Some Results on Two Forms of Erasure-Correction Coding for Packet Radio Networks

Siddhartha S. Borkotoky, Michael C. Dowling, and Michael B. Pursley

***Abstract*—We present some preliminary results from our investigation of two forms of erasure-correction coding, fountain coding and network coding, when they are employed for file transfers in a distributed packet radio network. The radios adapt the modulation and channel coding of the transmitted signals in response to variations in fading on the links of the network. The control information for the adaptive transmission protocol is provided by simple statistics that are derived in the demodulators and decoders of the radio receivers.**

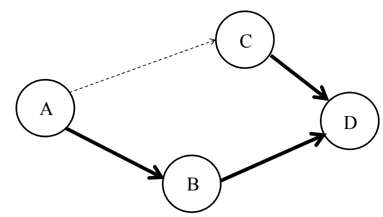

Fig. 1. A network of four half-duplex radios connected by time-varying wireless links.

I. Introduction

We consider the problem of delivering a file, which is divided into a set of packets, to one or more nodes in a distributed, tactical packet radio network consisting of half-duplex radios. The network configuration that we study in this investigation comprises four nodes as shown in Fig. 1. Node A is the source that has a file to transmit. Node D is a destination that requires A's file, whereas nodes B and C may or may not need the file. Suppose that the signal-to-noise ratio (SNR) of link A–D is too poor to support any packet delivery. Therefore, in order to transfer the file from A to D, all packets from A must be routed via B or C. We further assume that links A–B, B–D, and C–D have higher SNR as compared to link A–C.

We demonstrate that, by employing two forms of packet erasure-correction coding, namely network coding [1] and fountain coding [2], the service of both intermediate nodes B and C can be utilized to expedite the file transfer from A to D while avoiding the need to exchange a large amount of control information between the radios in the network. Two classes of protocols are described in this paper, *relay protocols* for scenarios in which the intermediate nodes do not need the file being transmitted by A and *broadcast protocols* for scenarios in which the intermediate nodes also need the file.

An adaptive transmission protocol is employed to ensure efficient packet delivery in the presence of temporal variations in the wireless links between the radios. This protocol adapts the error-control code and the modulation format on a packet-by-packet basis in response to changes in channel conditions. The adaptation is carried out with the help of a channel-quality statistic that is derived at the receiver.

II. Packet Erasure-Correction Coding

We restrict attention to network coding and fountain coding in GF(2). For both types of codes, the file at the source is divided into K fixed-length information-bit sequences, referred to as *information packets*. The encoder chooses information packets at random and combines them linearly using bitwise XOR operations in order to generate coded packets, also referred to as *data packets*. The data packets are encoded with a channel code to form *channel packets*, which are modulated and transmitted over the channel. At the recipient, the received packet is demodulated and then decoded by a decoder for the channel code. If the decoding attempt succeeds, we say that the corresponding data packet has been *recovered*. The recipient is able to retrieve the original K information packets after recovering a set of data packets containing K linearly independent combinations of information packets.

For network codes, the random linear network coding [3] model is employed. The K information packets are divided into g disjoint *generations* [4] of d packets each. In order to generate a data packet, the encoder first chooses a generation at random, and then randomly selects information packets from that generation for combining. The decoder applies Gaussian elimination to each generation to obtain the information packets. For our performance results on fountain coding, the erasure-correction code is the systematic raptor code described in [5] and Gaussian elimination is employed to provide maximum-likelihood decoding [6].

For network coding, $d + \log_2 g$ bits must be added to the header of each data packet to inform the decoder of the encoding vector for the packet. The encoding vector specifies the identity of the generation and the indices of the information packets from that generation that were combined to form the data packet. For fountain coding, all that is needed in the header is a sequence number for the data packet, because the sender and receiver know the mapping between the sequence number and the encoding vector for the fountain code.

Once a recipient is able to obtain all information packets by decoding the network code or the fountain code, it

This research was supported by Army Research Office Grant W911NF-11-1-0427. The authors are with Clemson University, Clemson, SC 29634.

notifies the sender of decoding success by including this information in an acknowledgement packet. When network coding is employed with multiple generations, the recipient also sends a notification to the sender after decoding each generation. For unicast distribution, this prompts the sender to avoid combining information packets from generations that have already been decoded by the recipient. For multicast transmissions, the sender avoids combining packets from generations that have been decoded by all the intended recipients. In the protocols that we propose, the decoding acknowledgements are "local" in the sense that any given acknowledgement must traverse only one link, the link from the recipient to the sender. It does not have to be forwarded to any other radio in the network.

