Practical Relay Networks: A Generalization of Hybrid-ARQ

Bin Zhao, *Student Member, IEEE,* and Matthew C. Valenti, *Member, IEEE*

***Abstract*—Wireless networks contain an inherent distributed spatial diversity that can be exploited by the use of *relaying*. Relay networks take advantage of the broadcast-oriented nature of radio and require node-based, rather than link-based protocols. Prior work on relay networks has studied performance limits either with unrealistic assumptions, complicated protocols, or only a single relay. In this paper, a practical approach to networks comprising multiple relays operating over orthogonal time slots is proposed based on a generalization of hybrid-automatic repeat request (ARQ). In contrast with conventional hybrid-ARQ, retransmitted packets do not need to come from the original source radio but could instead be sent by relays that overhear the transmission. An information theoretic framework is exposed that establishes the performance limits of such systems in a block fading environment, and numerical results are presented for some representative topologies and protocols. The results indicate a significant improvement in the energy-latency tradeoff when compared with conventional multihop protocols implemented as a cascade of point-to-point links.**

***Index Terms*—Block fading, cooperative diversity, hybrid-automatic repeat request (ARQ), relay channel.**

I. Introduction

TRADITIONAL multihop protocols treat wireless networks as a cascade of point-to-point links, with each radio directing its transmission to only a single receiver [1]. While such an approach allows mature technology developed for link-based wired-networks to be leveraged, it ignores the broadcast-oriented nature of radio which implies that protocols should be node-based [2]. If a network is constrained to use only point-to-point links, then the average throughput furnished to each source diminishes to zero as the number of nodes tends to infinity [3]. The fundamental reason for this constriction is that with a uniform traffic pattern, a typical node must expend so much effort forwarding other sources' information that few resources remain to transport its own message. One way to alleviate this limitation is by exploiting mobility in the network, e.g., by having each source transmit to every passing node in the hopes that one of the passing nodes will eventually come close to the destination [4]. A second way to alleviate the limitation is by exploiting the spatial diversity that is present when a node broadcasts to several receivers [5]. The focus of this paper is on practical strategies for realizing the gains promised in [5] when radios may receive different versions of the same message broadcast from several intermediate devices.

A classic example of distributed spatial diversity can be found in early work on the *relay channel* [6]. In the relay channel, a source broadcasts to both a relay and a destination. The relay also transmits information about the same message to the destination. The destination combines the information it receives from both the source and relay, thereby achieving diversity even if each device has only a single antenna. This idea can be generalized to networks with multiple relays that operate in parallel [7] or with interrelay communications [8]. In keeping with [8], we use the term *relay networks* in this paper to describe networks comprising a source, destination, and one or more interconnected relays. Whenever the source or a relay broadcasts, all the other nodes in the network hear the transmission, although the noise and interference could be too high for the message to be correctly decoded. By appropriately coordinating the actions of the source and relays and combining information at the destination, the devices on the network are able to cooperate to convey the message quickly and reliably. This is in stark contrast to what we term *(conventional) multihop* in this paper, where the message is sent over a predetermined route using a cascade of point-to-point links, each requiring only a single receiver to listen to each transmission and, hence, no spatial diversity is present.

In the aforementioned references, little or no constraints are placed on how the nodes cooperate aside from some limitations on transmitter power. Unless otherwise constrained, two impractical requirements emerge when the underlying optimization problem is solved. First, the relays are expected to simultaneously receive and transmit in the same channel, which is not cost effective with the current state-of-the-art in radio technology. Second, simultaneously transmitting nodes are expected to co-phase their transmissions so that they add coherently at a common receiver. While such a beamforming effect is challenging for traditional antenna arrays, it is even more difficult to implement when the antennas are distributed and driven by independent oscillators.

Recent work has imposed additional constraints that eliminate these undesirable network requirements. Høst Madsen [9] and Khojastepour *et al.* [10] constrained the relays to operate in a time-division duplexing (TDD) mode, thereby eliminating the first requirement. Laneman and Wornell added an additional constraint that the source and single relay [11] or multiple relays [12] transmit orthogonally, thereby eliminating the second requirement. Later, Kramer *et al.* [13] rigorously found the capacity for a noncoherent phase fading channel.

Manuscript received October 15, 2003; revised August 1, 2004. This work was supported in part by the Office of Naval Research under Grant N00014-00-0655.

B. Zhao was with West Virginia University, Morgantown, WV. He is now with Efficient Channel Coding, Inc., Brooklyn Heights, OH 44131 USA (e-mail: bzhao@eccincorp.com).

M. C. Valenti is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506-6109 USA (e-mail: mvalenti@wvu.edu).

Digital Object Identifier 10.1109/JSAC.2004.837352

While relaxing these requirements has made the prospect of relaying more feasible than ever, there is a considerable amount of research that must be conducted before complete end-to-end protocols can be designed that both enjoy the diversity benefits of relaying and lend themselves to practical implementation. While much work has focused on the practical use of a single relay or multiple relays that transmit simultaneously (perhaps using a space–time code [12]), little work has been devoted to using multiple relays over orthogonal time slots. When multiple relays are considered, the scheduling of the relays becomes a fundamental issue. The relays must know if and when to transmit and ideally should be able to make these decisions in a distributed fashion.

A viable solution for the relay scheduling problem was proposed by Zorzi and Rao [14], [15] and the resulting protocol termed geographic random forwarding (GeRaF). In GeRaF, the source broadcasts to a collection of potential relays. The node that is closest to the destination (i.e., most geographically advantaged) is selected (in a distributed fashion) to serve as the relay and transmits the message next. The protocol assumes that each node knows its own position, as well as that of the destination, and that a channel contention scheme exists to determine the forwarding node (the details of the contention scheme can be found in [14] and [15]). While GeRaF offers a solution to the relay scheduling problem, it unfortunately does not experience the distributed transmit diversity advantage of the previously described relay networks. This is because each potential relay only receives the transmissions of a single radio, either the source or current relay. Once a new relay is selected, all the nodes in the network flush their memory of prior transmissions and are, therefore, unable to combine information sent from multiple radios.

A diversity effect can be introduced to GeRaF by simply allowing the nodes to maintain previously received information concerning each active message. Each time a message is retransmitted, either from a new node (as in multihop) or from the same node [as in hybrid-automatic repeat request (ARQ)[1]], every node in the relay network will increase the amount of resolution information it has about the message. Once a node has accumulated sufficient information it will be able to decode the message and can act as a relay and forward the message (as in decode-and-forward [17]). This diversity effect can be viewed as a space–time generalization of the time-diversity effect of hybrid-ARQ as described in [18].

In this paper, we present a practical approach to designing wireless ad hoc networks that exploit the spatial diversity that can be achieved with relaying. As shown in the system model presented in Section II, the approach can be considered to be a generalization of hybrid-ARQ, whereby the retransmitted packets could originate from any node that has overheard and successfully decoded the message. We propose a baseline protocol in Section III that we term hybrid-ARQ-based intracluster geographic relaying (HARBINGER) and compare against some other candidate protocols. Section IV uses Monte Carlo integration to analyze the throughput and energy efficiency of these relaying protocols under various system constraints and network topologies. Finally, in Section V, we draw conclusions and propose future research.

[1]Hybrid-ARQ is a combination of forward error correction (FEC) and automatic repeat request (ARQ), whereby the receiver first tries to correct errors, but if it cannot correct all errors it will ask for a retransmission [16].

Before delving into the details of our work, we would like to make a few comments about semantics. Several new terms have emerged in the popular literature that are related to relaying: *Cooperative diversity* [17], *user cooperation diversity* [19], [20], *coded cooperation (diversity)* [21], and *cooperative coding* [22]. Most of these papers involve a twist on relaying, whereby two sources act as relays for each other. However, the term *cooperative diversity* is sometimes applied to relay networks with just a single source [17]. While this paper could be considered to be on the topic of single-source cooperative diversity, we favor the term *relay network* as it is less ambiguous.

II. System Model

Consider a *cluster* of nodes $\mathcal{N} = \{Z_k : 1 \leq k \leq K\}$ consisting of a *source* $Z_s = Z_1$, a *destination* $Z_d = Z_K$, and $K_r = K - 2$ *relays*. Relays are numbered according to their distance to the destination, with Z_2 being the furthest and Z_{K-1} being the closest. Each node has a single half-duplex radio and a single antenna. When any node in $\mathcal{N}$ transmits, all nodes also in $\mathcal{N}$ (but not also simultaneously transmitting) may receive the signal over a block fading channel. As we illustrate later, there is a practical upper limit on cluster size. This limit is due to two fundamental reasons. First, each node in the cluster must expend a nonnegligible quantity of energy to receive and process the message; for a large number of nodes this reception energy could actually exceed the energy consumed by transmitting the RF signal. Second, because nodes that are listening are not free to transmit their own message, channel resources are not quickly reused and, thus, the bandwidth efficiency of the system could suffer.

