all paths exhibit the same probability of success $q_i = q, i = 1..n_{max}$. In this case, there is a threshold value $q_+(r)$ beyond which P_{succ} is increasing as the number of used paths increases. This threshold is approximated by the following equation:

$$q_+(r) = \frac{1}{2(b+1)} + \frac{1}{r} \tag{18}$$

In figure 4 we can compare q_+ and its approximation described by equation (18). In the case of uniform block allocation on paths bearing the same failure probability, the following can be concluded:

- If $q \geq q_+(r)$, then P_a is ascending for $n > b$ and therefore the optimal number of paths is the number of available paths (all paths should be used). However, we have to take into account that P_{succ} encounters local maxima at positions kb, where $k \geq 1$, so if the number of available paths is n_{max}, then the optimal number of paths is:

$$n^*(r) = b \cdot \lfloor \frac{n_{max}}{b} \rfloor$$

- If $q < 1/r$, then P_a is descending with respect to n and so is the set of local maxima of P_{succ}. The optimal number of paths for this case is:

$$n^*(r) = b$$

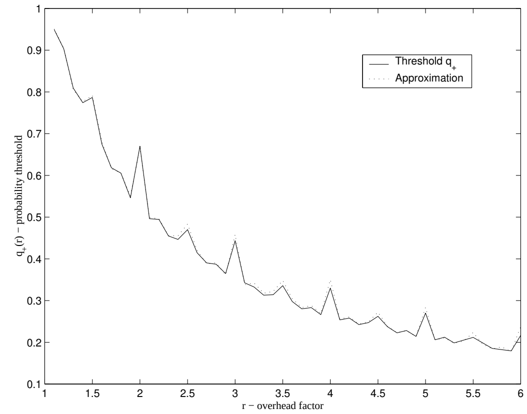

Fig. 4. Threshold q_+ and its approximation.

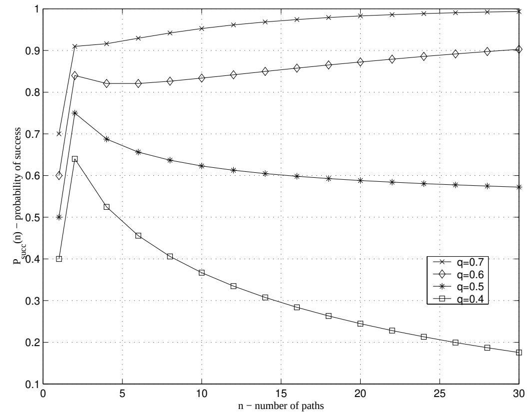

Fig. 5. Behavior of P_{succ} for different values of q.

- If $1/r \leq q < q_+(r)$, then P_a has a minimum at:

$$n_0(r, q) = \frac{1}{2(q - 1/r)}$$

These conclusions can be verified from figure 5, in which we plot P_{succ} (overhead factor $r = 2$) against the number of paths $n = k \cdot b, k = 1..15$, for different values of the path success probability. Incidentally, we note that $q_+(2) \approx 0.7$, whereas from equation (18) we find the exact value: $q_+(2) = 0.67$.

For the general case where the probability vector is arbitrary, we note that if $q_i > 1/r$:

$$\lim_{n\to+\infty} P_{succ}(n) = 1$$

and therefore it is advantageous to use a large number of paths, so as to achieve a probability of success close to 100%.

IV. Conclusions and future work

In this paper, we proposed a new scheme for multipath routing in mobile ad hoc networks. Our goal was to apply multipath techniques in an environment that has continuously changing topology and no infrastructure, so that the typical problems associated with nodal mobility and wireless links (unreliable transmissions, fading, etc.) will be alleviated. We argued that, if the mean time of packet transmission is much smaller than the mean time between variations in network topology, we can fairly assume that the probability that one or more path links fail is constant during the transmission of a packet.

Under these assumptions, we considered the general case of multipath transmission, in which n_{max} disjoint paths are available for a packet transmission. Each path is treated as a pure erasure channel and it is associated with some failure probability p_i, which was defined as the probability that, at transmission attempt time, the path is down. Based on the work done in [2], we used *M-for-N Diversity Coding*. This scheme splits the original packet into N blocks, adds M blocks of overhead (calculated using linear transformations from the original N blocks), and, finally, allocates one block to each one of $N+M$ paths. *M-for-N Diversity Coding* offers protection against at most M lost blocks out of the total $N+M$ blocks. In our scheme, rather than allocating one block per path, we assume an allocation of v_i blocks to path $i, i = 1..n_{max}$. Thus, we show what the optimal distribution of these blocks to the n_{max} disjoint paths should be, so that P_{succ} is maximized.