We say that a node *forwards* a data packet that it has received if it simply sends a channel-coded version of the data packet. Thus, *forwarding* a data packet does not involve either network coding or fountain coding. The data packet is not combined with other data packets.

III. Protocols for Packet Relay

In this section, we describe two protocols for relay distribution of a file in the packet radio network. One protocol uses network coding for relay distribution (NC-R) and the other uses fountain coding for relay distribution (FC-R). The variables in the following list are employed in our descriptions of the two relay protocols:

N_r: Number of data packets that B must recover from A before it starts transmitting to D in the NC-R protocol.

N_d: Number of data packets that B must deliver to D before it returns to receiving from A in the NC-R protocol.

N_w: Number of data packets that C must recover from A before it starts transmitting to D in the FC-R protocol.

$N_{\max}$: Maximum number of consecutive transmissions that C is allowed to make to D in the FC-R protocol.

N_f: Number of consecutive packet erasures on link A–C following which C begins forwarding any data packets in its buffer to D in the FC-R protocol.

A. The NC-R Protocol

In the NC-R protocol, node A begins the session by transmitting network-coded packets to B and C. Once B recovers N_r data packets from A, it starts forwarding random linear combinations of those data packets to D. After delivering N_d data packets to D, it starts receiving from A again and continues to do so until it accumulates another N_r data packets. Node C continues to receive data packets from A whenever B is transmitting to D. Nodes B and C communicate with each other by means of control packets so that C knows when B stops transmitting packets to D and returns to receiving from A. At that time, C begins forwarding to D data packets that it has accumulated. In this manner, B and C alternate between receiving from A and transmitting to D, and the process continues until D is able to decode the file. Although nodes B and C do not need the file themselves, they do, however, perform Gaussian elimination to check the rank of their decoding matrices corresponding to those generations for which d or more data packets have been recovered. The generations whose decoding matrices have full rank are identified and this information is conveyed to A by means of feedback packets. A in turn avoids combining information packets from generations that have already been decoded by the recipient for any given transmission.

B. The FC-R Protocol

The FC-R protocol was investigated in [7]. In this scheme, A transmits fountain-coded packets to B and C. B continues receiving data packets from A until it can decode the file. C, on the other hand, starts forwarding data packets to D as soon as it is able to recover N_w data packets from A. C also starts forwarding data packets to D if it has undelivered data packets in its buffer *and* it has encountered N_f consecutive packet failures while trying to receive from A. Once C switches to transmit mode, it continues sending data packets to D until either it has delivered all packets in its buffer or it has made $N_{\max}$ total transmissions. While C switches between receiving and forwarding, B continues to receive from A until it decodes the file, at which point both A and C withdraw from the session. B starts the fountain coding process from the point where A stopped, and then B sends the newly formed data packets to node D until D receives enough data packets to enable it to decode the file.

IV. Protocols for Broadcast

In this section, we describe two protocols for broadcast distribution of a file in a packet radio network. One uses network coding for broadcast distribution (NC-B) and the other uses fountain coding for broadcast distribution (FC-B). In this situation, nodes B, C, and D all wish to have the file that A is distributing, so B and C need to decode the file, not just relay packets that will enable D to do so. The following is a list of variables used in our description:

N_s: Number of data packets that node C must recover from A before it starts transmitting to D.

$N_{\max}$: Maximum number of consecutive transmissions that node C is allowed to make to D.

E_{avg}: The average value of error count per packet calculated at node C.

E_{th}: Threshold to which E_{avg} is compared in order to decide whether C should forward data packets to D.

In both NC-B and FC-B protocols, A begins the session by multicasting network-coded or fountain-coded packets to B and C. If the nominal condition of A–C is relatively good, then node C is allowed to periodically forward some data packets to D. In this case, as soon as C recovers N_s data packets from A, it forwards them to D before coming back to receive from A again. C is allowed to make a maximum of $N_{\max}$ transmission attempts to D at a time in order to deliver the N_s data packets. Until B decodes the file, C continues to alternate between receiving and forwarding. Once B is able to decode the file, it applies network/fountain coding to the information packets and transmits them to D until the latter obtains the file. For the FC-B protocol, B starts its fountain encoder at the point where A's encoder was when B received its last data packet from A. This ensures that D does not accumulate duplicate data packets. Once B starts transmitting, C stops forwarding data packets to D and continues to receive from A until it is able to decode the file. On the other hand, if the nominal condition of the link A–C is relatively poor, then C does not forward any data packets to D during the entire session.

