EURASIP Journal on Wireless Communications and Networking 2005:5, 610–624
© 2005 B. Zhao and M. C. Valenti

Position-Based Relaying with Hybrid-ARQ for Efficient Ad Hoc Networking

Bin Zhao
Lane Department of Computer Science and Electrical Engineering, College of Engineering and Mineral Resources, West Virginia University, Morgantown, WV 26506-6109, USA
Email: bzhao@csee.wvu.edu

Matthew C. Valenti
Lane Department of Computer Science and Electrical Engineering, College of Engineering and Mineral Resources, West Virginia University, Morgantown, WV 26506-6109, USA
Email: mvalenti@csee.wvu.edu

Received 15 June 2004; Revised 3 January 2005

This paper presents and analyzes an integrated, cross-layer protocol for wireless ad hoc networking that utilizes position location (e.g., through an onboard GPS receiver) and jointly performs the operations of network-layer relaying and link-layer ARQ-based error control. The protocol is a modified version of the hybrid-ARQ-based intra-cluster geographically-informed relaying (HARBINGER) protocol (2005) and unifies the concepts of geographic random forwarding (GeRaF) (2003), point-to-point hybrid-ARQ (2001), and cooperative diversity (2004). The modification makes the protocol especially suitable for sensor networks whose nodes cycle in and out of sleep states and permits a closed-form analysis. Performance bounds and simulations indicate the potential for a dramatic improvement in the tradeoff between active node density and end-to-end message delay as compared with the GeRaF protocol and are used to motivate further study of practical implementation issues.

Keywords and phrases: relay networks, ad hoc networking, cross-layer protocols, hybrid-ARQ, GeRaF, HARBINGER.

1. INTRODUCTION

Wireless ad hoc networks in general, and sensor networks in particular, must be energy efficient and able to deliver messages with low latency. One way to improve the energy-latency tradeoff is to exploit the inherent spatial diversity that arises when multiple relay nodes are within transmission range of each source node [1, 2]. A properly designed cross-layer protocol could enable multiple single-antenna devices located in close proximity of one another to operate as a virtual antenna array by implementing a strategy known as *cooperative diversity* [3]. Another way to conserve energy is to periodically put each radio into a sleep mode, since listening to idle channels consumes significant processing and transceiver power [4]. The lifetime of the network is primarily a function of the duty cycle of the nodes, and networks whose nodes are in a sleep state for a higher percentage of time will last longer. However, these two strategies conflict with one another. A network with an aggressive sleep cycle might not have a high enough active node density for cooperative diversity to be effective. In this paper, we will describe and analyze efficient cross-layer protocols that simultaneously allow a wireless network to exploit distributed diversity while maintaining an aggressive sleep schedule.

The protocols discussed in this paper are based upon the HARBINGER[1] protocol that we first introduced in [1]. HARBINGER is a generalization of the concept of hybrid-ARQ [5]. With hybrid-ARQ, messages are encoded using a low-rate mother code and broken into several frames of *incremental redundancy* (IR). The transmitter will send IR frames one at a time until the receiver is able to decode the message and responds with a positive acknowledgment. With traditional point-to-point hybrid-ARQ, all IR frames are sent by the source node. However, in dense wireless networks, nodes near the source and/or destination may *overhear* the transmitted frames. A *cluster* can be formed by pooling the source, destination, and several nearby *relay* nodes. If any

[1]Hybrid-ARQ-based intra-cluster geographically-informed relaying.

of the relay nodes are able to successfully decode the message, then they can transmit the next IR frame. This adds a dimension of transmit spatial diversity, since all transmitted frames do not come from the same location. HARBINGER is a true cross-layer protocol because it combines elements of link-layer error control (through transmission of incremental redundancy) and network-layer routing (through relay selection).

Though simple in concept, implementing HARBINGER poses several challenges. The most crucial issue is that several relays in the cluster could overhear the transmission and a contention scheme is required to determine which relays transmit and when they do so. The solution suggested in [1], and also adopted in this paper, is to use geographic information to guide the relay schedule. It is assumed that each node knows its own location (by using an onboard GPS receiver or a localization algorithm) and that messages are addressed by the physical location of the destination. When a message is successfully decoded by multiple relays, then the relay that is closest to the destination will be the one that transmits the next IR frame, thus maximizing the forward progress of the message. Implementation details of this contention scheme are discussed later in this paper.

Another issue with the basic version of HARBINGER is that it does not lend itself to networks with aggressive sleep schedules and requires each node to buffer a fairly large number of received IR frames. This is because all nodes in the cluster must remain awake and available to transmit the next IR frame until the message is successfully decoded at the destination. Furthermore, each node must keep copies of every IR frame it receives from every node in the cluster until the message is finally decoded by the destination. Because each node buffers all of the frames it receives and these frames are sent from multiple transmitters, the memory in the system precludes an efficient closed-form analysis, and thus performance must be assessed through simulation (all numerical results in [1] were found through simulation).

The twist on HARBINGER considered in this paper is to allow all nodes to flush their memory of previously transmitted IR frames every time a new relay is selected to forward the message, that is, every time there is *forward progress*. Though seemingly a minor modification, this has a profound impact on the system. First, it reduces the required buffer size at each relay and second, it allows nodes to go back to sleep once a new relay is selected. Just as some nodes in the cluster go back to sleep, others may wake up, thereby making the cluster composition time-varying, adding an additional element of time-diversity. Finally, and perhaps most importantly for this paper, by constraining the nodes to flush their memory each time the message hops to the next relay, a closed-form analysis is possible.

Even though nodes flush their memory after each forward hop, hybrid-ARQ is still an important feature of the protocol. To see this, consider a situation where the propagation environment is isotropic and the channel is unfaded (thereby producing concentric circles of equal signal-to-noise ratio). The low-rate mother code is broken into M equal-sized frames of incremental redundancy. When the first IR frame is sent, all nodes within some range R_1 of the source will be able to successfully decode the message, where R_1 depends on the minimum SNR required to decode the first IR frame. If there is no node within range R_1, then the source can send the next IR frame. The implication of sending the second frame is that the code rate has effectively been lowered, and therefore the reachable range will have increased; therefore, any node within range $R_2 > R_1$ will be able to decode the second frame (provided that it was awake when the first frame was transmitted). This process continues until, finally, the Mth frame is sent and any node within range R_M is able to decode the frame.

Under the memory-flushing constraint considered in this paper, HARBINGER is related to an independently developed protocol known as geographic random forwarding (GeRaF) [6, 7]. Like our protocol, GeRaF is a cross-layer protocol that uses position location to guide the selection of a relay. However, GeRaF does not use hybrid-ARQ, and is therefore only able to reach nodes within range R_1. In fact, GeRaF is a special case of HARBINGER, and in particular corresponds to the case that $M = 1$. The benefit of using hybrid-ARQ ($M > 1$) is that the coverage area effectively increases after each transmission. As illustrated in the numerical results, the coverage expansion effect allows the network to operate with a lower density of active nodes, thereby allowing the system to operate with a more aggressive sleep schedule than if it used GeRaF.

The rest of this paper is organized as follows. In Section 2, the basic HARBINGER protocol is briefly reviewed and modifications related to memory flushing are discussed. Two new versions of HARBINGER, termed *fast HARBINGER* [8] and *slow HARBINGER* [9] are presented. Section 3 presents an analysis of these two versions of HARBINGER through a nontrivial generalization of GeRaF. Section 4 provides numerical results and studies the impact of parameters such as active node density, path loss exponent, and M (the maximum number of IR frames). Simulation results are provided to validate the analysis. Finally, Section 5 draws conclusions and suggests paths for future research.

2. MODIFIED HARBINGER

Consider a *network* $\mathcal{N} = \{Z_k : 1 \leq k \leq K\}$ consisting of a *source* Z_s, a *destination* $Z_d = Z_K$, and $K - 2$ *relays*. Each node has a single half-duplex radio and a single antenna. The propagation environment is isotropic and impaired only by exponential path loss and additive white Gaussian noise (AWGN). While the channel is likely to be affected also by interference and fading, such issues were already discussed in [1], are outside the scope of the present paper, and will only obscure the analysis that we present here. Nodes are numbered according to their distance to the destination, with Z_1 being the furthest and Z_{K-1} being the closest. Initially, the source is node $Z_s = Z_1$, but the identity of the source node changes as the message propagates through the network. Time is divided into *slots* s, which are of equal duration. Nodes cycle on and off according to a pseudorandom sleep schedule, and we denote the *cluster* $\mathcal{C}(s) \subset \mathcal{N}$ to be the set of *geographically advantaged*[2] active nodes within range R_M of the source during the sth slot. The average *density* of active nodes per unit area is denoted by ρ. For analytical purposes, it is assumed that the nodes are distributed according to a two-dimensional Poisson process, though the protocol itself will work for any arbitrary node distribution.

The source begins by encoding a b_d bit message into a codeword of length n symbols. The codeword is broken into M *frames*, each of length $L = n/M$ and rate $r = b_d/L$. The code itself could simply be a *repetition* code, in which case all M frames are identical and each node in the cluster will *diversity combine* [10] all frames that it has received. More generally, *incremental redundancy* [10] could be used, whereby each frame is obtained by puncturing a rate r/M mother code. With incremental redundancy, a different part of the codeword is transmitted each time, and after the mth frame, a receiver will pass the rate r/m code that it has until then received through its decoder (*code combining*). As in [5], M is called the *rate constraint*.

During each slot, the source transmits the next ARQ frame in the sequence, while all other nodes in the cluster listen for the frame. The frames $1 \leq m \leq M$ are transmitted during consecutive slots $\{s_1, \dots, s_M\}$. Each frame has a header that contains the ARQ frame sequence number $m : 1 \leq m \leq M$, location of the source, and location of the destination. The header is encoded separately by a rate r/M code so that all nodes in the cluster can decode every frame's header. To improve efficiency, an RTS-CTS dialogue could be used. An RTS packet could be sent prior to the ARQ frame. The RTS would contain the same information in the frame header and would also be encoded by a rate r/M code. If the current network configuration and interference conditions will not allow the message to make any forward progress, the source could wait until more favorable conditions prevail. Details of the dialogue go beyond the scope of this paper, but are a straightforward modification of the handshaking procedure discussed in [6, 7].