While small networks (e.g., $K \approx 10$) could consist of just a single cluster (possibly with source, destination, and relays periodically switching roles), larger networks will need to be decomposed into several clusters. Messages that must travel far would be routed from cluster to cluster and a higher level networking protocol will still be needed to handle this routing. However, the networking protocol would only have to route at the cluster-level rather than at the node-level. While this concept is similar to other hierarchical routing protocols like clusterhead gateway switch routing (CGSR) [23], the key difference is that routing within the cluster is now handled implicitly by the retransmission process of the ARQ protocol rather than explicitly by a network-layer routing algorithm.

Two types of relays are possible: *decoding relays*, which must successfully decode the message before forwarding (*decode-and-forward*), and *amplifying relays*, which simply repeat an amplified version of the received signal without first decoding (*amplify-and-forward*) [17]. More generally, relays may adaptively switch between decoding and amplifying modes [24]. Laneman *et al.* [17] indicates that adaptive decode-and-forward strategies offer the same performance as fixed amplify-and-forward. Therefore, in the following, we limit our attention to decode-and-forward relaying, which has the side benefit of permitting a more straightforward exposition. Our approach could be

easily generalized to include amplifying relays, but this would only obscure the main results.

Time is divided into *slots* s, which are of equal duration.[2] During slot s, a node may transmit or receive, but not both. If the cluster is part of a chain conveying messages over long distances, then the source (destination) will need to spend roughly half its time acting as the destination (source) of the previous (next) cluster. This could be accomplished through time division duplexing, e.g., a node could act as source for the current cluster during even s and as destination for the previous cluster during odd s.

The source begins by encoding a b bit message into a codeword of length n symbols. The codeword is broken into M blocks (or bursts), each of length $L = n/M$ and rate $R = b/L$. The code itself could simply be a *repetition* code, in which case all M blocks are identical and each node will *diversity combine* [16] all blocks that it has received. More generally, *incremental redundancy* [16] could be used, whereby each block is obtained by puncturing a rate $r_M = R/M$ mother code. With incremental redundancy, a different part of the codeword is transmitted each time, and after the mth block, a receiver will pass the rate $r_m = R/m$ code that it has until then received through its decoder (*code combining*).

Let $S_m = \{s_1, \ldots, s_m\}$ denote the set of slots over which the first m blocks are sent. While these m time slots need not be contiguous, in the numerical results that we present later we assume that they are. More generally, the time $s_m - s_{m-1}$ between transmissions could be chosen to ensure a desired level of temporal decorrelation and randomized to mitigate interference (*time-hopping*). The set of nodes that transmit during slot s is denoted $\mathcal{K}(s)$. All transmissions are considered to be *broadcast* and, thus, every nontransmitting node in the cluster may receive each transmission. Initially, only the source has knowledge of the codeword and, thus, $\mathcal{K}(1) = \{Z_s\}$. During subsequent slots $s, s \geq 2$, *any* node in the cluster that has successfully decoded the message could re-encode it and transmit the next block of the mother code. The exact composition of $\mathcal{K}(s)$ is determined by the protocol being used, as discussed later.

Let $\mathbf{x}[m] = (x_1[m], \ldots, x_L[m])$ denote the mth block of the codeword. The symbols in $\mathbf{x}[m]$ are normalized to have unity power and, thus, $E\{x_\ell[m]\} = 1$. This block is transmitted by node $Z_k \in \mathcal{K}(s_m)$ with average energy per symbol $\mathcal{E}_k[m]$. Hardware constraints preclude any node from transmitting with symbol energy greater than some maximum value, $\mathcal{E}_{\max}$. For the sake of mathematical tractability, we follow [18] and assume circularly symmetric complex Gaussian symbols are transmitted. Note that while each node in $\mathcal{K}(s_m)$ transmits identical blocks, they do not need to transmit the blocks with equal energy (though in the numerical results that we provide, we assume that they do). More generally, the different nodes in $\mathcal{K}(s_m)$ could transmit different coded sequences, for instance different rows from an orthogonal space–time code [12]. However, this adds to the complexity of the protocol and is outside the scope of the present paper.

The copy of block m that is transmitted by Z_k is received at $Z_j, j \notin \mathcal{K}(s_m)$, with average energy per symbol $\mathcal{E}_{k,j}[m]$. Signal energy decays exponentially with distance such that $\mathcal{E}_{k,j}[m] = (G_{k,j})^2 \mathcal{E}_k[m] = (\lambda_c/4\pi d_o)^2 (d_{k,j}/d_o)^{-\mu} \mathcal{E}_k[m]$, where $G_{k,j}$ is the *channel gain* between Z_k and Z_j, $d_{k,j}$ is the distance between Z_k and Z_j, d_o is a reference distance, λ_c is the wavelength of the carrier, and μ is a path loss coefficient with values typically in the range $1 < \mu < 4$ [25].

Because multiple nodes could be simultaneously transmitting the same block, Z_j receives the superposition of several signals observed through independent block fading channels. In particular, block m is received by Z_j as

$$\mathbf{y}_j[m] = \sum_{k \in \mathcal{K}(s_m)} c_{k,j}[m] \sqrt{\mathcal{E}_{k,j}[m]} \mathbf{x}[m] + \boldsymbol{\nu}_j[m] \quad (1)$$

where $\boldsymbol{\nu}_j[m]$ is a vector of circularly symmetric complex Gaussian noise with independent identically distributed (i.i.d.) components with variance N_o, and $c_{k,j}[m]$ is a unit-power complex fading coefficient that describes the random amplitude and phase fluctuations in the channel between nodes k and j (possibly including the effects of shadowing). We assume that the fading coefficient is constant for the duration of a block and varies from block to block (cf. block fading [26]–[28]). While the fading coefficients may have any arbitrary distribution and correlation (both temporally and spatially), it is common to assume that the coefficients are Rayleigh (or Rician) distributed and independent from both block-to-block and node-to-node [27]. We assume that the fading coefficients are not known to the transmitter, but known to the receiver. As a consequence, it is impossible for the nodes to co-phase their transmissions. Interference will arise if there are other nodes nearby (perhaps associated with a different cluster) transmitting different messages. Due to the Gaussian channel inputs, this interference will also be Gaussian, although the assumption of block fading implies that separate clusters must be synchronized. The exact nature of the out-of-cluster interference can be taken into account by the statistical model of the interference, though this issue is an open problem and outside the scope of the present paper.

Because each node in $\mathcal{K}(s_m)$ transmits the same block, $\mathbf{x}[m]$ can be factored out of the summation in (1) to yield

$$\mathbf{y}_j[m] = \mathbf{x}[m] \sum_{k \in \mathcal{K}(s_m)} c_{k,j}[m] \sqrt{\mathcal{E}_{k,j}[m]} + \boldsymbol{\nu}_j[m]. \quad (2)$$

The corresponding instantaneous signal-to-noise ratio (SNR) can be found by noting that the summation represents an equivalent channel over which the block has been sent. Thus, the SNR of block m at Z_j is

$$\gamma_j[m] = \frac{1}{N_o} \left| \sum_{k \in \mathcal{K}(s_m)} c_{k,j}[m] \sqrt{\mathcal{E}_{k,j}[m]} \right|^2. \quad (3)$$

Note that had the nodes been able to cophase their transmissions, the SNR would be in the form $(\sum |c_{k,j}[m]| \sqrt{\mathcal{E}_{k,j}[m]})^2 / N_o$.

[2]It is sometimes advantageous for the source and relay transmission slots to be of nonidentical length [9], [21], but this leads to an awkward implementation. We conjecture that a similar benefit can be more easily obtained by controlling the relative powers of the source and relay or, in the randomized retransmission protocol that we consider, by using nonidentical transmission probabilities $p_k[s]$.

Due to fading, power control, out-of-cell interference, and the protocol's relay selection, the instantaneous SNR varies from block to block, and we denote the corresponding average SNR by Γ_j.

Let $I(\gamma)$ denote the mutual information between the input and output of a channel with instantaneous SNR γ. For Gaussian noise and inputs (and, hence, Gaussian interference), $I(\gamma) = (1/2)\log_2(1+\gamma)$. Note that since γ is random, so is $I(\gamma)$ and, therefore, a Shannon-sense (ergodic) capacity does not exist [26]. Let $I_j[m]$ denote the mutual information accumulated by node j during the first m transmissions. Under code combining, the system behaves like a set of m parallel Gaussian channels and, thus, $I_j[m] = \sum_m I(\gamma_j[m])$ [18]. Alternatively, under diversity combining the system is a single Gaussian channel with total SNR equal to the sum of the individual SNRs, i.e., $I_j[m] = I(\sum_m \gamma_j[m])$ [18]. Since $\sum_m \log_2(1+\gamma_j[m]) \geq \log_2(1+\sum_m \gamma_j[m])$, code combining is always at least as good as diversity combining and is, therefore, the focus of the remainder of this paper (though we present results in Section IV showing the performance difference).