Given the path failure probabilities, the overhead factor, and the allocation of the original and overhead blocks to the n_{max} paths, we developed an analytical formula for the probability function P_{succ}, namely, the probability that no more than M blocks are lost. This is the probability that the original N blocks can be reconstructed at the destination, and, as a consequence, the transmission is successful. We showed how to maximize P_{succ} (in terms of the number of paths) fast enough, so that the requirement for a real-time recalculation of the optimal solution, due to topology changes, could be met.

Our scheme proposed here offers increased protection against route failures. Under some constraints on the path failure probabilities, it was found that the probability of a successful communication of packets between source and destination increases with the number of used paths. Moreover, this would effectively reduce transmission delay and traffic congestion through load balancing.

The proposed scheme can also be used to enforce error rate QoS requirements, whenever the characteristics of the offered paths make it possible. In that case, we do not have to maximize P_{succ}, but, instead, simply set it to the required probability (indicated by the QoS requirements) and then find the number of paths and the block allocation that satisfies it. This could make real-time data transmission feasible in an environment that is hostile to such type of communication. Moreover, by keeping track of the probability of success and by constantly comparing it with the QoS requirement, we obtain a metric that may be used in order to trigger new route discoveries, for example if P_{succ} tends to drop below the requirements. By extending the definition of the path failure probabilities, we could enforce different classes of QoS requirements, such as maximum delay requirements. This can be done, by simply defining the path failure probability, as the probability that a packet will not arrive on time, i.e., within the maximum delay time, and, as a result, we assume it is lost.

Our goals for future research include:

- evaluation of the proposed scheme when used for achieving load balancing and satisfying delay constraints,
- development of algorithms in order to estimate the probability vector $\underline{p}$ on a real-time basis,
- derivation and optimization of P_{succ} in the case of correlated paths, and
- implementation of our scheme on top of existing routing protocols and comparative performance evaluation.

References

[1] N. F. Maxemchuk, *Dispersity Routing*, Proceedings of IEEE ICC '75, pp. 41.10-41.13, San Francisco, CA, USA, June 1975.

[2] E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo, *Diversity Coding for Transparent Self-Healing and Fault-Tolerant Communication Networks*, IEEE Transactions on communications, vol. 41, no. 11, Nov. 1993.

[3] R. Krishnan and J. A. Silvester, *Choice of Allocation Granularity in Multipath Source Routing Schemes*, IEEE INFOCOM ' 93, p. 322-9, vol.1.

[4] A. Banerjea, *Simulation study of the capacity effects of dispersity routing for fault tolerant realtime channels*, Computer Communication Review (USA) Computer Communication, Oct. 1996.

[5] A. Nasipuri and S.R. Das, *On-Demand Multipath Routing for Mobile Ad Hoc Networks*, IEEE ICCCN 1999, p. 64-70.

[6] D. Johnson and D. Maltz, *Dynamic source routing in ad hoc wireless networks*, in Mobile Computing, T. Imielinski and H. Korth, editors, Kluwer Academic, 1996.

[7] L. Wang, L. Zhang, Y. Shu, and M. Dong, *Multipath Source Routing in Wireless Ad Hoc Networks*, Canadian Conference on Electrical and Computer Engineering, 2000, p. 479-483, vol.1.

[8] S.J. Lee and M. Gerla, *Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks*, ICC 2001.

[9] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi, *On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks*, MobiHOC 2000, p. 150, 3-10.

[10] N. Gogate, D. Chung, S. Panwar, and Y. Wang, *Supporting Video/Image Applications in a Mobile Multihop Radio Environment Using Route Diversity*, IEEE ICC '99.

[11] A. Tsirigos and Z.J. Haas, *Analysis of multipath routing, part 2: Mitigation of the effects of frequently changing network topologies*, submitted for journal publication, 2001.

[12] T. C. Fry, *Probability and its Engineering Uses*, 2nd ed., D. Van Nostrand, Inc., Princeton, N.J., 1965, pp. 229-234.