The source continues to transmit ARQ frames until either all M frames have been transmitted, the destination decodes the message, or a relay is able to decode the message and is elected to forward the message. In the case that neither relay nor destination was able to decode the message, the process starts over with the source once again transmitting up to M frames. On the other hand, if relay Z_r is able to decode the message and is elected to forward the message, then it assumes the role of the source, and the process starts over with the new source $Z_s = Z_r$ transmitting up to M frames. Finally, if the destination is able to decode the message, the process halts and the message is delivered to the application.

Nodes periodically make an independent decision to wake up, go to sleep, or remain in the same state. Nodes may change sleep states at one of two instances, depending on the version of the HARBINGER protocol. In *fast* HARBINGER, nodes may change state at the end of each slot, and so the network topology is fixed for only one slot at a time. In *slow* HARBINGER, nodes may only change state once every M slots. The M slots are arranged into a *superslot* that is long enough for all M ARQ frames to be transmitted. The hybrid-ARQ protocol is synchronized with the superslots so that the first ARQ frame must be sent during the first slot of the superslot, and so on. This guarantees that the topology will remain fixed for all M ARQ frames, but also means that the network must wait until the start of the next superslot before the message can be forwarded from the new source.

Each frame is transmitted by the source node Z_s with average energy per symbol $\mathcal{E}_s$, which is assumed to be constant for all frames. For the sake of mathematical tractability, we follow [5] and assume that circularly symmetric complex Gaussian symbols are transmitted. The frame is received at node $Z_k \in \{\mathcal{C}(s) \setminus Z_s\}$ with average energy per symbol $\mathcal{E}_k = K_o d_k^{-\mu} \mathcal{E}_s$, where d_k is the distance from Z_s to Z_k, μ is a path loss exponent, and K_o is a constant that depends on the wavelength λ_c and free-space reference distance d_o [11].

The signal is received at Z_k over an additive white Gaussian noise (AWGN) channel with signal-to-noise ratio (SNR) $\mathcal{E}_k/N_o$, where N_o is the one-sided noise spectral density. If only one frame was sent, the channel would have a capacity of $C = (1/2)\log_2(1+\mathcal{E}_k/N_o)$. However, due to the use of hybrid-ARQ, node Z_k could have received more than just one frame. Consider the case when node Z_k has received m frames. For a diversity combining system, the SNR adds [5], and thus the capacity becomes $C_k(m) = (1/2)\log_2(1 + m\mathcal{E}_k/N_o)$, while for code combining, the capacities add [5], and thus $C_k(m) = (m/2)\log_2(1 + \mathcal{E}_k/N_o)$.

Any node Z_k whose capacity after the mth transmission is greater than the rate r will have accumulated enough information to decode the message. Define the *decoding set* $\mathcal{D}(s_m) \subset \mathcal{C}(s_m)$ to be the set of all nodes that have decoded the message after the mth frame has been transmitted, that is, $\mathcal{D}(s_m) = \{Z_k : C_k(m) > r\}$. As soon as the destination is admitted to the decoding set, the message is delivered to the application. Once a relay is added to the decoding set, it could potentially become the new source and forward the message. The two modifications of HARBINGER differ in how the forwarding relay is selected from the decoding set, as discussed in the next two sections.

2.1. Slow HARBINGER

In slow HARBINGER, the composition of the cluster $\mathcal{C}(s)$ remains fixed for all $s : s_1 \leq s \leq s_M$, that is, for an entire superframe. After the mth frame has been transmitted, all nodes within some distance d_m will be able to decode the message and will be added to the decoding set. The distance d_m is found from the capacity expression and exponential path loss model to be

$$d_m = \left(\frac{K_0 \mathcal{E}_s / N_o}{2^{2r/m} - 1} \right)^{1/\mu} \tag{1}$$

[2]Geographically advantaged nodes are closer to the destination than the source is to the destination [6].

for code combining, and

$$d_m = \left(\frac{mK_0\mathcal{E}_s/N_o}{2^{2r}-1}\right)^{1/\mu} \tag{2}$$

for diversity combining. To remove dependency on the parameters K_0, $\mathcal{E}_s/N_o$, and the actual physical distances, we normalize the transmission distance so that the range that can be reached after the first ARQ frame is transmitted is unity. We denote normalized distance as R_m, so that $R_1 = 1$ and

$$R_m = \begin{cases} \left(\dfrac{2^{2r}-1}{2^{2r/m}-1}\right)^{1/\mu} & \text{for code combining,} \\ m^{1/\mu} & \text{for diversity combining.} \end{cases} \tag{3}$$

Thus, under slow HARBINGER, $\mathcal{D}(s_m) = \{Z_k : Z_k \in \mathcal{C}(s_m),\ d_k < R_m\}$. We define the mth *coverage band* B_m to be the geographically advantaged area that is between distance R_{m-1} and R_m from the source. Band B_0 is defined to contain only the source. We further define $B_{m'}$ to be the band that contains the node in the cluster that is closest to the destination. If two or more nodes are at the same minimum distance to the destination but in different bands, then $B_{m'}$ will be the band which is closer to the source (has smallest subscript). If $m' = 0$, then the cluster contains only the source, and therefore it should not transmit any ARQ frames during the current superslot (the source can determine if there are any other nodes in the cluster by sending out an RTS packet).

Since the sleep states are synchronized to only change once every M slots, the network must wait until the start of the next superslot before the message can be transmitted from a new source, that is, the message may only make forward progress once every M slots. Because of this, there are two very different strategies for picking which node in the cluster will forward the message. The first strategy, termed *slow HARBINGER A*, minimizes the source-destination latency, while the second strategy, termed *slow HARBINGER B*, minimizes the energy consumption.

Minimizing the latency is equivalent to maximizing the forward progress of the message. This is accomplished in slow HARBINGER A by selecting the forwarding node after frame m' is sent to be the relay that is closest to the destination, that is, the $Z_k \in \mathcal{C}(s_{m'})$ with the largest index k (since nodes are indexed according to distance to the destination). Note that it is possible for more than one relay to be added to the decoding set during the final hybrid-ARQ transmission m'. This occurs if there are more than one relay in band $B_{m'}$. In this case, a contention mechanism is needed to pick the relay that is closest to the destination. The contention scheme from [6] could be adopted, which slices the cluster into several *priority regions* based on the distance to the destination. Nodes that are in the priority region closest to the destination are given the opportunity to contend for the channel first. If no nodes are found, then the second closest priority zone has the opportunity to contend, and so on. If multiple nodes are present in the same priority zone, a random backoff procedure can be used to further resolve the contention. Once a forwarding relay is selected, all nodes in the cluster may go back to sleep, with the forwarding relay waking up again at the start of the next superslot.

Due to the exponential path loss effect, minimizing energy consumption is equivalent to minimizing the number of ARQ transmissions required for the message to make forward progress in each superslot. This is accomplished in slow HARBINGER B by selecting the forwarding node from among the first relays added to the decoding set. Once any relay is added to the decoding set, it will signal an acknowledgment and the source will stop transmitting frames. If multiple relays are added to the decoding set at the same time, then the same contention scheme used for slow HARBINGER A can be used to select the relay that is closest to the destination (the contention scheme will also prevent acknowledgments from colliding).

2.2. *Fast HARBINGER*

With fast HARBINGER, the composition of the cluster $\mathcal{C}(s)$ may change after each slot. If a node Z_k is located in coverage band B_j, then it will be able to decode the message after the mth ARQ frame is transmitted if it was awake for the last j out of the m ARQ transmissions. Once a node wakes up and receives the next ARQ frame, it must make a local decision to stay awake or go back to sleep. The node will compare the ARQ sequence number against its own location, and will go back to sleep if it will be unable to decode the message after the last (Mth) ARQ frame is transmitted. A node located in B_j will go back to sleep if it wakes up *after* slot $m = M - j$. Otherwise, it will stay awake for the remaining ARQ transmissions until either it decodes the message or another node in the cluster decodes the message and sends an acknowledgment. Once a node is admitted to the decoding set, the source stops transmitting and the node in the decoding set that is closest to the destination begins to forward the message during the next slot. If more than one node are added to the decoding set after the same frame, the same contention scheme used by slow HARBINGER can be used to select the node that is closest to the destination.

3. RECURSIVE ANALYSIS

The analysis of modified HARBINGER is a nontrivial generalization of the analysis of GeRaF introduced in [6]. The analysis gives recursive upper and lower bounds on the average end-to-end latency (in number of slots) and the average number of ARQ transmissions for the message to be delivered to the destination. The first metric is of interest because it quantifies the network delay, while the second metric is related to energy consumption (since each ARQ frame is transmitted with equal energy). Because of the complexity of the analysis, we only present the main results in this section. Full details of the analysis can be found in the appendix.

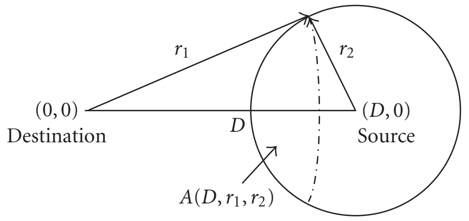

FIGURE 1: The area of intersection of two circles of radii r_1 and r_2 separated by a distance of D.

As with GeRaF, we assume that the active nodes are distributed according to a two-dimensional Poisson process. This is an accurate model when the density of actual nodes is high and each node uses an exponential sleep timer [6]. The analysis relies on certain features of Poisson processes which implies that (1) the number ($|\mathcal{C}(s)|$) of active nodes in a cluster $\mathcal{C}(s)$ is a Poisson random variable; and (2) if the node distribution of entire network is two-dimensional Poisson with density ρ, then any region within the network will have a Poisson node distribution with density ρ.