Node Z_j is in an *outage* after the mth block has been transmitted if $I_j[m] \leq R$. The *outage probability*[3] is then $P_j[m] = \text{Prob}\{I_j[m] \leq R\}$ and can be found by integrating the joint pdf of the m-block channel $p(\gamma_j[1],\ldots,\gamma_j[m])$ over the *outage region* $\{\gamma_j[1],\ldots,\gamma_j[m] : I_j[m] \leq R\}$. We define the *end-to-end outage probability* P_o to be the outage probability at the destination after either all M blocks have been transmitted or a delay constraint of D slots has been reached, whichever comes first.

In a *direct-transmission* system, $K = 2$, and since there is no relay, only the source transmits, $\mathcal{K}(s_m) = \{Z_s\}, \forall m$. When $K > 2$, several relaying strategies are possible. With conventional *multihop*, messages must flow through the cluster as a series of direct transmissions determined *a priori* by a routing algorithm [1]. The destination may not decode the source's direct transmission, even if the instantaneous source-destination SNR is sufficiently high to do so.

If we allow the destination to also "hear" the source, then several other options are possible. First consider a system with $K = 3$ and $M = 2$, which is discussed in more detail in [11], [30]. While the first block is always transmitted by the source, $\mathcal{K}(s_1) = \{Z_s\}$, the second block could again be transmitted by the source or it could instead be transmitted by the relay $Z_r = Z_2$ provided that it decoded the first block, i.e., if $I_r[1] = I(\gamma_{s,r}[1]) > R$. If the relay is in an outage ($I_r[1] \leq R$), then the transmission ceases after the first block and an end-to-end outage occurs if the source-destination link was in an outage ($I_d[1] = I(\gamma_{s,d}[1]) \leq R$). Otherwise, the relay will transmit and an end-to-end outage occurs if the parallel channels from source and relay to destination are in an outage, $I_d[2] = I(\gamma_{s,d}[1]) + I(\gamma_{r,d}[2]) \leq R$.

A modest amount of adaptability can be introduced by using channel state information (CSI) to guide which of the two nodes transmits the second block [17], [24]. In particular, if the source knows that the relay was in an outage during the first block, then it could transmit the second block instead. Furthermore, if the source and relay know that the relay-destination SNR is less than the source-destination SNR (i.e., $\gamma_{r,d}[2] \leq \gamma_{s,d}[2]$)), then the source could transmit the second block, even if the source-relay link was not in an outage. While these adaptive techniques could be extended to permit multiple relays ($K > 3$) and more transmitted blocks ($M > 2$), the need for each node to have *a priori* knowledge of the CSI of various channels and for the cluster to coordinate transmissions quickly makes this approach unwieldy. The solution that we advocate for selecting which node in a multiple relay network transmits a particular block is to embedded the selection process into the hybrid-ARQ protocol, as discussed in the next section.

III. Hybrid-ARQ Based-Relaying Protocols

A system that used FEC only, rather than a combination of FER and ARQ, would transmit all M blocks of the codeword before moving on to the next message. This is wasteful of network resources, as often the destination may be able to successfully decode after receiving some earlier block $m < M$. On the other hand, with hybrid-ARQ the cluster will transmit new blocks of the codeword until one of the following occurs [18]: 1) the destination successfully decodes the message and signals back with a positive acknowledgment (ACK), which we assume for the sake of exposition is conveyed over an error- and delay-free feedback channel; 2) all M blocks have been transmitted, $m = M$; or 3) a maximum latency has been exceeded, $s > D$ (M and D constitute a *rate constraint* and a *delay constraint*, respectively).

First, consider how hybrid-ARQ can be used to effectively determine the set $\mathcal{K}(s)$ of transmitters. Let $\mathcal{D}(s)$ denote the set of nodes with knowledge of the codeword at the start of slot s; we call $\mathcal{D}(s)$ the *decoding set* and its members *decoding nodes*.[4] Under decode-and-forward relaying, only decoding nodes may transmit and, thus, $\mathcal{K}(s) \subseteq \mathcal{D}(s)$. Initially, the decoding set contains only the source, $\mathcal{D}(s_1) = \{Z_s\}$. After the first block and at the start of the mth block, the decoding set will contain the source plus all relays that have previously accumulated enough information to decode successfully, i.e., $\mathcal{D}(s_m) = \{Z_s, Z_k : I_k[m-1] > R\}$. Once a relay is added to the decoding set, it is never taken out, so $|\mathcal{D}(s)| \geq |\mathcal{D}(s-1)|$, where $|\mathcal{X}|$ is the cardinality of set $\mathcal{X}$. Once a node is in the decoding set, it no longer needs to listen and, therefore, does not expend any more energy receiving and processing additional blocks of the codeword (aside from listening for ACK messages).

The source begins by broadcasting the first block during the first slot ($s_1 = 1$). The destination can decode the message if $I_d[1] > R$ and, if successful, will broadcast an ACK. Otherwise, a retransmission will be necessary. After the source's initial broadcast, some of the relays may have successfully decoded the transmission, namely those for which $I_k[1] > R$. These decoding relays are included in $\mathcal{D}(2)$. During the next transmission slot $s \geq 2$, *any* node in $\mathcal{D}(2)$ can transmit the second block of the codeword. But which? The answer to this question rests

[3]This is also termed *information outage probability* [27] and *outage event probability* [17] and is related to the *outage capacity* [29].

[4]The decoding set concept was proposed in [12] for a nonadaptive system and, thus, with no dependence on the slot s.

in the design of the hybrid-ARQ protocol that governs the behavior of the relay network. Below, we discuss several candidate protocols, which are compared numerically in Section IV.

A. HARBINGER

As in [14], assume that each node in the network has an accurate estimate of its own position, as well as the position of the source and destination. It can measure its own position with an onboard global positioning system (GPS) receiver, and the header of each message could contain the location of the source and destination. Given this position information and knowledge of the channel model, the node could estimate the average SNR of the channel between it and the destination. Equivalently, if nodes are not supplied with a GPS receiver, they could still measure the average SNR to the destination by keeping track of the strength of the ACK packets (assuming reciprocal channels).

Given this information, the relaying node can be selected using a protocol similar to GeRaF [14]. Like GeRaF, the protocol is designed so that the node in the decoding set $\mathcal{D}(s)$ that is closest to the destination will transmit the next block of the message. For our isotropic propagation model, picking the node closest to the destination is equivalent to picking the one whose channel to the destination has the highest *average* SNR. How this node is selected is irrelevant to the numerical results that we present in Section IV. In practice, the protocol could begin with the source sending out block $m = 1$. Following the transmission of this block, the network enters a contention period. The contention process is similar to the request-to-send–clear-to-send (RTS-CTS) handshaking common to traditional networks with the key distinction being that the contention occurs *after* the block has been transmitted, rather than before. The contention interval is divided into K_r subintervals (which we call windows), one for each relay. During the first window, relay Z_{K-1}, which is closest to the destination, sends an ACK packet if it is in $\mathcal{D}(s)$, otherwise, it will remain silent. This process continues so that during window $n = 1, \ldots, K_r$, relay Z_{K-n} sends an ACK packet if and only if it is in $\mathcal{D}(s)$. Once a node has sent an ACK signal, the network will then know which node is closest to the destination and that node is free to send the second block (with another, identical contention process run after that block is sent). If no node sends an ACK during the contention period, then the second block will simply be sent by the source.

The protocol described above requires that each relays be assigned a unique window during the contention period, and assigning relays to windows will involve a certain amount of overhead that could be undesirable in the presence of mobility (though perfectly acceptable for applications with low mobility, such as sensor networks). An alternative to assigning a specific relay to each window is to assign zones to each window, as was done in [14]. Window n would be associated with a minimum $d_{\min}[n]$ and maximum $d_{\max}[n]$ range and any node whose distance to the destination falls between these two ranges will signal with an ACK. If the number of zones is large compared with the number of relays, then the probability of collision (multiple relays in the same zone) will be small. When collisions occur, the system could either enter into a secondary contention resolution process or else could allow the multiple nodes to simultaneously transmit over the equivalent channel defined by (2).

While this protocol has much in common with GeRaF, there is a crucial difference. With GeRaF, once the relay node is selected, all the other nodes in the network flush their memory of the message. The system then starts over with the newly selected relay behaving as if it was a new source. In contrast, we propose that the relays maintain information about the message and do not flush away this information until the destination successfully decodes the message. Thus, the relays and destination can combine information sent by not only the source, but also by other relays. This provides a transmit diversity effect that GeRaF does not possess. Also, GeRaF does not use hybrid-ARQ, while our protocol does. To distinguish our protocol from GeRaF, we give it the descriptive name *HARBINGER*.