A channel-quality statistic, referred to as the *error count*, is used by the protocol in order to assess the condition of link A–C and decide whether C should forward data packets to D. The error count is the number of binary symbol errors at the output of a recipient's demodulator. The error count may be provided by the channel decoder. If it is not, then the error count can be determined by re-encoding the decoded information bits with the channel code that was used for transmission and comparing the resulting codeword bitwise with the hard-decision demodulator output. Our broadcast protocol maintains a running average of the error count observed at node C. Note that the error count can be obtained only when a channel packet is decoded correctly. If the packet fails to decode, the error count is set to the number of binary code symbols in the packet for the purpose of computing the average. Let the average error count at C after it recovers the first N_s data packets be given by E_{avg}. The value of E_{avg} is compared with a threshold E_{th}. If $E_{\text{avg}} \leq E_{\text{th}}$, then C is allowed to periodically forward data packets to D. Otherwise, only B forwards data packets to D once the former is able to decode the file.

V. An Adaptive Transmission Protocol

Our adaptive transmission protocol adapts the channel code and the modulation format from one packet to the next in response to changes in channel conditions. Let $\mathcal{B} = \{\mathcal{B}_j : 1 \leq j \leq N\}$ denote the set of N code-modulation combinations available to the protocol, indexed in increasing order of the number of binary information symbols per modulation symbol. The protocol chooses a code-modulation combination for each transmission with the aid of the *error count*, which is described in Section IV. Every recipient reports its error count to the sender along with its acknowledgement packet.

For unicast transmissions, the adaptive transmission protocol applies an interval test to the error count reported by the recipient for the previous packet in order to select a combination for the next packet. The partition for the interval test depends on the code-modulation combination that was used with the previous packet. Let E_p be the error count obtained for the previous packet which employed combination $\mathcal{B}_p$. If $\{\mathcal{I}_p(j) : 1 \leq j \leq N\}$ is the partition associated with combination $\mathcal{B}_p$, then combination $\mathcal{B}_n$ is chosen for the next packet transmission if $E_p \in \mathcal{I}_p(n)$. Recall that the error count can be obtained only when the recipient is able to decode the channel packet. In the event of a packet failure, the error count cannot be computed and the sender switches to the next lower-rate code-modulation combination for the next transmission unless it is already using the lowest-rate combination.

For multicast transmissions, additional steps are necessary in order to select a code-modulation combination for the next packet. In our relay and broadcast protocols, any multicast transmission involves two recipients. The adaptive transmission protocol first applies interval tests to the error counts reported by the two recipients in order to obtain two code-modulation indices, referred to as the *suggested indices*. One of two multicast adaptation criteria is then used to select a combination for the next packet based on the two suggested indices. In the *max-index* criterion, the larger among the two indices is used for the next transmission. The *maximum data-recovery rate* or *max-DRR* criterion [8], on the other hand, attempts to maximize the ratio of the expected number of total information bits recovered by the two recipients to the number of time units required for the next transmission.

Recall that all transmissions made by our NC-R protocol are unicast. In the FC-R protocol, the transmissions made by node A can be either unicast or multicast depending on whether node C is transmitting to D or receiving from A, respectively. For multicast transmissions in FC-R, the max-index criterion is used. In the broadcast protocols NC-B and FC-B, the max-DRR criterion is used whenever A transmits packets to B and C simultaneously.

VI. Performance Results

For our numerical results, the file at the source is divided into $K = 500$ information packets of 2400 bits each. The values of the different parameters associated with the protocols are as follows: $N_r = N_d = 20$ for NC-R, $N_w = 20$, $N_{\text{max}} = 50$, $N_f = 3$ for FC-R, and $N_s = 20$, $N_{\text{max}} = 30$, $E_{\text{th}} = 400$ for NC-B and FC-B.