The source and destination are separated by D units, where a *unit* is the range of the first ARQ transmission $R_1 = 1$. To enable recursive calculation, space is divided into ν *increments* per unit distance. Each increment has length $1/\nu$. The upper and lower bounds coincide as $\nu \to \infty$. We define the *message transfer probability* $\omega(j,k,b,m)$ to be a joint probability, where j is the number of increments separating the source and destination, k is the forward progress (in increments) of the message during the current hop, b is the number of slots that have elapsed for the current hop, and m is the number of received ARQ frames during the current hop. We define the *empty hop probability* $\omega_0(j)$ to be the probability that no forward progress has been made in the current hop when the source is j increments from the destination. In the following analysis, we assume that $j > \nu R_M$, that is, that direct communications is not possible between source and destination.

Let $A(D,r_1,r_2)$ denote the area of intersection of two circles with radii r_1 and r_2 separated by a center-to-center distance of D. This area is indicated in Figure 1 and is computed using

$$A(D,r_1,r_2) = 2\int_{D-r_2}^{r_1} \arccos\left(\frac{x^2+D^2-r_2^2}{2Dx}\right) x\,dx. \quad (4)$$

3.1. Slow HARBINGER

As derived in the appendix, the lower bound on average message delay when the source and destination are separated by $j \le \nu D$ increments is

$$n(j) = \sum_{k=1}^{\nu R_M}\sum_{m=1}^{M} \omega(j,k,M,m)\big(n(j-k)+M\big) + \omega_0(j)\big(n(j)+M\big), \quad (5)$$

while the lower bound on the average number of ARQ transmissions is

$$e(j) = \sum_{k=1}^{\nu R_M}\sum_{m=1}^{M} \omega(j,k,M,m)\big(e(j-k)+m\big) + \omega_0(j)e(j). \quad (6)$$

The corresponding upper bound is found by replacing the $(j-k)$ terms in (5) and (6) with $(j-k+1)$.

The empty hop probability for slow HARBINGER (both types) is given by

$$\omega_0(j) = \exp\left\{-\rho A\left(\frac{j}{\nu},\frac{j}{\nu},R_M\right)\right\}. \quad (7)$$

The message transfer probability depends on the type of protocol. For slow HARBINGER A, it is

$$\begin{aligned}\omega(j,k,M,m) &= \exp\left\{-\rho A\left(\frac{j}{\nu},\frac{j-k}{\nu},R_M\right)\right\}\\ &\cdot\Bigg[\exp\left\{\rho\left(A\left(\frac{j}{\nu},\frac{j-k}{\nu},R_{m-1}\right) - A\left(\frac{j}{\nu},\frac{j-k+1}{\nu},R_{m-1}\right)\right)\right\}\\ &- \exp\left\{\rho\left(A\left(\frac{j}{\nu},\frac{j-k}{\nu},R_m\right) - A\left(\frac{j}{\nu},\frac{j-k+1}{\nu},R_m\right)\right)\right\}\Bigg],\end{aligned} \quad (8)$$

while for slow HARBINGER B, it is

$$\begin{aligned}\omega(j,k,M,m) &= \exp\left\{-\rho A\left(\frac{j}{\nu},\frac{j}{\nu},R_{m-1}\right)\right\}\\ &\cdot\Bigg[\exp\left\{-\rho\left(A\left(\frac{j}{\nu},\frac{j-k}{\nu},R_m\right) - A\left(\frac{j}{\nu},\frac{j-k}{\nu},R_{m-1}\right)\right)\right\}\\ &- \exp\left\{-\rho\left(A\left(\frac{j}{\nu},\frac{j-k+1}{\nu},R_m\right) - A\left(\frac{j}{\nu},\frac{j-k+1}{\nu},R_{m-1}\right)\right)\right\}\Bigg].\end{aligned} \quad (9)$$

The end-to-end delay is computed recursively. For slow HARBINGER A, the recursion starts from a distance separation of $\nu R_M + 1$ increments, that is, the index j in (5) and (6) is initially set to $j' = \nu R_M + 1$. For slow HARBINGER B, the recursion starts at $j' = \nu R_1 + 1$. The initial conditions for the recursion are $n(j) = M$ for $j \le \nu R_M$ and $e(j) = m$ for $\nu R_{m-1}+1 \le j \le \nu R_m$, where $1 \le m \le M$. During the first step of the recursion, the message delay $n(j')$ and number of ARQ transmissions $e(j')$ are computed for the initial condition j'. These results are then used to compute the message delay at increment $j = j' + 1$. The process continues recursively, until the message delay and number of ARQ transmissions at increment $j = \nu D$ are computed.

3.2. Fast HARBINGER

Because the composition of the cluster $\mathcal{C}(s)$ changes after each slot in fast HARBINGER, its statistics are different than slow HARBINGER's. In particular, the lower bound on average message delay when the source and destination are separated by $j \le \nu D$ increments is

$$n(j) = \sum_{k=1}^{\nu R_M} \sum_{b=1}^{M} \sum_{\ell=1}^{b} \omega(j,k,b,\ell)\big(n(j-k)+b\big) + \omega_0(j)\big(n(j)+M\big), \tag{10}$$

and the lower bound on the average number of ARQ transmission is

$$e(j) = \sum_{k=1}^{\nu R_M} \sum_{b=1}^{M} \sum_{\ell=1}^{b} \omega(j,k,b,\ell)\big(e(j-k)+\ell\big) + \omega_0(j)e(j). \tag{11}$$

The corresponding upper bound is found by replacing the $(j-k)$ terms in (10) and (11) with $(j-k+1)$.

For fast HARBINGER, the empty hop probability is

$$\omega_0(j) = \prod_{i=1}^{M} \exp\left\{ -\rho A\left(\frac{j}{\nu}, \frac{j}{\nu}, R_i\right)\right\}, \tag{12}$$

while the message transfer probability is

$$\omega(j,k,b,m) = \begin{cases} \Omega(j,k,b,m) - \Omega(j,k,b,m-1) & \text{for } m \le b, \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

where

$$\begin{aligned} \Omega(j,k,b,m) &= \left(\exp\left\{ -\rho A\left(\frac{j}{\nu}, \frac{j}{\nu}, R_M\right)\right\}\right)^{b-m} \\ &\quad \times \left(\prod_{i=1}^{m-1} \exp\left\{ -\rho A\left(\frac{j}{\nu}, \frac{j}{\nu}, R_i\right)\right\}\right) \\ &\quad \cdot \left[\exp\left\{ -\rho A\left(\frac{j}{\nu}, \frac{j-k}{\nu}, R_m\right)\right\} \right. \\ &\qquad \left. - \exp\left\{ -\rho A\left(\frac{j}{\nu}, \frac{j-k+1}{\nu}, R_m\right)\right\}\right]. \end{aligned} \tag{14}$$

The end-to-end delay and number of ARQ transmissions are computed recursively just as in slow HARBINGER. The initial conditions are identical to that of slow HARBINGER B, and in particular $j' = \nu R_1 + 1$, $n(j) = 1$ for $j \le \nu R_1$, and $e(j) = 1$ for $j \le \nu R_1$.

4. NUMERICAL RESULTS

In this section, both analytical and simulation results are presented to illustrate the behavior of modified HARBINGER and demonstrate its advantage over GeRaF. The simulation setup is discussed in Section 4.1. Since code combining allows the coverage circles R_m to expand at a faster rate than with diversity combining, we begin by presenting numerical results for code combining. The average latency and number of ARQ transmissions are presented for code combining in Sections 4.2 and 4.3, respectively. A comparison of code combining and diversity combining is then given in Section 4.4. Finally, the impact of the path loss exponent μ is assessed in Section 4.5.

4.1. Simulation setup

To validate the analysis, a set of computer simulations was executed. In each simulation trial, the source and relay are first located a fixed distance D apart. The relay topology is then periodically created at random according to a two-dimensional Poisson process with density ρ. Note that the number of nodes located within a coverage area of size A is in itself a Poisson random variable with mean $\varepsilon = \rho A$. The homogeneous Poisson distribution is generated following the methodology of [12] by first generating a Poisson random variable with mean $\varepsilon = \rho A$ to determine the number $\mathcal{P}$ of nodes within the area of interest, and then independently placing each node within the area according to a uniform distribution.

The rate that the network topology changes depends on the version of HARBINGER. For slow HARBINGER, the cluster composition remains fixed for an entire superslot at a time. Thus, the simulation must draw from the Poisson process only once per superslot. If the cluster contains more than just the source, then the message will make forward progress, otherwise a new distribution is drawn. In either case, a message delay counter is incremented by M slots. For slow HARBINGER A, the message progresses to either the relay in the cluster that is closest to the destination or to the destination itself (if it is in the cluster). For slow HARBINGER B, the message progresses to either the most geographically advantaged node that is first added to the decoding set or to the destination itself if it is added to the decoding set first. Each time the message makes forward progress, an ARQ frame counter is incremented by amount κ if the node that the message progresses to is in band B_κ. Once the message reaches the destination, the simulation halts and a new trial is run. For each set of simulation parameters, 5000 trials are run.

For fast HARBINGER, it is necessary to update the cluster configuration prior to each slot. Note that the sequence of

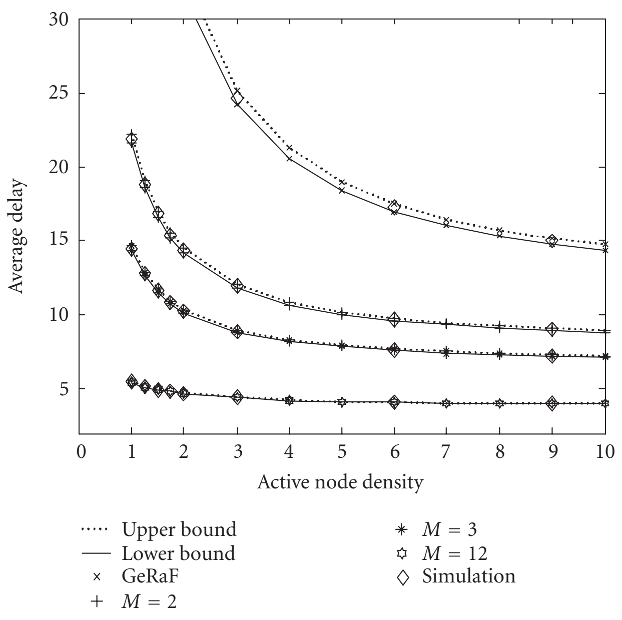

FIGURE 2: Upper and lower bounds on message delay (in units of superslots) for slow HARBINGER A under different rate constraints M, where the perframe code rate $r = 1$, path loss exponent $\mu = 3$, $\nu = 50$ increments per unit distance, source-destination distance $D = 10$, and code combining hybrid-ARQ is used. GeRaF corresponds to the case that $M = 1$.