B. Variations on HARBINGER

The baseline HARBINGER protocol described above is designed to select the relay that is closest to the destination, but other strategies are worth considering. One possibility is to pick the relay from the decoding set with the highest *instantaneous SNR* at the destination. We call this variation *instantaneous-relaying* for brevity. This strategy is in contrast with HARBINGER, which in an isotropic propagation environment, picks the relay with the highest *average SNR* at the destination. Because instantaneous-relaying requires knowledge of the current instantaneous SNRs, it is not nearly as practical as HARBINGER. However, it is informative to see if there is any benefit to using instantaneous SNR as the criterion for selecting the relay node.

Another option is to randomize the relay selection process, which eliminates the need for a contention scheme. During time slot s, each node $Z_k \in \mathcal{D}(s)$ will transmit with probability $p_k[s]$. We call this scheme *random-relaying* for short. Because there is no contention scheme, collisions cannot be prevented. However, by picking $p_k[s]$ to be sufficiently small, the probability of collision can be made arbitrarily low at the cost of increased end-to-end latency. If $p_k[s]$ is a constant across all nodes and all slots, then there is no guarantee that the relay that is selected is a good one. Furthermore, as the size of the decoding set grows, the probability of collision increases. These problems can be alleviated by adapting the value of $p_k[s]$. For instance, the value could be scaled by the size of the decoding set as $p_k[s] = p_t/|\mathcal{D}(s)|$, where p_t is the transmission probability when there is only one node in the decoding set. Furthermore, position location could be used to influence the value of $p_k[s]$, with nodes closer to the destination given larger values than nodes that are located further away.

C. Comparison With Multihop

With multihop, the message must flow through the cluster following a series of direct peer-to-peer connections that are determined *a priori* by a routing algorithm. Without loss of generality, we assume that under multihop the message must flow through *all* K_r relays before reaching the destination and that the relays are indexed in the order that they are used. Under multihop, only the *next* node $Z_{|\mathcal{D}(s)|+1}$ not yet in the decoding set receives the transmission, while with relaying *all* nodes not

yet in the decoding set $\{Z_k \notin \mathcal{D}(s)\}$ receive. With multihop, all relays in the cluster must eventually decode the message, $\mathcal{D}(s_M+1)=\mathcal{N}$, but with relaying, it is irrelevant which relays have successfully decoded; all that matters is if the destination was able to decode successfully, i.e., $Z_d \in \mathcal{D}(s_M+1) \subseteq \mathcal{N}$. With the proposed relaying protocols, relays that are repeatedly in an outage are bypassed, thereby eliminating potential bottlenecks. Furthermore, a network-layer protocol is not needed to preselect the transmission path, rather the "path" selection is embedded into the ARQ mechanism (although we argue that the term *path* becomes meaningless). Also, power/range control becomes less important in a relaying network. In a multihop network, if the transmit power is too high, then the extra energy is wasted. However, if the power is set too high in a relay network then intermediate relays will simply be "leapfrogged" and, therefore, won't need to be used.

IV. Numerical Results

In this section, we compare the performance of the three deterministic protocols discussed in the last section (HARBINGER, instantaneous-relaying, and multihop), as well as random relaying. To better illuminate certain characteristics, we impose some additional constraints on the network. Note that these conditions are imposed to highlight certain behaviors and that the mathematical model presented in Section II are still valid without these conditions. First, we only consider performance within a single cluster, treating out-of-cluster interference as additional Gaussian noise. The three deterministic protocols signal over contiguous time slots, so $s_m = m$ and $D = M$, i.e., the delay and rate constraints are identical. The channel is Rayleigh block fading, and the fading is independent over time and space. We note that this is a pessimistic assumption, and that the relaying protocols will exhibit an even more drastic improvement over multihop when the fading is correlated in time (since then spatial diversity will dominate). All of the deterministic protocols are able to perfectly resolve contentions, and so only one node transmits at a time, i.e., $|\mathcal{K}(s)| = 1$. For purposes of comparison, the random relaying protocol also operates with exactly one node transmitting during each time slot, but the node is chosen at random from the decoding set. We assume that all nodes in the network transmit with identical energy $\mathcal{E}_k[m] = \mathcal{E}_s$. In most cases, the topology is a *line network* comprising a set of K_r relays spaced equally along the line between source and destination, though we also consider a clustered line network.

Monte Carlo integration is used to generate numerical results for the relaying protocols, while closed form solutions were used for multihop whenever possible. For the results shown in the plots, the block/burst rate is $R = 1$, transmit frequency $f_c = 2.4$ GHz, path loss coefficient $\mu = 3$, reference distance $d_o = 1$ m, and source-destination are separated by 100 m. Code combining is assumed except in Figs. 6 and 7, which compare diversity combining with code combining. We begin by eliminating any constraint on rate and delay, i.e., $M \to \infty$ and $D \to \infty$, focusing on the tradeoffs between energy, throughput, and latency. In the final subsection, we consider finite-rate/delay constraints, which give rise to a nonzero outage probability at the destination.

A. Throughput Analysis

As in [18], we would like to adapt the renewal-reward theorem of [31] to compute bounds on throughput. We first define the following random variables.

- $\mathcal{R}$ A random reward, which equals R if the packet is successfully decoded by the destination and zero, otherwise.
- $\mathcal{T}$ The time (in number of slots) spent attempting to transmit an arbitrary message (until either success or until the delay/rate constraints expire).
- $\mathcal{M}$ The total number of blocks transmitted for an arbitrary message until either success or the constraints expire.

Under these definitions, the system throughput is

$$\eta = \frac{E[\mathcal{R}]}{E[\mathcal{T}]} \tag{4}$$

in units of messages per slot. When $D, M \to \infty, E[\mathcal{R}] \to R$. Furthermore, when exactly one node transmits in each slot, $\mathcal{T} = \mathcal{M}$ and so the average delay is equal to the average number of transmitted blocks.

With multihop, messages are passed sequentially through peer-to-peer links. If the nodes are equally spaced and the propagation environment isotropic, then the links behave identically and the end-to-end performance can be assessed in terms of the performance of any one link. In particular, the average delay for the ith hop $E[\mathcal{T}_i] = E[\mathcal{M}_d]$, where $E[\mathcal{M}_d]$ denotes the expected number of blocks transmitted for an arbitrary message in a point-to-point direct link. Correspondingly, the delay of multihop over an equally spaced line network is the accumulation of delay components at each individual hop, i.e., $E[\mathcal{T}] = (K-1)E[\mathcal{M}_d]$. A derivation of $E[\mathcal{M}_d]$ is given in the Appendix. Therefore, the throughput under the given constraints is

$$\eta = \begin{cases} R/E[\mathcal{M}], & \text{for relaying} \\ R/((K-1)E[\mathcal{M}_d]), & \text{for multihop} \end{cases}. \tag{5}$$

Fig. 1 shows the throughput of the different protocols for a line network as a function of transmit SNR $\mathcal{E}_s/N_o$ for $K_r = \{0, 1, 10\}$ relays, where $K_r = 0$ corresponds to a direct transmission link (all protocols behave the same when there are no relays). At low SNR, HARBINGER is slightly better than multihop, and multihop actually outperforms both instantaneous relaying and random relaying. This suggests that using the instantaneous SNR to the destination as the metric to select the relaying node is not a productive strategy, since it ignores the interrelay SNRs and is still unable to predict future SNRs. Thus, HARBINGER, with its use of average SNRs (through geographic location) is the most efficient protocol under these conditions. At low SNR, the performance of random relaying is rather poor, indicating that random relay selection is not sophisticated enough to provide meaningful gains. At high SNR, the throughput of multihop begins to saturate due to the

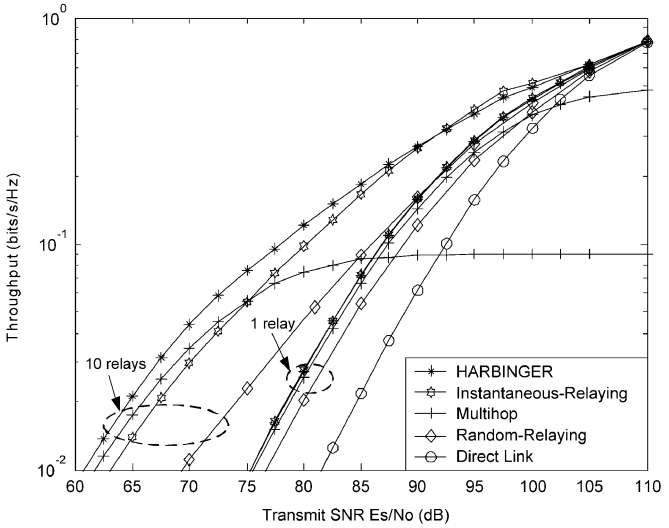

Fig. 1. Throughput of a line network as a function of the per-burst transmit SNR for $K_r = \{1, 10\}$ equally spaced relays without a delay constraint. Results for direct transmission link $(K_r = 0)$ are also shown. All protocols use code combining.