The links between the radios experience time-varying propagation loss due to fading. The four links are assumed to vary independently of one another according to a Nakagami-m fading process [9]. An equal step-size Markov chain with 12 states is used to model the Nakagami-m fading channels. Each state of the Markov chain represents a unique fade level in dB. The transition probabilities and the steady-state probabilities of the Markov chain are derived according to the method described in [10]. While these Markov-chain models allow for any value of m greater than 1/2, we restrict attention to $m = 2.5$ for performance illustrations in this paper. For this particular realization of the fading process, the value of channel gain can vary from -9 dB to 6 dB in steps of 1.25 dB. Channel transitions are not restricted to adjacent states alone. The rapidness of the variations depends on the Doppler frequency of the channel according to the equation $\rho = J_0^2(2\pi f_d T_s)$, where T_s is the average time duration between the start of one packet transmission to the start of the next, f_d is the Doppler frequency, ρ is correlation coefficient for samples of the fading process that are separated in time by T_s, and J_0 is the Bessel function of the first kind of order zero. For our performance results, we set $f_d T_s$ to 0.02, which indicates a relatively fast fading channel.

The set of 13 code-modulation combinations given in Table I is used for packet transmissions. These 13 combinations are derived from a set of five turbo-product codes [11] of rates 0.260, 0.346, 0.472, 0.620, and 0.766, and four modulation formats, 64-biorthogonal key modulation (64-BOK), binary phase-shift key (BPSK), quadriphase shift key (QPSK), and 16-quadrature amplitude modulation (16-QAM). Bit-interleaved coded modu-

TABLE I
LIST OF CODE-MODULATION COMBINATIONS USED BY THE ADAPTIVE TRANSMISSION PROTOCOL.

$\mathcal{B}_n$	Modulation	Code rate
$\mathcal{B}_1$	64-BOK	0.260
$\mathcal{B}_2$	64-BOK	0.472
$\mathcal{B}_3$	64-BOK	0.766
$\mathcal{B}_4$	BPSK	0.260
$\mathcal{B}_5$	BPSK	0.346
$\mathcal{B}_6$	QPSK	0.260
$\mathcal{B}_7$	QPSK	0.346
$\mathcal{B}_8$	QPSK	0.472
$\mathcal{B}_9$	QPSK	0.620
$\mathcal{B}_{10}$	QPSK	0.766
$\mathcal{B}_{11}$	16-QAM	0.472
$\mathcal{B}_{12}$	16-QAM	0.620
$\mathcal{B}_{13}$	16-QAM	0.766

lation is used for transmissions. The metric for iterative soft-decision decoding of the channel codes is the log-likelihood bit metric. The length of each information block for the code of rate 0.260 is 1200; therefore, each channel packet consists of two codewords when this code is used. The other four codes have information blocks of length 2400 bits, hence each channel packet consists of one codeword.

We refer to the elemental rectangular pulses that constitute a modulation symbol as *modulation chips*. The chip duration and the average energy per chip are held constant for all modulation formats in order to maintain a constant spectral occupancy and to limit the interference to other radios. Therefore, we employ the chip-energy to noise-density ratio (CENR) as the measure of SNR. The value of CENR in dB can be computed as $\text{CENR} = 10\log_{10}(\mathcal{E}_c/N_0)$, where $\mathcal{E}_c$ is the average chip energy and N_0 is the one-sided power-spectral density of the additive Gaussian noise. For all our numerical results, it is assumed that the nominal CENR (i.e., the value of CENR in the absence of fading) of links A–B, B–D, and C–D is the same, and we denote its value by CENR*. The nominal CENR of link A–C is CENR$^* - 10\,$dB.

The performance metric that we employ for comparison of the protocols is *session throughput*. For the relay scenario, we compute the session throughput by dividing the total number of information bits delivered to node D by the total number of time units required to complete the session. The session throughput for broadcast is defined as the total number of information bits in the file divided by the total number of time units required to deliver the file to all destinations. One time unit is defined as the duration of one modulation chip.