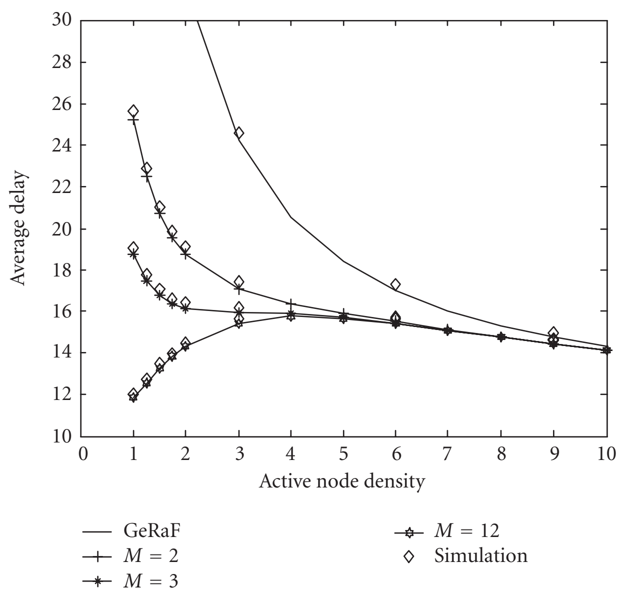

FIGURE 3: Lower bounds on message delay (in units of superslots) for slow HARBINGER B under the same conditions used in Figure 2.

cluster configurations is actually a *correlated* Poisson process because nodes located in band B_j that wake up prior to slot s_m will stay awake during slot s_{m+1} if $m \leq M - j$. Prior to slot s_1, a two-dimensional Poisson process is generated for each coverage band $\{B_1, \ldots, B_M\}$. What happens next depends on whether these coverage bands are empty or not. If all M coverage bands are empty, then prior to slot s_2, a new two-dimensional Poisson process is created for coverage bands $\{B_1, \ldots, B_{M-1}\}$. Note that nodes do not need to be placed in band B_M because they will wake up too late to decode the message. On the other hand, if some band B_κ is nonempty after slot s_1, then prior to slot s_2, a new two-dimensional process is created for each coverage band $\{B_1, \ldots, B_{\kappa-1}\}$. In this case, new nodes do not need to be placed in band B_κ or higher because the node already in band B_κ will be able to decode the message earlier. This entire process continues recursively until either the Mth ARQ frame is transmitted or the message makes forward progress. If the message did not make any forward progress, then an ARQ frame counter and a delay counter will be incremented by M, and the process will start over again from the same source node. On the other hand, if the message does make forward progress, then the two counters will be incremented by the message delay b and the actual number of transmitted ARQ frames m, respectively. If the message progresses to a relay, then the process will start over at the relay (which becomes the new source). Otherwise, if the message progresses to the destination, then the trial will halt and the simulation will move on to the next trial.

4.2. Message delay

Bounds on message delay for both slow HARBINGER and fast HARBINGER are plotted in Figures 2, 3, and 4 for perframe code rate $r = 1$, path loss exponent $\mu = 3$, $\nu = 50$ increments per unit distance, source-destination distance $D = 10$, and several values of the rate constraint M. The figures show the average end-to-end delay versus the node density ρ, where delay is in units of superslots for slow HARBINGER and in units of slots for fast HARBINGER and the node density is in units of nodes per unit area. In each Figure, the performance of GeRaF ($M = 1$) is included for reference. Also the corresponding simulation results are shown. Figure 2 shows both upper and lower bounds for slow HARBINGER A. Note that the two bounds are close to one another and that the simulation result lies between these two bounds. The tightness of the bounds is a function of the number of increments ν per unit distance, and as $\nu \to \infty$, the bounds get tighter. Due to the tightness of both bounds, we will only show *lower* performance bounds for the rest of this paper.

In Figure 2, we observe that the message delay in slow HARBINGER A decreases significantly with increasing M for all node densities. This result is rather intuitive, since from the message delay perspective, slow HARBINGER A is essentially GeRaF with its coverage radius expanded to R_M. Asymptotically, as the active node density $\rho \to \infty$, the message delay will converge to $\lfloor D/R_M + 1 \rfloor$. Unlike slow HARBINGER A, both slow HARBINGER B and fast HARBINGER have a similar delay performance as that of GeRaF in a relatively dense network, as shown in Figures 3

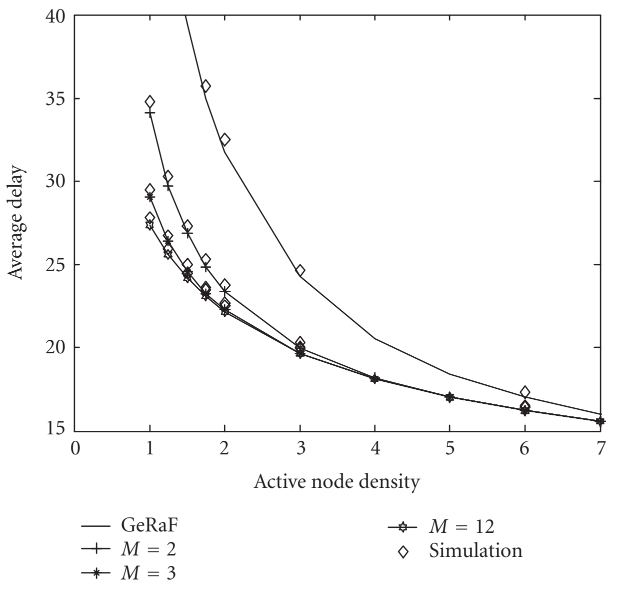

FIGURE 4: Lower bounds on message delay (in units of slots) for fast HARBINGER under the same conditions used in Figure 2.

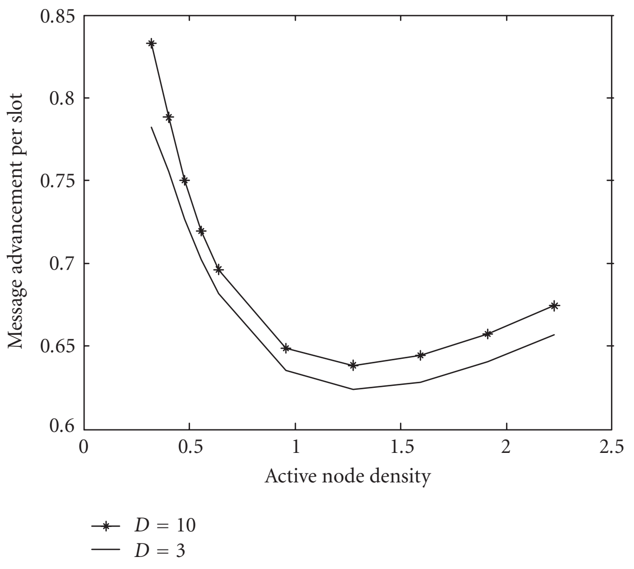

FIGURE 5: The average message advancement per slot for slow HARBINGER B with rate constraint $M = 12$ for source-destination separation $D = 3$ and 10, perframe code rate $r = 1$, path loss exponent $\mu = 3$, $\nu = 50$ increments per unit distance, and code combining.

and 4. In fact, they all asymptotically converge to a message delay of $\lfloor D + 1 \rfloor$ as node density $\rho \rightarrow \infty$. The major benefit of HARBINGER is in sparse networks, that is, where $\rho \rightarrow 0$. From these figures, it is apparent that the same average delay can be achieved with a lower node density by using HARBINGER instead of using GeRaF. For instance, consider fast HARBINGER with a delay of 25 slots. Using GeRaF, the density needs to be around $\rho = 3$ to achieve this delay. But by using fast HARBINGER with just $M = 2$, the required density is reduced to $\rho = 2$. By increasing M to 12, the required density is around $\rho = 1.5$ or about half what is needed for GeRaF, implying that the nodes may be asleep twice as often. It is interesting to note that the performance for $M = 3$ is nearly identical to that of $M = 12$ suggesting that diminishing returns kick in quickly and high values of M might not be needed in practice.

For both slow HARBINGER A and fast HARBINGER, the delay is a monotonically decreasing function of node density. However, an interesting phenomenon we observed for slow HARBINGER B in Figure 3 is that as the rate constraint gets fairly large, that is, $M = 12$, the delay is not a monotonic function of density. In particular, in low-density networks and for $M = 12$, the message delay actually decreases along with the node density. This observation is counterintuitive, but can be explained. Recall that with slow HARBINGER B, the forwarding node is selected from among the relays that are added to the decoding set first. In a dense network, the forwarding node will almost always be within band B_1 and so there will not be much forward progress. However, as the density decreases, the probability that the forwarding node is in B_1 decreases. In a less-dense network, it becomes likely that the forwarding node is in some further ring B_m, where $m > 1$, implying that each hop will have more forward progress.

To further explain this phenomenon, Figure 5 shows the average message advancement Avg(j) in the network per superslot as a function of node density, where

$$\text{Avg}(j) = \sum_{k=1}^{\nu R_M} \sum_{m=1}^{M} \left(\frac{k}{\nu}\right) \omega(j, k, M, m). \tag{15}$$

Notice that in Figure 5, the message progress is actually larger in networks with lower density, indicating that nodes closer to the destination are more likely to be chosen as a relay. This leads to smaller end-to-end delay at low node densities, as shown in Figure 3.