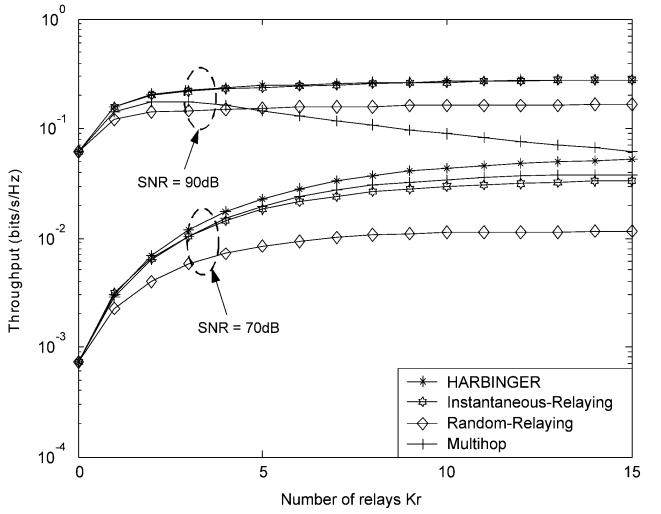

Fig. 2. Throughput of a line network as a function of the number of relays for two different per-burst transmit SNRs.

requirement to transmit over all of the relays and the resulting bottleneck effect. At high SNR, even random relaying outperforms multihop and the performance difference of the different relaying protocols becomes less pronounced. This is because at high SNR, the message is often correctly decoded by the destination after just one or two transmissions and so the choice of relay is less important.

Fig. 2 shows the throughput of a line network as a function of the number of equally spaced relays for two different transmit SNR's, $\mathcal{E}_s/N_o = \{70, 90\}$ dB. We observe that the throughput of all relaying protocols monotonically increase with the number of relays. Furthermore, it is rather interesting, although not unexpected, to notice that the throughput of multihop initially increases with the number of relays, but then decreases as more and more relays are added. The initial increase in throughput for multihop can be attributed to the decrease in the interrelay distances which decreases the delay

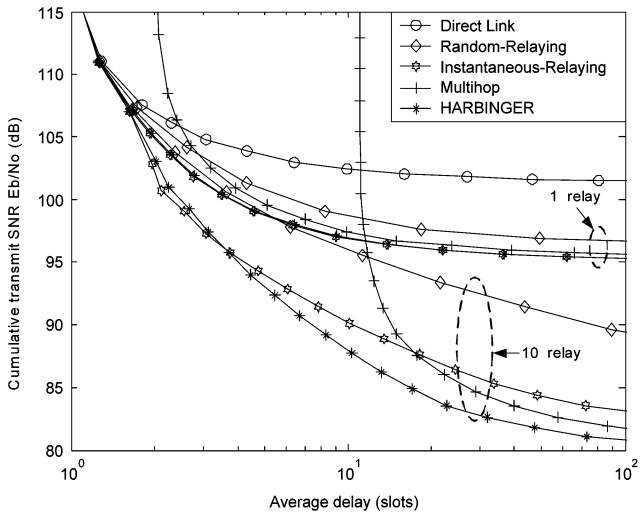

Fig. 3. Cumulative transmit SNR $\mathcal{E}_b/N_o$ as a function of average delay in a line network with $K_r = \{1, 10\}$ relays. Results for direct transmission link $(K_r = 0)$ are also shown.

$E[\mathcal{M}_d]$ of each hop. However, if too many relays are added, then the $1/K$ term in (5) begins to dominate and the throughput becomes inversely proportional to the number of relays. This effect is more pronounced at high SNR. One could argue that the performance of multihop could always be improved by selecting a new route that uses fewer relays. However, the beauty of relaying is that it will do this automatically without needing to adjust the route since relaying is less sensitive than multihop to the number of relays.

B. Energy-Delay Tradeoff

In order to determine the total amount of energy required to convey a message bit from end-to-end, one must take into account not only the transmitted energy per symbol $\mathcal{E}_s$ but also the number of blocks that are transmitted $E[\mathcal{M}]$. Applying renewal-reward theorem, the average *cumulative* transmit energy is

$$\mathcal{E}_b = \frac{\mathcal{E}_s E[\mathcal{M}]}{E[\mathcal{R}]}. \tag{6}$$

Rather than representing the energy transmitted by any *single* node, $\mathcal{E}_b$ characterizes the energy consumed by the *entire* cluster by enumerating the total number of transmitted blocks per correct message without regard to which nodes transmitted the blocks. Without delay/rate constraints and when one node transmits per slot, the required transmit energy for different protocols becomes

$$\mathcal{E}_b = \begin{cases} \mathcal{E}_s E[\mathcal{M}]/R, & \text{or relaying} \\ \mathcal{E}_s (K-1) E[\mathcal{M}_d]/R, & \text{for multihop} \end{cases}. \tag{7}$$

Fig. 3 shows the transmit energy efficiency $\mathcal{E}_b/N_o$ as a function of average delay for the four different protocols and $K_r = \{0, 1, 10\}$ relays. As expected, both instantaneous relaying and HARBINGER are always more efficient than random relaying. Although relaying is always more efficient than direct-transmission, multihop is actually worse under low average delay. This is again due to the bottleneck created when using multihop that cannot be overcome by simply increasing power; instead, a new

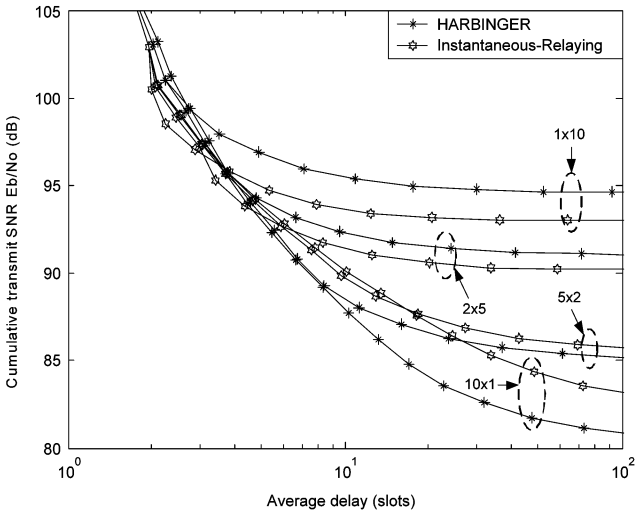

Fig. 4. Minimum cumulative transmit SNR required as a function of average delay for a $u \times v$ line network comprising u equally spaced groups of v relays each.

route would need to be created but the process of creating a new route could in itself add to the latency. When a relatively large delay is allowed, the energy efficiency of both multihop and relaying is significantly improved over direct-transmission. For instance, with one relay, random relaying provides a 5-dB gain at an average delay of 11 over direct-transmission, while multihop, instantaneous relaying, and HARBINGER have a 7-dB gain over direct-transmission. With ten relays, random relaying gains 11 dB over direct-transmission, instantaneous relaying gains 18 dB, multihop gains 19 dB, and HARBINGER gains more than 20 dB. In general, the energy efficiency is improved by allowing longer average delay. This agrees with Caire's assertion that "the longer we wait the more we gain" [18].

C. Effect of Network Topology

While Fig. 3 shows that HARBINGER has the best tradeoff between energy efficiency and delay among the four protocols in an equidistant line network, is it the best protocol in any arbitrary topology? When the nodes are homogeneously spaced along the line, then a macrodiversity effect prevails. But what if nodes bunch up in such a way that microdiversity dominates over macrodiversity? To demonstrate the impact of the homogeneity of the network, we consider a generalized line network where the relays collect into u equally spaced groups each containing v relays. Nodes within a group are spaced close enough together that they all have the same channel gain to the source, destination, or another group. However, they are far enough apart that they experience independent fading. For fair comparison, $u \times v = 10$ for each case. Therefore, there are four possible network configurations: $u \times v = 1 \times 10, 2 \times 5, 5 \times 2$, and 10×1, where 10×1 corresponds to an equidistant line network.

Fig. 4 shows the cumulative transmit energy $\mathcal{E}_b/N_o$ required for HARBINGER and instantaneous relaying under the four topologies. The equidistant line (10×1) has the best efficiency among the five network topologies, while the performance with a single group (1×10) has the worst. When the network contains a small number of groups, instantaneous relaying outperforms HARBINGER. As the number of groups increases, the advantage of instantaneous relaying over HARBINGER diminishes until eventually HARBINGER is better. With just one or two groups, the transmit microdiversity effect dominates, which favors the use of instantaneous channel estimates. However when there are more groups that are more sparsely populated and further apart, the differences in path loss begin to dominate, and HARBINGER is better able to exploit opportunities for macrodiversity. We can conclude from Fig. 4 that macrodiversity is more important than microdiversity and, thus, it is worthwhile to carefully position relay nodes rather than randomly scattering them.