4.3. *Number of ARQ transmissions*

In this section, we investigate the average number of ARQ transmissions required for the message to reach the destination. Since all ARQ frames are transmitted with the same energy, the average number of ARQ transmissions is related to the energy efficiency of the protocol. We note that there are other issues that impact the energy efficiency of the protocol, such as how RTS, CTS, and other signaling packets are handled. However, these issues are highly implementation-dependent and outside the scope of the paper. Also, the energy consumed transmitting short control packets is generally less than the energy when transmitting the longer message frames. Another very important issue dictating energy efficiency is the duty cycle of the nodes themselves, as often

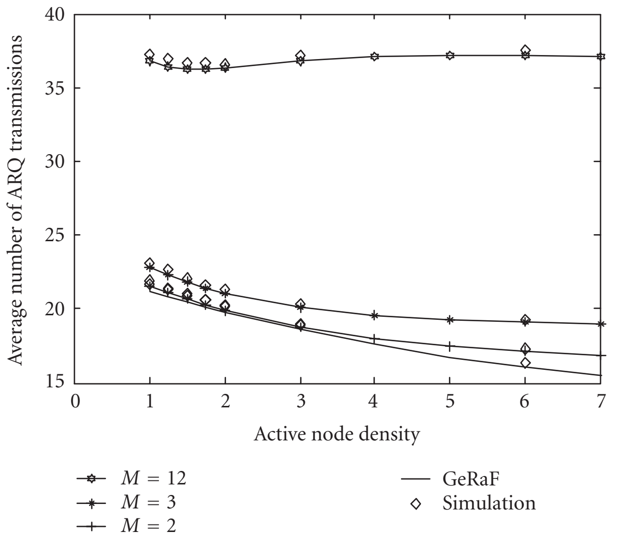

Figure 6: Lower bound on the average number of ARQ transmissions per message in slow HARBINGER A under the same conditions used in Figure 2.

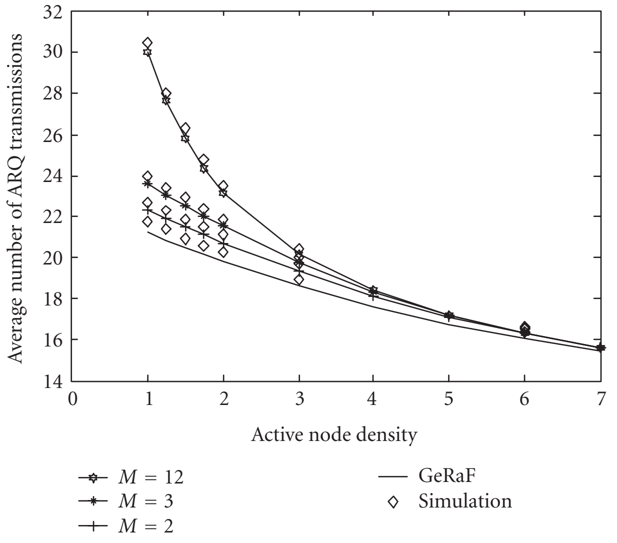

Figure 7: Lower bound on the average number of ARQ transmissions per message in slow HARBINGER B under the same conditions used in Figure 2.

the energy required for a node just to stay awake is similar to the amount of RF power required for it to transmit [4].

As with the delay, the upper and lower bounds on the number of end-to-end ARQ transmissions are tight for sufficiently high ν (e.g., the $\nu = 50$ used here), and so in this section, we only plot the lower bounds for all three versions of HARBINGER in Figures 6, 7, and 8 for $r = 1$, $\mu = 3$, $\nu = 50$, and $D = 10$. Simulation results are also provided. Notice that in all three figures, HARBINGER requires more frames

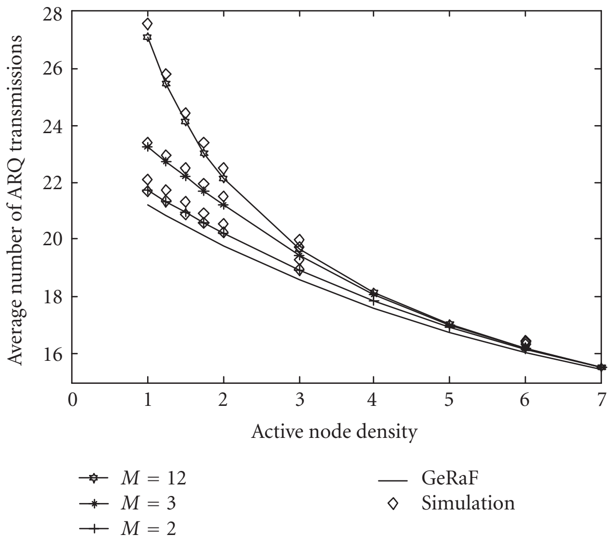

Figure 8: Lower bound on the average number of ARQ transmissions per message in fast HARBINGER under the same conditions used in Figure 2.

to be transmitted per message than GeRaF, and the number of required transmissions increases with M. At first glance, this would imply that the energy efficiency of HARBINGER is much worse than that of GeRaF. This would be true if the energy-latency tradeoff was the same and if nodes only consumed energy when they transmitted. However, the key benefit of HARBINGER is that it allows a lower node density to achieve the same latency target, and thus nodes can save a very significant amount of energy by remaining in a sleep state for a higher proportion of time. We also note that, as shown in [1], additional energy savings can be achieved by removing the memory-flushing condition from the network, though this greatly complicates the analysis and requires nodes to remain in a ready state longer.

Further, notice that although both slow HARBINGER and fast HARBINGER require more ARQ transmissions than GeRaF in low-density networks, they all converge to GeRaF in high-density networks. In fact, as $\rho \to \infty$, both slow HARBINGER B and fast HARBINGER asymptotically require $\lfloor D + 1 \rfloor$ ARQ transmissions for each message. As M gets fairly large, that is, $M = 12$, the message delay of fast HARBINGER is almost equivalent to the average number of ARQ transmissions per message, indicating that with fast HARBINGER, there almost always exists at least one relay in the first coverage band (B_1). Since the delay performance yields diminishing returns of high values of M and the number of ARQ transmissions increases with M, it seems most appropriate to pick a rate constraint of about $M = 2$ or $M = 3$. Fortunately, use of a lower rate constraint also simplifies many of the implementation details.

4.4. Diversity combining versus code combining

HARBINGER with incremental redundancy and code combining always outperforms its repetition coding and diversity

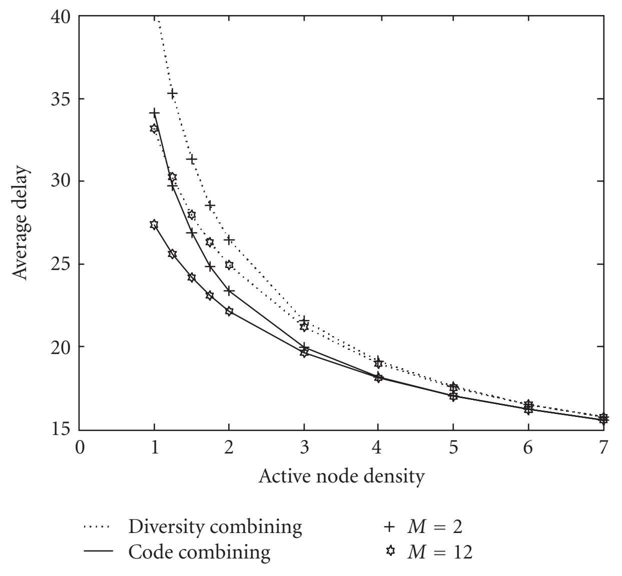

FIGURE 9: Lower bound on message delay (in slots) for fast HARBINGER with diversity combining and code combining under rate constraints $M = \{2, 12\}$, perframe code rate $r = 1$, path loss exponent $\mu = 3$, $\nu = 50$ increments per unit distance, and source-destination distance $D = 10$.

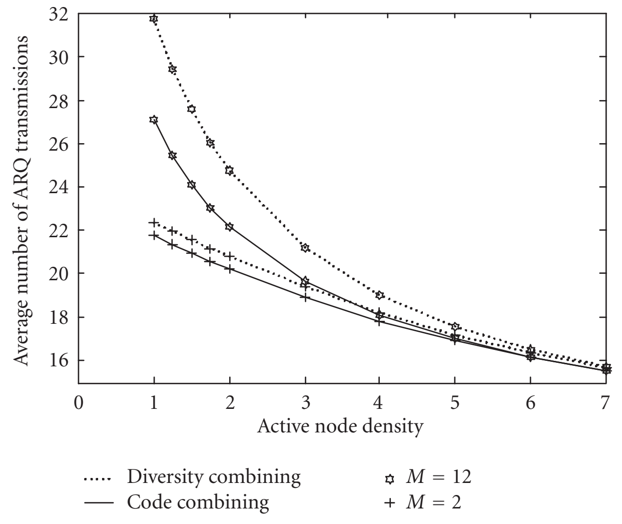

FIGURE 10: Lower bound on the average number of ARQ transmissions required per message for fast HARBINGER with diversity combining and code combining under the same conditions used in Figure 9.

combining counterpart. However, code combining is more complex than diversity combining, and therefore will require more complicated hardware which consumes more power to *process* the ARQ frames. The question remains whether the extra complexity required by code combining is justified by its superior performance. In Figures 9 and 10, we compare the performance of fast HARBINGER with code

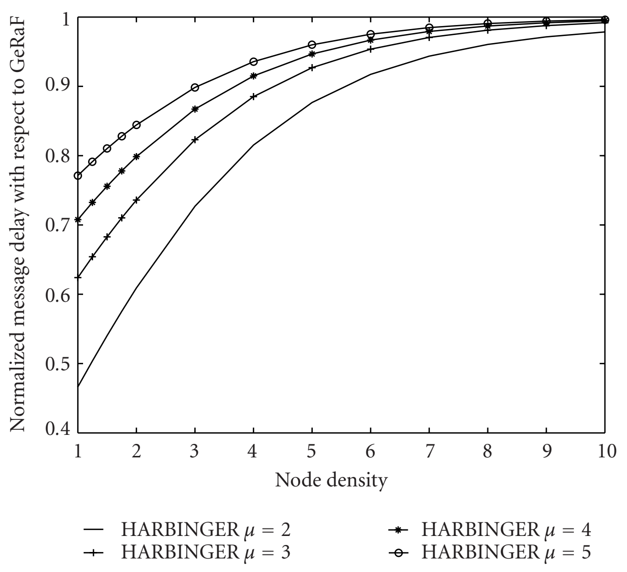

FIGURE 11: The influence of different propagation exponents on the latency of fast HARBINGER (relative to GeRaF) with rate constraint $M = 2$, code combining, perframe code rate $r = 1$, $\nu = 50$ increments per unit distance, and source-destination distance $D = 10$.

combining against fast HARBINGER with diversity combining for $M = 2$ and 12. The extension to slow HARBINGER is straightforward. We observe that diversity combining performs consistently worse than code combining in terms of message delay and energy efficiency. However, under a small rate constraint, for example, $M = 2$, the energy-efficiency improvement of code combining over diversity combining becomes marginal. If we further take into account the processing energy savings in the receiver, diversity combining turns out to be a very attractive low-cost extension to the GeRaF protocol. In addition, we note that HARBINGER with code combining reduces to its diversity combining counterpart for low per-block code rate r since

$$\lim_{r\to 0}\left(\frac{2^{2r}-1}{2^{2r/m}-1}\right)^{1/\mu} = \lim_{r\to 0}\left(\frac{2r\ln 2 + \mathcal{O}(r^2)}{2r\ln 2/m + \mathcal{O}(r^2)}\right)^{1/\mu} = m^{1/\mu}. \tag{16}$$

4.5. Path loss effect

While the previous results were entirely for a path loss exponent $\mu = 3$, we also explored the impact of μ on the performance of the HARBINGER protocol. In particular, Figures 11 and 12 show the delay of fast HARBINGER, normalized with respect to the delay of GeRaF, for $M = 2, 12$ and $\mu = \{2, 3, 4, 5\}$. Notice that HARBINGER always provides considerable gain in terms of average delay over GeRaF regardless of propagation coefficient, although the gain tends to decrease in environments with high path loss.

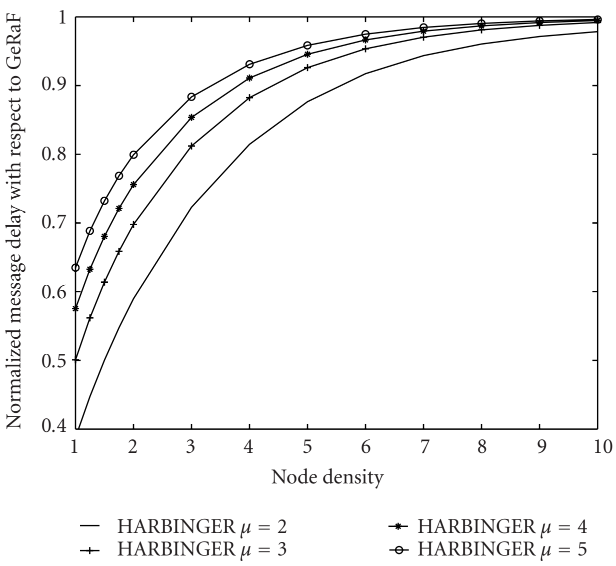

FIGURE 12: The influence of different propagation exponents on the latency of fast HARBINGER (relative to GeRaF) with rate constraint $M = 12$ and the other conditions used in Figure 11.

5. CONCLUSIONS

By introducing a cross-layer hybrid-ARQ mechanism into the GeRaF protocol, significant improvements in the trade-off between latency and active node densities can be achieved. While the total number of transmitted ARQ packets increases with the rate constraint M of the hybrid-ARQ mechanism, most of the latency improvements are realized when the hybrid-ARQ protocol uses small values of M, such as $M = 2$ or 3. For such values of M, it is possible to reduce the active node density by a factor of two or more, implying that nodes will be able to conserve a significant amount of energy by remaining asleep longer. Alternatively, for the same node densities, a lower end-to-end latency can be achieved.

In this paper, the channel was impaired by only exponential path loss and AWGN. Furthermore, it was assumed that the hybrid-ARQ mechanism used capacity-approaching coding and that the control signaling was perfect. The benefit of making these assumptions is that it permits an elegant recursive analysis that very accurately bounds the information-theoretic performance limits. These limits show the benefit of the proposed modified HARBINGER protocols relative to GeRaF and serve as a motivation for further study into practical aspects of the protocol. Issues that should be considered in future research include the practical implementation of control signaling; the performance of actual FEC codes, modulation formats, and receivers; and the impact of interference, collisions, and fading. However, such effects are quite complicated and can only be assessed through simulation which can be very time-intensive for large networks.

APPENDIX

Suppose the source is located at coordinates $(D, 0)$ and the destination at $(0, 0)$, as in [6]. First define the *coverage disk* O_m to be the circular region with radius R_m and center $(D, 0)$. The mth *coverage ring* $\mathbb{R}_m$ is then defined as $\mathbb{R}_m = O_m - O_{m-1}$. Nodes in $\mathbb{R}_m$ require m ARQ frames to decode the message. Under rate constraint M, there are altogether M coverage rings.

Likewise, the *distance disk* Q_k is defined as a circular region with radius k/ν and center $(0, 0)$, where ν denotes the number of increments per unit distance. The kth *distance interval* $\triangle_k$ is defined as $\triangle_k = Q_k - Q_{k-1}$. With a quantization level $1/\nu$, the separation distance D between the source and destination is divided into νD distance intervals. Finally, we define the *coverage band* B_m as the geographically advantaged region in the mth coverage ring, for example, $B_m = \mathbb{R}_m \cap Q_{\nu D}$.

The coverage rings and distance intervals divide the geographically advantaged region $O_M \cap Q_{\nu D}$ into a two-dimensional grid of partitions. Each partition $\mathcal{S}_{m,j}$ is defined as the intersection of the mth coverage ring and the jth distance interval,

$$\mathcal{S}_{m,j} = \begin{cases} (O_m - O_{m-1}) \cap (Q_j - Q_{j-1}) \\ \quad \text{for } (D - R_m)\nu + 1 \le j \le \nu D,\ 1 \le m \le M, \\ \varnothing \quad \text{for } j < (D - R_m)\nu + 1,\ 1 \le m \le M, \end{cases} \tag{A.1}$$

where m and j are nonnegative integers. Any active node in $\mathcal{S}_{m,j}$ is able to decode the message by receiving exactly m ARQ frames from the source. It is straightforward to show that

$$\begin{gathered} \bigcup_m \bigcup_j \mathcal{S}_{m,j} = O_M \cap Q_{\nu D}, \\ \mathcal{S}_{m,j} \cap \mathcal{S}_{n,k} = \varnothing \quad \text{if } m \neq n, \text{ or } j \neq k. \end{gathered} \tag{A.2}$$

We further define the following regions:

$$\begin{aligned} \mathcal{S}_{m^-,j} &= \bigcup_{n=1}^{m} \mathcal{S}_{n,j} = O_m \cap (Q_j - Q_{j-1}), \\ \mathcal{S}_{m,j^-} &= \bigcup_{k=1}^{j} \mathcal{S}_{m,k} = (O_m - O_{m-1}) \cap Q_j, \\ \mathcal{S}_{m^-,j^-} &= \bigcup_{n=1}^{m} \bigcup_{k=1}^{j} \mathcal{S}_{n,k} = O_m \cap Q_j. \end{aligned} \tag{A.3}$$

Notice that (A.3) are general definitions which may result in an empty set under certain conditions, for instance when $j \le (D - R_m)\nu$, $\mathcal{S}_{m^-,j^-} = \varnothing$.

Let $X^t_{\bullet,\circ}$ denote the event that region $\mathcal{S}_{\bullet,\circ}$ contains at least one potential relay during the tth slot, where "$\bullet$" corresponds to either m or m^- in (A.3) and "$\circ$" corresponds to either j or j^- in (A.3). Whenever $\mathcal{S}_{\bullet,\circ} = \varnothing$, its corresponding event probabilities are $X^t_{\bullet,\circ} = 0$ and $\bar{X}^t_{\bullet,\circ} = 1$ (a bar over an event denotes its complement). Although the time index t is necessary to trace the performance of fast HARBINGER, for slow HARBINGER, $X^t_{\bullet,\circ}$ simply reduces to $X_{\bullet,\circ}$, since the network topology remains fixed for the entire superslot.

In this appendix, we will derive two important event probabilities, namely $\omega(\nu D, k, b, m)$ and $\omega_0(\nu D)$. $\omega(\nu D, k, b, m)$ is a joint probability, where νD is the number of increments separating the source and destination, k is the forward progress (in increments) of the message during the current hop, b is the number of slots that have elapsed for the current hop, and m is the number of received ARQ frames during the current hop. We define the *empty hop probability* $\omega_0(j)$ to be the probability that no forward progress has been made in the current hop when the source is j increments from the destination.

A. Slow HARBINGER

Slow HARBINGER with a coverage radius R_M is a straightforward extension of GeRaF in the sense that its network topology remains fixed every superslot, therefore every hop always takes M slots. Different relay selection criteria lead to the two variations on slow HARBINGER, and consequently affect the event probability $\omega(\nu D, k, b, m)$. In particular, slow HARBINGER A selects a relay that is within the distance ring with smallest index (closest to the destination) to minimize the message delay, while slow HARBINGER B selects a relay that is within the coverage ring with smallest index (reachable with minimum number of ARQ frames) to minimize the number of ARQ transmissions.