D. Total Energy Consumption

While the use of more sophisticated relaying protocols results in a reduction of required *transmit* energy, this benefit must be weighed against the extra costs. Perhaps the most critical issue is that now *all* nodes that are not yet in the decoding set must receive every transmission, as opposed to multihop which requires that only *one* node receives. Thus, a fair comparison between relaying and multihop should also account for the energy a node consumes when it *receives* a symbol, i.e., the energy dissipated by the circuits that detect and decode the block. By taking into account the costs to receive a message, we can generalize the definition of cumulative energy dissipation by first defining $\mathcal{E}_r$ as the energy consumed by the *receiver* when detecting and processing a signal. Then the total energy consumed by both transmitting and receiving becomes

$$\mathcal{E}_b^a = \frac{1}{R}\left(\mathcal{E}_s E[\mathcal{M}] + \mathcal{E}_r E\left[\sum_s (K - |\mathcal{D}(s)|)\right]\right) \tag{8}$$

for relaying and

$$\mathcal{E}_b^a = \frac{1}{R}(\mathcal{E}_s + \mathcal{E}_r)(K-1)E[\mathcal{M}_d] \tag{9}$$

for multihop.

In general, it is difficult to select appropriate values for $\mathcal{E}_r$, as this is a highly implementation dependent parameter. Instead, a better way to assess the impact of receive energy dissipation is to find the ratio of transmitter versus receiver energy consumption $\mathcal{E}_s/\mathcal{E}_r$ for which relaying outperforms multihop. This ratio can be found by equating (8) with (9) and solving for $\mathcal{E}_s/\mathcal{E}_r$

$$\frac{\mathcal{E}_s}{\mathcal{E}_r} = \frac{(K-1)E[\mathcal{M}_d] - E\left[\sum_s (K - |\mathcal{D}(s)|)\right]}{E[\mathcal{M}] - (K-1)E[\mathcal{M}_d]}. \tag{10}$$

Fig. 5 shows this minimum ratio as a function of the number of relays in a line network for a variable number of equally spaced relays and two transmit SNRs. We focus on HARBINGER since it is consistently the best protocol for this topology. Each SNR curve shows a breaking point; systems with a $\mathcal{E}_s/\mathcal{E}_r$ ratio above the curve favor HARBINGER, while systems below the curve favor multihop. At transmit SNR of 90 dB, the minimum ratio first increases with the number of relays, and then decreases. This agrees with a similar behavior for the throughput of multihop in Fig. 2. The initial increase in $\mathcal{E}_s/\mathcal{E}_r$ indicates that HARBINGER becomes less advantageous over multihop because more energy is dissipated to receive each

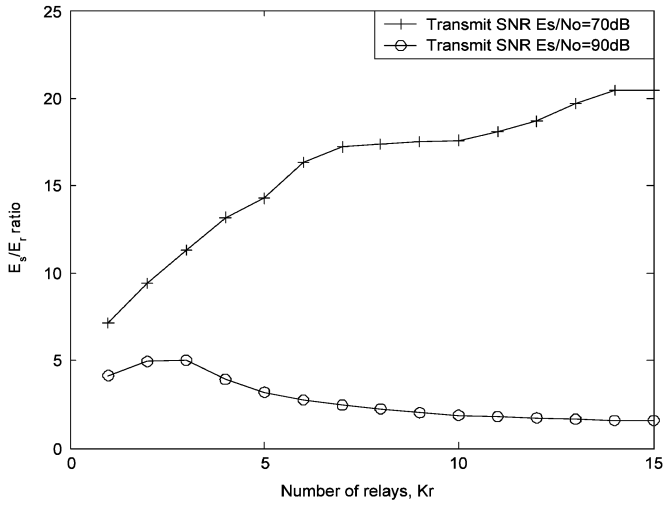

Fig. 5. Minimum required ratio of transmit versus receiver energy dissipation per symbol for HARBINGER to outperform multihop in an equidistant line network with K_r relays.

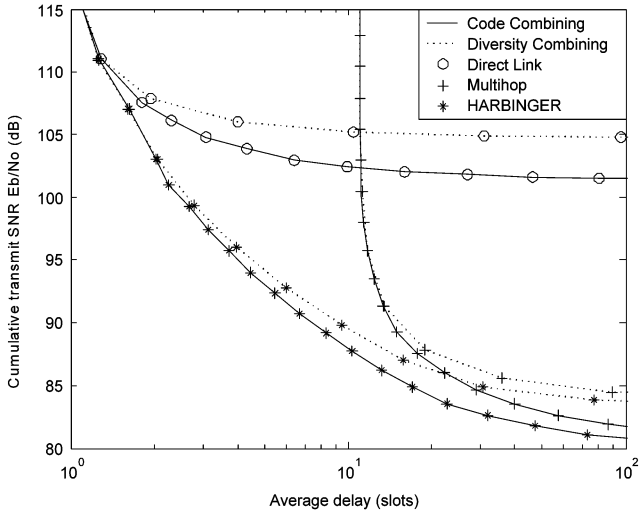

Fig. 7. Cumulative transmit SNR $\mathcal{E}_b/N_o$ of code- and diversity combining as a function of average delay in a line network with $K_r = \{1, 10\}$ relays.

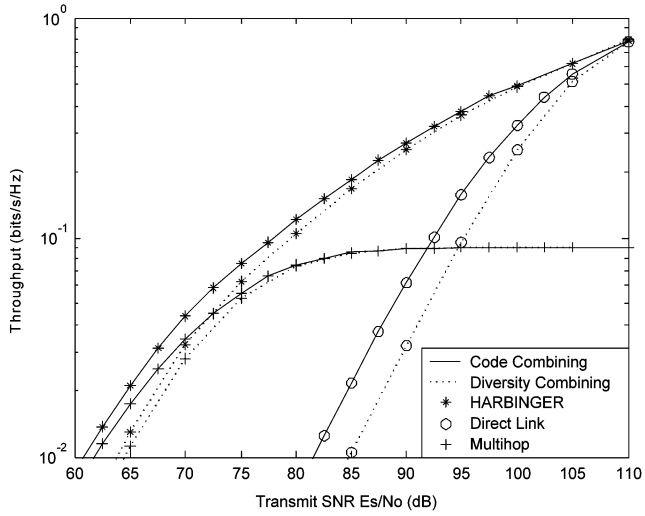

Fig. 6. Throughput of code- and diversity combining in an equidistant line network with $K_r = \{1, 10\}$ relays as a function of the per-burst transmit SNR.

transmission when the number of relays increases. With too many relays, although HARBINGER spends more energy in receiving, the bottleneck effect degrades the energy efficiency of multihop at a rate much faster than that of HARBINGER, resulting in a decrease in the minimum ratio.

E. Diversity Combining Versus Code Combining

Relaying with incremental-redundancy and code combining outperforms that with repetition coding and diversity combining. However, code combining is more complex than diversity combining, and we wish to see if the extra complexity required by code combining is justified. In Figs. 6 and 7, we compare the throughput and energy efficiency of diversity combining versus code combining with HARBINGER and multihop. We observe that at low SNR, code combining has almost twice the throughput of diversity combining and at large delay it is 2–3 dB more efficient. However, at relatively high transmit SNR or small delay, the advantages of code combining become marginal. Thus, for applications requiring low latency, diversity combining is a very attractive alternative to code combining.

F. Finite-Delay Constraint

While the previous discussion has focused on the performance without a constraint on delay D (or, equivalently, on M), practical systems must often impose hard deadlines. If a message arrives after time D, then its content is no longer useful and so the system should abort any further attempt to transmit the message. The main implication of finite D is that now the end-to-end outage probability P_o is nonzero. This in turn influences the throughput and tradeoff between energy consumption and average delay since the expected random reward becomes $E[\mathcal{R}] = R(1 - P_o)$. However, as long as P_o is sufficiently small (e.g., 10^{-2}), the impact on throughput, energy efficiency, and average delay becomes negligible since then $(1 - P_o) \approx 1$. Since it is not attractive for systems to operate at high outage probabilities, the key issue is how the delay constraint D influences the outage probability.

Figs. 8–11 show the outage probability of different relaying protocols as a function of delay constraint D for $K_r = 1$ and 10 relays and $\mathcal{E}_s/N_o = 70$ and 90 dB under code combining hybrid-ARQ. In each case, the outage probability remains close to unity until a particular threshold on delay is reached, at which point the curves begin to rapidly decrease with increasing D. The curves are steeper for a large number of relays or large SNR. The protocol has an impact on the steepness, with deterministic relaying having the steepest descent. Multihop has almost the same steepness as deterministic relaying, but random relaying has a significantly less steep descent. We can see that random relaying is worse than multihop for low SNR or just one relay. However, Fig. 11 shows an interesting result that under finite D, random relaying is actually superior to multihop with ten relays, high SNR, and outage probability above $P_o \approx 3 \times 10^{-4}$. However, due to the shallow slope of random relaying, the curves cross at $P_o \approx 3 \times 10^{-4}$, at which point multihop becomes superior in terms of outage probability. We observed in our simulations that the throughput and energy efficiency under finite D are

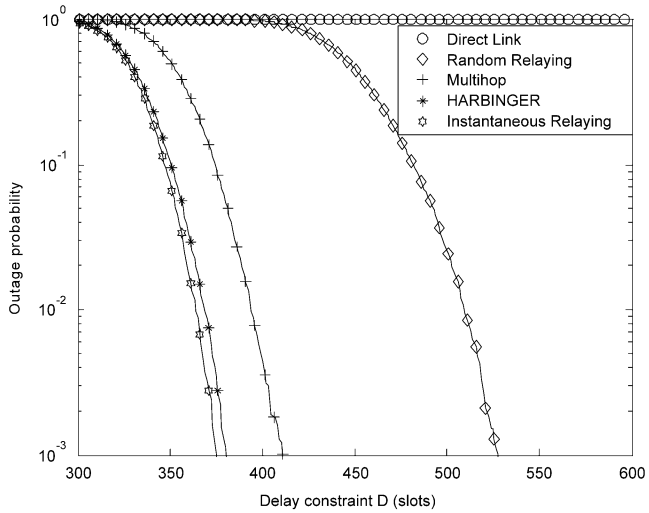

Fig. 8. Outage probability of different relaying protocols as a function of delay constraint for a single relay line network with transmit SNR $\mathcal{E}_s/N_o = 70$ dB and code combining. A threshold in the outage probability of direct link appears around $D = 1400$ (not shown).

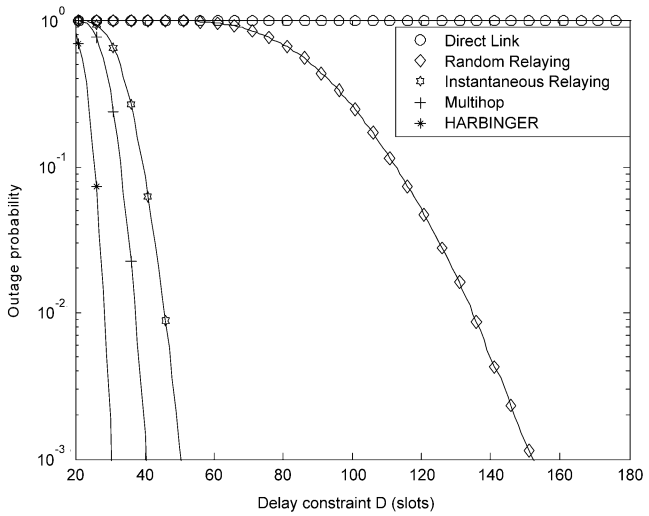

Fig. 9. Outage probability of different relaying protocols as a function of delay constraint for a ten relay line network with transmit SNR $\mathcal{E}_s/N_o = 70$ dB and code combining.

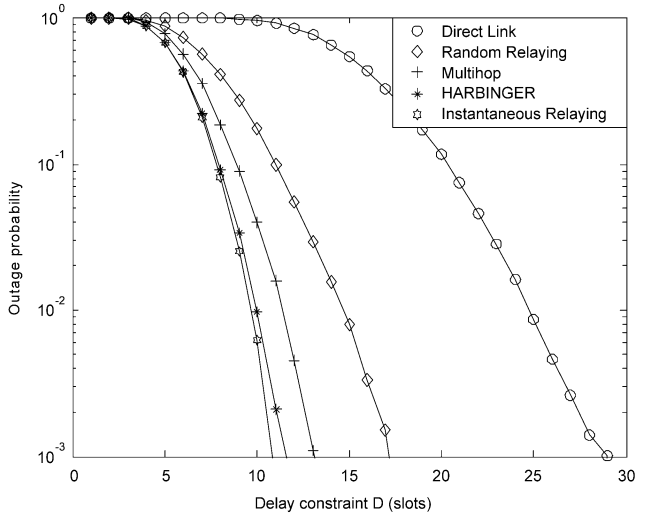

Fig. 10. Outage probability of different relaying protocols as a function of delay constraint for a single relay line network with transmit SNR $\mathcal{E}_s/N_o = 90$ dB and code combining.

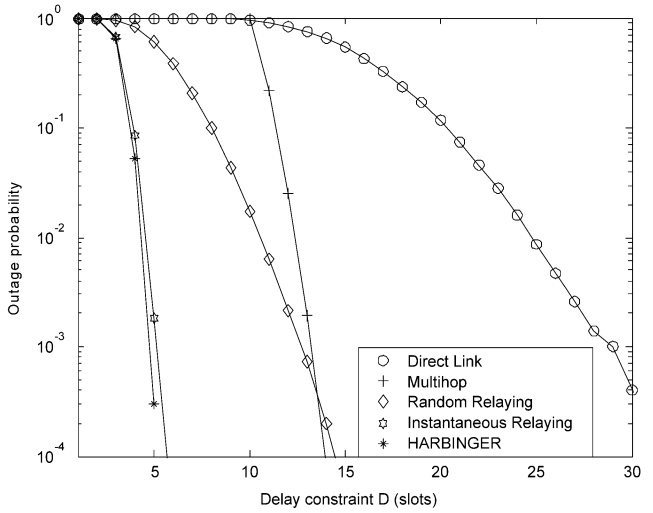

Fig. 11. Outage probability of different relaying protocols as a function of delay constraint for a ten relay line network with transmit SNR $\mathcal{E}_s/N_o = 90$ dB and code combining.

nearly identical to those of infinite D provided that D is above this threshold by some margin. Therefore, we do not reproduce curves for throughput and energy efficiency for finite D.

V. Conclusion

A practical way to implement relay networks is to generalize the concept of hybrid-ARQ. In contrast with point-to-point hybrid-ARQ, the retransmissions do not need to come from the original source; instead they could come from any relay that overhears and decodes earlier transmitted blocks. This provides a spatial-diversity effect that supplements the time-diversity already present in conventional hybrid-ARQ. The diversity is achieved without requiring that relays co-phase their transmissions. Relaying can offer a better tradeoff between throughput, energy consumption, and delay as compared with conventional multihop. Furthermore, relaying can react to changing topologies and channel conditions much faster than multihop, as new routes do not need to be explicitly calculated. The relaying protocols discussed in this paper are truly cross-layer, combining the mechanisms of medium access control (MAC) and routing. Just as a point-to-point hybrid-ARQ does not need to select the code rate in advance, generalized-ARQ does not need to select a route.

The cost of the proposed relaying schemes is that now more than just one radio listens to each transmission and, therefore, the nonnegligible costs associated with reception must be taken into account. This implies that for each scenario there will be an upper limit on the number of relays that should be used. The MAC protocol is now more complicated, since it needs to provide a mechanism for relay selection. The performance is sensitive to the topology, and the nodes should ideally be evenly spread out to maximize the macrodiversity effect. While most of the results in this paper were for code combining, which is

rather complicated to implement, we found acceptable performance even when using less complex diversity combining.

The goal of this paper has been to provide a general framework for studying the information-theoretic performance limits of relay networks that are implemented using generalized hybrid-ARQ. While we believe that this paper represents a significant contribution in this area, there is still much work that remains to be completed. Although this paper focused on energy efficiency, many networks contain devices with finite energy reserves and, thus, the performance under such energy limitations needs to be studied [32]. With finite-energy sources, "hot-spots" become a problem, as some nodes that are in good locations tend to burn out quickly; the protocol will need to be modified to mitigate this problem. While we have looked at a few representative topologies, more should be considered. Networks with more than one source should be considered, as should networks comprising multiple clusters. More sophisticated channels with Rician fading and blocks that are correlated in time (and possibly even in space) should be considered. While this paper has focused on capacity approaching coding with unconstrained (Gaussian) input symbols and infinite block length, the performance when the modulation and block length are constrained should be further studied. A more thorough investigation of the MAC protocol should be conducted that studies the impact of lost ACK packets and suggests rules for making the system robust when ACK packets are lost.

The numerical results presented in this paper used Monte Carlo integration, but closed form analytical expressions would allow the aforementioned effects to be evaluated much more quickly. While such expressions will be difficult to find under the current assumptions, there is some hope for a more analytical treatment if certain constraints are imposed. In particular, if the channels are assumed to be additive white Gaussian noise (AWGN) rather than block fading and if the nodes were to flush their memory of past blocks whenever a new relay is selected, then the model will be similar to the one considered in [14]. The main difference is that while the GeRaF protocol in [14] did not use hybrid-ARQ, the HARBINGER protocol proposed in this paper does. While such an analysis goes beyond the scope of this paper, the interested reader is directed to our recent work that analyzes the performance of HARBINGER in AWGN with memory flushing [33], [34].