First consider slow HARBINGER A. When $D > R_M$, in order to make a forward progress of k increments in the current hop with m ARQ frames, $\mathcal{S}_{M^-,(\nu D-k)^-}$ should be empty (otherwise, a forward progress larger than k increments might occur). In addition, $\mathcal{S}_{(m-1)^-,\nu D-k+1}$ should be empty (otherwise, fewer ARQ frames are necessary to achieve the same forward progress), while $\mathcal{S}_{m,\nu D-k+1}$ should be nonempty. Likewise, in order to make the same forward progress with m ARQ frames under slow HARBINGER B, bands $\{B_1, \dots, B_{m-1}\}$ should be empty. In addition, $\mathcal{S}_{m,(\nu D-k)^-}$ in band B_m should be empty while $\mathcal{S}_{m,\nu D-k+1}$ should be nonempty. Therefore, the joint probability $\omega(\nu D, k, M, m)$ becomes

$$\omega(\nu D,k,M,m) = \begin{cases} \Pr\{\bar{X}_{M^-,(\nu D-k)^-}\}\Pr\{X_{m,\nu D-k+1}\cap\bar{X}_{(m-1)^-,\nu D-k+1}\} \\ \qquad \text{for slow HARBINGER A,} \\ \Pr\{\bar{X}_{(m-1)^-,(\nu D)^-}\}\Pr\{X_{m,\nu D-k+1}\cap\bar{X}_{m,(\nu D-k)^-}\} \\ \qquad \text{for slow HARBINGER B.} \end{cases} \tag{A.4}$$

An empty hop occurs when all coverage bands are empty, therefore

$$\omega_0(\nu D) = \Pr\{\bar{X}_{M^-,(\nu D)^-}\}. \tag{A.5}$$

Given a two-dimensional Poisson process, individual event probabilities in (A.4) and (A.5) could be evaluated as

$$\begin{aligned}
\Pr\{\bar{X}_{M^-,(\nu D-k)^-}\} &= \exp\Big\{-\rho A\Big(D, D-\frac{k}{\nu}, R_M\Big)\Big\}, \\
\Pr\{\bar{X}_{(m-1)^-,(\nu D)^-}\} &= \exp\{-\rho A(D,D,R_{m-1})\}, \\
\Pr\{X_{m,\nu D-k+1}\cap\bar{X}_{(m-1)^-,\nu D-k+1}\} &\\
= \exp\Big\{\rho\Big(A\Big(D, D-\frac{k}{\nu}, R_{m-1}\Big) &- A\Big(D, D-\frac{k-1}{\nu}, R_{m-1}\Big)\Big)\Big\} \\
- \exp\Big\{\rho\Big(A\Big(D, D-\frac{k}{\nu}, R_m\Big) &- A\Big(D, D-\frac{k-1}{\nu}, R_m\Big)\Big)\Big\}, \\
\Pr\{X_{m,\nu D-k+1}\cap\bar{X}_{m,(\nu D-k)^-}\} &\\
= \exp\Big\{-\rho\Big(A\Big(D, D-\frac{k}{\nu}, R_m\Big) &- A\Big(D, D-\frac{k}{\nu}, R_{m-1}\Big)\Big)\Big\} \\
- \exp\Big\{-\rho\Big(A\Big(D, D-\frac{k-1}{\nu}, R_m\Big) & \\
- A\Big(D, D-\frac{k-1}{\nu}, R_{m-1}\Big)\Big)\Big\}, &\\
\Pr\{\bar{X}_{M^-,(\nu D)^-}\} &= \exp\{-\rho A(D,D,R_M)\},
\end{aligned} \tag{A.6}$$

where $A(D, r_1, r_2)$ denote the area of intersection of two circles with radii r_1 and r_2 separated by a center-to-center distance of D. This area is indicated in Figure 1 and is computed using (4).

On the other hand, when $D \le R_M$ and particularly if the destination is located in the pth coverage band, the geographically advantaged region is not empty, therefore

$$\omega_0(\nu D) \equiv 0. \tag{A.7}$$

In this case, slow HARBINGER A will forward the message directly to the destination during the very first hop, and thus

$$\omega(\nu D,k,M,m) = \begin{cases} 1, & m = p, \\ 0 & \text{otherwise.} \end{cases} \tag{A.8}$$

With slow HARBINGER B, nodes closer to the source might be chosen as the forwarding relay, therefore

$$\omega(\nu D,k,M,m) = \begin{cases} \Pr\{\bar{X}_{(p-1)^-,(\nu D)^-}\} & \text{for } k = \nu D,\ m = p, \\ \Pr\{\bar{X}_{(m-1)^-,(\nu D)^-}\cap X_{m,\nu D-k+1}\cap\bar{X}_{m,(\nu D-k)^-}\} & \\ & \text{for } k \le R_{p-1}\nu,\ m \le p-1, \\ 0 & \text{otherwise.} \end{cases} \tag{A.9}$$

B. Fast HARBINGER

Unlike slow HARBINGER, cluster $\mathcal{C}(s)$ changes from slot to slot. The source has no a priori knowledge regarding which node will be chosen as the forwarding relay and when. Therefore, the message delay and number of ARQ transmissions required for each hop are heavily influenced by the time-varying nature of network.

First, consider the event probability of empty hop. An empty hop occurs if and only if the following joint event occurs: bands $\{B_1, B_2, \dots, B_M\}$, for example $\mathcal{S}_{M^-,(\nu D)^-}$, are empty during s_1; bands $\{B_1, B_2, \dots, B_{M-1}\}$, for example $\mathcal{S}_{(M-1)^-,(\nu D)^-}$, are empty during s_2; and so forth; band $\{B_1\}$, for example, $\mathcal{S}_{1^-,(\nu D)^-}$, is empty during s_M. Notice that during s_2, band B_M does not need to be empty because it does not affect the event probability of empty hop. In particular, nodes in band B_M need M ARQ frames to decode the message. When they just awake during s_2, they have already missed the first ARQ frame and the remaining $M-1$ ARQ frames are not enough for these nodes to decode. For the same reason, only $\mathcal{S}_{(M-2)^-,(\nu D)^-}$ needs to be empty during s_3, and so forth and $\mathcal{S}_{1^-,(\nu D)^-}$ needs to be empty during s_M. Therefore, the corresponding event probability could be summarized as

$$\omega_0(\nu D) = \Pr\left\{\bigcap_{t=1}^{M} \bar{X}^t_{(M+1-t)^-,(\nu D)^-}\right\}. \quad (A.10)$$

Deriving $\omega(\nu D, k, b, m)$ for fast HARBINGER is fairly complicated. Instead, we study a slightly different event probability $\Omega(\nu D, k, b, m)$. $\Omega(\nu D, k, b, m)$ is a joint event probability, where k denotes the forward progress, b denotes the message delay, and m indicates that at most m ARQ frames will be transmitted in the current session/hop.

It is straightforward to show that

$$\omega(\nu D, k, b, m) = \Omega(\nu D, k, b, m) - \Omega(\nu D, k, b, m-1). \quad (A.11)$$

Consider $D > R_M$. As a simple example, first assume that $M = 2$. Notice that $m \le b$; thus $\Omega(\nu D, k, b, m)$ has no-zero value only for three cases, for example, $b = 1$, $m = 1$; $b = 2$, $m = 1$; and $b = 2$, $m = 2$. More specifically, their corresponding event probability is

$$\Omega(\nu D, k, b, m) = \begin{cases} \Pr\{\bar{X}^1_{1,(\nu D-k)^-} \cap X^1_{1,\nu D-k+1}\}, & b = m = 1, \\ \Pr\{\bar{X}^1_{2^-,(\nu D)^-} \cap \bar{X}^2_{1,(\nu D-k)^-} \cap X^2_{1,\nu D-k+1}\}, & b = 2,\ m = 1, \\ \Pr\{\bar{X}^1_{1^-,(\nu D)^-} \cap \bar{X}^1_{2,(\nu D-k)^-} \cap \bar{X}^2_{1,(\nu D-k)^-} \cap (X^2_{1,\nu D-k+1} \cup X^1_{2,\nu D-k+1})\}, & b = m = 2. \end{cases} \quad (A.12)$$

The expressions for $\Omega(\nu D, k, 1, 1)$ and $\Omega(\nu D, k, 2, 1)$ are quite intuitive, and thus the discussion will be focused on joint event probability $\Omega(\nu D, k, 2, 2)$. In particular, band B_1 should be empty during s_1 (otherwise, the current session/hop will terminate with only 1 slot of message delay). In addition, $\mathcal{S}_{2,(\nu D-k)^-}$ should be empty during s_1 and $\mathcal{S}_{1,(\nu D-k)^-}$ should be empty during s_2, otherwise a message progress greater than k increments might occur. Finally, to make a forward progress of k increments, distance interval $\triangle_{\nu D-k+1}$ should be nonempty. In particular, a nonempty $\mathcal{S}_{1,\nu D-k+1}$ during s_2 and/or a nonempty $\mathcal{S}_{2,\nu D-k+1}$ during s_1 ensures that at most 2 ARQ frames are transmitted during the hop.

Following the same rationale, (A.12) could be generalized for $M > 2$. In particular, in order to make a forward progress of k increments with b slots of message delay and at most m ARQ frames, the following sequence of events should occur. First of all, bands $\{B_1, B_2, \ldots, B_{m-1}\}$ should be empty during s_{b-m+1}; bands $\{B_1, B_2, \ldots, B_{m-2}\}$ should be empty during s_{b-m+2}; $\ldots$; bands $\{B_1\}$ should be empty during s_{b-1} (otherwise, the current session/hop will terminate with a delay smaller than b). Secondly, bands $\{B_1, B_2, \ldots, B_M\}$ should be empty during $\{s_1, s_2, \ldots, s_{b-m}\}$ (otherwise, more than m ARQ frames will be transmitted). In addition, $\mathcal{S}_{m,(\nu D-k)^-}$ should be empty during s_{b-m+1}; $\mathcal{S}_{m+1,(\nu D-k)^-}$ should be empty during s_{b-m+2}; and so forth; $\mathcal{S}_{1,(\nu D-k)^-}$ should be empty during s_b (otherwise, a forward progress greater than k increments might occur). Finally, to make a forward progress of k increments in the current hop, at least one of the following events should occur: a nonempty $\mathcal{S}_{m,(\nu D-k+1)^-}$ during s_{b-m+1}; a nonempty $\mathcal{S}_{m-1,(\nu D-k+1)^-}$ during s_{b-m+2}; and so forth; a nonempty $\mathcal{S}_{1,(\nu D-k+1)^-}$ during s_b. In summary,

$$\begin{aligned} &\Omega(\nu D, k, b, m) \\ &\quad = \Pr\left\{\left(\bigcap_{t=1}^{b-m} \bar{X}^t_{M^-,(\nu D)^-}\right) \cap \left(\bigcap_{t=b-m+1}^{b-1} \bar{X}^t_{(b-t)^-,(\nu D)^-}\right)\right. \\ &\qquad\qquad \left. \cap \left(\bigcap_{l=1}^{m} \bar{X}^{b+1-l}_{l,(\nu D-k)^-}\right) \cap \left(\bigcup_{l=1}^{m} X^{b+1-l}_{l,\nu D-k+1}\right)\right\}, \end{aligned} \quad (A.13)$$