APPENDIX
STATISTICS OF HYBRID-ARQ BASED MULTIHOP

The purpose of this Appendix is to derive the expected number of transmissions $E[\mathcal{M}_d]$ of hybrid-ARQ over a block fading direct link with average SNR Γ. After the mth block has been transmitted, the outage probability of the link is

$$P_d[m] = \text{Prob}\left\{\sum_{i=1}^{m} I(\gamma_d[i]) \leq R\right\} \tag{11}$$

for code combining and

$$P_d[m] = \text{Prob}\left\{I\left(\sum_{i=1}^{m} \gamma_d[i]\right) \leq R\right\} \tag{12}$$

for diversity combining. The instantaneous SNR's $\{\gamma_d[i]\}$ are i.i.d. exponential random variables with mean $E\{\gamma_d[i]\} = \Gamma$. Let $J_d[m] = P_d[m-1] - P_d[m]$ denote the probability mass function (pmf) of $\mathcal{M}_d$, the number of transmitted blocks over the direct link. When the rate constraint $M \to \infty, \mathcal{J}_d(z) = \sum_m J_d[m] z^m$ corresponds to the characteristic function of $\mathcal{M}_d$.

The pmf for diversity combing can be found in closed form [35]

$$P_d[m] = 1 - \exp\left\{-\frac{2^{2R}-1}{\Gamma}\right\} \sum_{i=1}^{m-1} \frac{1}{i!} \left(\frac{2^{2R}-1}{\Gamma}\right)^i$$

with a corresponding characteristic function

$$\begin{aligned}\mathcal{J}_d(z) &= (P_d[m-1] - P_d[m]) z^m \\ &= z \cdot \exp\left\{(z-1)\frac{2^{2R}-1}{\Gamma}\right\}. \end{aligned} \tag{13}$$

The expected value under diversity combining can then be found by differentiating the characteristic function

$$\begin{aligned} E[\mathcal{M}_d] &= \sum_{m=1}^{\infty} m J_d[m] \\ &= \left.\frac{d\mathcal{J}_d(z)}{dz}\right|_{z=1} \\ &= \left(1 + \frac{2^{2R}-1}{\Gamma}\right). \end{aligned} \tag{14}$$

While a similar approach can be used to find $E[\mathcal{M}_d]$ for code combining, the resulting integration has no closed form expression (though it can be solved using Monte Carlo integration).

ACKNOWLEDGMENT

The authors would like to thank D. Goeckel, D. Reynolds, and S. Wei for their feedback on this work.

REFERENCES

[1] G. Lauer, "Packet-radio routing," in *Routing in Communications Networks*, M. Steenstrup, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1995, ch. 11, pp. 351–396.

[2] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Algorithms for energy-efficient multicasting in static ad hoc wireless networks," *Mobile Networks Appl. (MONET)*, vol. 6, pp. 251–263, Jun. 2001.

[3] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, pp. 388–404, Mar. 2000.

[4] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad-hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, pp. 477–486, Aug. 2002.

[5] P. Gupta and P. R. Kumar, "Toward an information theory of large networks: An achievable rate region," *IEEE Trans. Inf. Theory*, vol. 49, pp. 1877–1894, Aug. 2003.

[6] T. M. Cover and A. A. El Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inf. Theory*, vol. 25, pp. 572–584, Sep. 1979.

[7] B. Schein and R. Gallager, "The Gaussian parallel relay network," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Sorrento, Italy, Jun. 2000, p. 22.

[8] M. Gastpar and M. Vetterli, "On the capacity of wireless networks: The relay case," in *Proc. INFOCOM*, New York, pp. 1577–1586.

[9] A. Høst-Madsen, "On the capacity of wireless relaying," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Vancouver, BC, Canada, Sep. 2002.

[10] M. Khojastepour, A. Sabharwal, and B. Aazhang, "On the capacity of 'cheap' relay networks," in *Conf. Inf. Sci. Syst.*, Baltimore, MD, Apr. 2003.

[11] J. N. Laneman and G. W. Wornell, "Exploiting distributed spatial diversity in wireless networks," in *Proc. Allerton Conf. Commun., Control, Comput.*, Allerton, IL, Oct. 2000.
[12] ——, "Distributed space–time coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2415–2425, Oct. 2003.
[13] G. Kramer, M. Gastpar, and P. Gupta, "Capacity theorems for wireless relay channels," in *Proc. Allerton Conf. Commun., Control, Comput.*, Allerton, IL, Oct. 2003.
[14] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Trans. Mobile Comp.*, vol. 2, pp. 337–348, Oct. 2003.
[15] ——, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance," *IEEE Trans. Mobile Comp.*, vol. 2, pp. 349–365, Oct. 2003.
[16] S. Wicker, *Error Control Systems for Digital Communications and Storage*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[17] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, Cooperative diversity in wireless networks: Efficient protocols and outage behavior, in *IEEE Trans. Inform. Theory*, 2004, to be published.
[18] G. Caire and D. Tuninetti, "The throughput of hybrid-ARQ protocols for the Gaussian collision channel," *IEEE Trans. Inf. Theory*, vol. 47, pp. 1971–1988, Jul. 2001.
[19] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—Part I: System description," *IEEE Trans. Commun.*, vol. 51, pp. 1927–1938, Nov. 2003.
[20] ——, "User cooperation diversity—Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, pp. 1939–1948, Nov. 2003.
[21] T. E. Hunter and A. Nosratinia, "Performance analysis of coded cooperation diversity," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Anchorage, AK, May 2003.
[22] A. Stefanov and E. Erkip, "Cooperative coding for wireless networks," in *Proc. IEEE Conf. Mobile Wireless Commun. Netw.*, Stockholm, Sweden, Sep. 2002.
[23] E. M. Royer and C. K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Pers. Commun. Mag.*, vol. 6, pp. 46–55, Apr. 1999.
[24] B. Zhao and M. C. Valenti, "Some new adaptive protocols for the wireless relay channel," in *Proc. Allerton Conf. Commun., Control, Comput.*, Allerton, IL, Oct. 2003.
[25] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
[26] L. Ozarow, S. Shamai, and A. D. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 359–378, May 1994.
[27] R. Knopp and P. A. Humblet, "On coding for block fading channels," *IEEE Trans. Inf. Theory*, vol. 46, pp. 189–205, Jan. 2000.
[28] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *IEEE Trans. Inf. Theory*, vol. 44, pp. 2619–2692, Oct. 1998.
[29] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, vol. 6, pp. 311–335, Mar. 1998.
[30] M. C. Valenti and B. Zhao, "Distributed turbo codes: Toward the capacity of the relay channel," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Orlando, FL, Oct. 2003.
[31] R. Wolff, *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ: Prentice-Hall, 1999.
[32] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Resource management in energy-limited, bandwidth-limited, transceiver-limited wireless networks for session-based multicasting," *Comput. Netw.*, vol. 39, pp. 113–131, Jun. 2002.
[33] M. C. Valenti and B. Zhao, "Hybrid-ARQ based intra-cluster geographic relaying," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Monterey, CA, Oct. 2004.
[34] B. Zhao, R. Iyer Seshadri, and M. C. Valenti, "Geographic random forwarding with hybrid-ARQ for ad hoc networks with rapid sleep cycles," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dallas, TX, Dec. 2004.
[35] J. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

Bin Zhao (S'00) received the B.S.E.E. and M.S.E.E. degrees from Shanghai Jiaotong University, Shanghai, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical engineering from West Virginia University, Morgantown, in 2004, where he worked as a Research Assistant in the Wireless Communication Research Laboratory.

He is currently a Communications Engineer with Efficient Channel Coding, Inc., Brooklyn Heights, OH. Prior to attending graduate school at West Virginia University, he was a DSP Engineer with Huawei Technologies Company Ltd., Shenzhen, China, where he was engaged in the development of a real-time speech and channel codec for the IS-95 system. His research interests are in the areas of communication theory, error correction coding, sensor networks, and information theory.

Matthew C. Valenti (M'99) received the B.S.E.E. degree from Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, in 1992, the M.S.E.E. degree from The Johns Hopkins University, Baltimore, MD, in 1995, and the Ph.D. degree in electrical engineering from Virginia Tech in 1999, where he was a Bradley Fellow.

He is currently an Assistant Professor in the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown. Prior to attending graduate school at Virginia Tech and State University, he was an Electronics Engineer at the United States Naval Research Laboratory, Washington, DC, where he was engaged in the design and development of a space–bourne adaptive antenna array and a system for the collection and correlation of maritime ELINT signals. He is a consultant to several companies engaged in various aspects of turbo codec design, including software radio, FPGA, and ASIC implementations for military, satellite, and third-generation cellular applications. His research interests are in the areas of communication theory, error correction coding, applied information theory, and wireless multiple-access networks.

Dr. Valenti serves as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and has been on the technical program committee for several international conferences.