which could be further decomposed into a product of conditional probabilities

$$\begin{aligned} &\Omega(\nu D, k, b, m) \\ &\quad = \Pr\left\{\left(\bigcap_{l=1}^{m} \bar{X}^{b+1-l}_{l,(\nu D-k)^-}\right) \cap \left(\bigcup_{l=1}^{m} X^{b+1-l}_{l,\nu D-k+1}\right)\right. \\ &\qquad\qquad \left| \left(\bigcap_{t=1}^{b-m} \bar{X}^t_{M^-,(\nu D)^-}\right) \cap \left(\bigcap_{t=b-m+1}^{b-1} \bar{X}^t_{(b-t)^-,(\nu D)^-}\right)\right\} \\ &\qquad \Pr\left\{\left(\bigcap_{t=b-m+1}^{b-1} \bar{X}^t_{(b-t)^-,(\nu D)^-}\right) \middle| \left(\bigcap_{t=1}^{b-m} \bar{X}^t_{M^-,(\nu D)^-}\right)\right\} \\ &\qquad \times \Pr\left\{\left(\bigcap_{t=1}^{b-m} \bar{X}^t_{M^-,(\nu D)^-}\right)\right\}. \end{aligned} \quad (A.14)$$

Given a two-dimensional Poisson distributed network, when partition A of the network is empty during s_i, the node distribution of partition B will follow Poisson process during s_{i+1} as long as partition B is a subset of partition A. Therefore, each term in (A.14) could be computed as

$$\begin{aligned}
&\Pr\left\{\bigcap_{t=1}^{b-m}\bar{X}^{t}_{M^-,(\nu D)^-}\right\}\\
&\quad=\Pr\{\bar{X}^{1}_{M^-,(\nu D)^-}\}\Pr\{\bar{X}^{2}_{M^-,(\nu D)^-}\mid\bar{X}^{1}_{M^-,(\nu D)^-}\}\cdots\\
&\qquad\times\Pr\left\{\bar{X}^{b-m}_{M^-,(\nu D)^-}\,\middle|\,\bigcap_{t=1}^{b-m-1}\bar{X}^{t}_{M^-,(\nu D)^-}\right\}\\
&\quad=\left(\Pr\{\bar{X}_{M^-,(\nu D)^-}\}\right)^{b-m}\\
&\quad=\left(\exp\{-\rho A(D,D,R_M)\}\right)^{b-m}\\
&\Pr\left\{\bigcap_{t=b-m+1}^{b-1}\bar{X}^{t}_{(b-t)^-,(\nu D)^-}\,\middle|\,\bigcap_{t=1}^{b-m}\bar{X}^{t}_{M^-,(\nu D)^-}\right\}\\
&\quad=\Pr\{\bar{X}_{(m-1)^-,(\nu D)^-}\}\Pr\{\bar{X}_{(m-2)^-,j^-}\}\cdots\Pr\{\bar{X}_{1^-,(\nu D)^-}\}\\
&\quad=\prod_{i=1}^{m-1}\exp\{-\rho A(D,D,R_i)\},\\
&\Pr\left\{\left(\bigcap_{l=1}^{m}\bar{X}^{b+1-l}_{l,(\nu D-k)^-}\right)\cap\left(\bigcup_{l=1}^{m}X^{b+1-l}_{l,\nu D-k+1}\right)\right.\\
&\qquad\left.\left|\left(\bigcap_{t=1}^{b-m}\bar{X}^{t}_{M^-,(\nu D)^-}\right)\cap\left(\bigcap_{t=b-m+1}^{b-1}\bar{X}^{t}_{(b-t)^-,(\nu D)^-}\right)\right.\right\}\\
&\quad=\Pr\left\{\left(\bigcap_{l=1}^{m}\bar{X}_{l,(\nu D-k)^-}\right)\cap\left(\bigcup_{l=1}^{m}X_{l,\nu D-k+1}\right)\right\}\\
&\quad=\exp\{-\rho A(D,D-\frac{k}{\nu},R_m)\}\\
&\qquad-\exp\left\{-\rho A\left(D,D-\frac{k-1}{\nu},R_m\right)\right\}.
\end{aligned}\tag{A.15}$$

Likewise, the closed-form expression for $\omega_0(\nu D)$ becomes

$$\omega_0(\nu D)=\prod_{i=1}^{M}\exp\{-\rho A(D,D,R_i)\}.\tag{A.16}$$

When $D\le R_M$, the destination is located in O_M, therefore the geographically advantaged region is not empty, thus $w_0(\nu D)\equiv 0$ and $\Omega(\nu D,k,b,m)=0$ when $m\ne b$. More specifically, suppose that the destination is located within the pth coverage band. If a forward progress of $k\le\nu R_{p-1}$ is to be made with a message delay $b\le p-1$, the event probability becomes

$$\begin{aligned}
&\Omega(\nu D,k,b,b)\\
&\quad=\Pr\left\{\left(\bigcap_{m=1}^{b-1}\bar{X}^{b-m}_{m^-,(\nu D)^-}\right)\cap\left(\bigcap_{m=1}^{b}\bar{X}^{b+1-m}_{m,(\nu D-k)^-}\right)\right.\\
&\qquad\left.\cap\left(\bigcup_{m=1}^{b}X^{b+1-m}_{m,\nu D-k+1}\right)\right\}\\
&\quad=\Pr\left\{\left(\bigcap_{m=1}^{b}\bar{X}^{b+1-m}_{m,(\nu D-k)^-}\right)\cap\left(\bigcup_{m=1}^{b}X^{b+1-m}_{m,\nu D-k+1}\right)\right.\\
&\qquad\left.\left|\left(\bigcap_{m=1}^{b-1}\bar{X}^{b-m}_{m^-,(\nu D)^-}\right)\right.\right\}\Pr\left\{\bigcap_{m=1}^{b-1}\bar{X}^{b-m}_{m^-,(\nu D)^-}\right\}\\
&\quad=\left[\exp\left\{-\rho A\left(D,D-\frac{k}{\nu},R_b\right)\right\}\right.\\
&\qquad\left.-\exp\left\{-\rho A\left(D,D-\frac{k-1}{\nu},R_b\right)\right\}\right]\\
&\qquad\cdot\prod_{i=1}^{b-1}\exp\{-\rho A(D,D,R_i)\}.
\end{aligned}\tag{A.17}$$

If on the other hand, to make a forward progress of $k>\nu R_{p-1}$, the destination should always be chosen as the relay, thus $b=p$ and $k=\nu D$,

$$\begin{aligned}
\Omega(\nu D,k,b,b)&=\Pr\left\{\left(\bigcap_{m=1}^{b-1}\bar{X}^{b-m}_{m^-,(\nu D)^-}\right)\right\}\\
&=\prod_{i=1}^{b-1}\exp\{-\rho A(D,D,R_i)\}.
\end{aligned}\tag{A.18}$$

Otherwise, $\Omega(\nu D,k,b,b)=0$.

ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research under Grant N00014-00-0655. Parts of this paper have appeared at the 2004 IEEE Military Communications Conference (MILCOM) and the 2004 IEEE Global Telecommunications Conference (GLOBECOM).

REFERENCES

[1] B. Zhao and M. C. Valenti, "Practical relay networks: A generalization of hybrid-ARQ," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 7–18, 2005.

[2] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Resource management in energy-limited, bandwidth-limited, transciever-limited wireless networks for session-based multicasting," *Computer Networks*, vol. 39, no. 2, pp. 113–131, 2002.

[3] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[4] R. Min, M. Bhardwaj, S.-H. Cho, et al., "Energy-centric enabling technologies for wireless sensor networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 28–39, 2002.

[5] G. Caire and D. Tuninetti, "The throughput of hybrid-ARQ protocols for the Gaussian collision channel," *IEEE Trans. Inform. Theory*, vol. 47, no. 5, pp. 1971–1988, 2001.

[6] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, 2003.

[7] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 349–365, 2003.

[8] 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 Telecommunications Conference (GLOBECOM '04)*, vol. 5, pp. 3047–3052, Dallas, Tex, USA, November–December 2004.

[9] M. C. Valenti and B. Zhao, "Hybrid ARQ-based intra-cluster geographically-informed relaying," in *Proc. IEEE Military Communication Conference (MILCOM '04)*, Monterey, Calif, USA, November 2004.

[10] S. Wicker, *Error Control Systems for Digital Communications and Storage*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.

[11] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[12] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*, Springer, New York, NY, USA, 1991.

Bin Zhao received the B.S.E.E. and M.S.E.E. degrees from Shanghai Jiaotong University, Shanghai, China, in 1995 and 1998, respectively. He received a Ph.D. degree in electrical engineering from West Virginia University (Morgantown, WVa, USA) in 2004, where he worked as a Research Assistant in the Wireless Communications Research Laboratory. He is currently a communications engineer in Efficient Channel Coding Inc. (Brooklyn Heights, Ohio, USA). His research interests are in the areas of communication theory, error-correction coding, sensor networks, and information theory. Prior to attending graduate school at West Virginia University, he was a DSP engineer with Huawei Technologies Co. Ltd. where he was engaged in the development of real-time speech and channel codec for IS-95 system.

Matthew C. Valenti received a B.S.E.E. degree from Virginia Tech, Blacksburg (USA), in 1992, an M.S.E.E. degree from the Johns Hopkins University (Baltimore, Md, USA) in 1995, and a 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 at West Virginia University (Morgantown, WVa, USA). He serves as an Associate Editor for IEEE Transactions on Vehicular Technology, and has been on the technical program committee for several international conferences. His research interests are in the areas of communication theory, error-correction coding, applied information theory, and wireless multiple-access networks. He also acts as 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. Prior to attending graduate school at Virginia Tech, he was an electronics engineer at the United States Naval Research Laboratory (Washington, DC, USA) 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.