The performance of the NC-R and FC-R protocols are shown in Fig. 2 along with that of a conventional store-and-forward protocol that does not employ packet erasure-correction coding and utilizes relay B alone. Three generation sizes are considered for the NC-R protocol, namely $d =$ 100, 250, and 500. It can be seen that, at

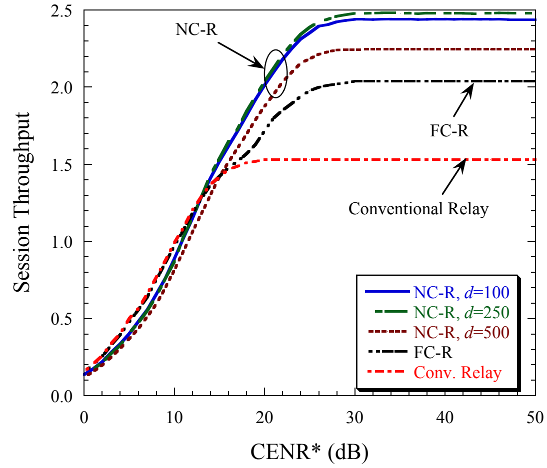

Fig. 2. Performance comparison of the relaying protocols. Links are subject to Nakagami-m fading with $m = 2.5$ and $f_d T_s = 0.02$.

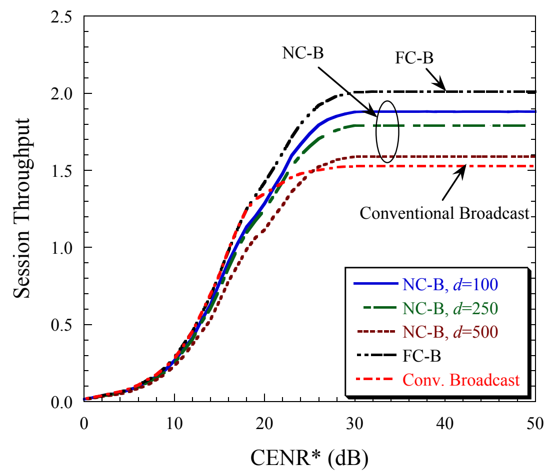

Fig. 3. Performance comparison of the broadcast protocols. Links are subject to Nakagami-m fading with $m = 2.5$ and $f_d T_s = 0.02$.

moderate and high values of CENR*, the three realizations of the NC-R protocol significantly outperform the FC-R protocol, which in turn provides approximately 33% increase in throughput as compared to the conventional store-and-forward protocol. At lower values of CENR*, the FC-R protocol performs very close to the conventional protocol, but NC-R suffers some performance degradation due to the transmission of the encoding vector. Among the three generation sizes for NC-R, $d = 250$ performs the best. However, $d = 100$ provides only slightly lower throughput than $d = 250$.

Fig. 3 shows the performance of the NC-B and FC-B protocols. Also shown is the throughput of a conventional protocol in which only node B transmits packets to node D after the former is able to decode the file itself. This protocol is assumed to use a hypothetical packet erasure-correction code that is able to decode the file with probability 1 after recovering K data packets, i.e., no excess packets are required. The FC-B protocol is found to outperform the conventional protocol by providing throughput improvements of about 33% at high CENR*. The throughput of the NC-B protocols with different generation sizes is lower than the FC-B protocol due to the overhead incurred by transmitting the encoding vector with each packet. However, for large values of CENR*, these protocols still provide significant performance improvements over the conventual broadcast scheme, especially for $d = 100$ and $d = 250$.

VII. Conclusion

We have described protocols for relay and broadcast of data in a half-duplex packet radio network using random linear network coding and fountain coding. Our suggested protocols are able to provide substantial improvements in throughput over conventional techniques. For the simple four-node network, we found that network coding is better than fountain coding if the intermediate nodes do not wish to receive the file but instead perform only a relay function. On the other hand, if the intermediate nodes do wish to receive the file, and therefore must decode it, then we found that fountain coding is better than network coding.

References

[1] T. Ho and D. S. Lun, *Network coding: An introduction*, Cambridge University Press, Cambridge, U.K., 2008.

[2] D. J. C. MacKay, "Fountain codes," *IEE Proceedings – Communications*, vol. 152, no. 6, pp 1062–1068, December 2005.

[3] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," *in Proc. IEEE International Symposium on Information Theory*, pp. 442–447, 2003.

[4] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *in Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2003.

[5] M. Luby, A. Sokrollahi, M. Watson, and T. Stockhammer, "Reliable forward error correction scheme for object delivery," IETF Request for Comments 5053, Oct. 2007.

[6] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inform. Th.*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[7] S. S. Borkotoky and M. B. Pursley, "Preliminary results on the performance of an adaptive protocol for packet relay with fountain coding," *Proc. IEEE Military Communications Conf.*, pp. 362–367, 2014.

[8] J. D. Ellis and M. B. Pursley, "Adaptive capacity-achieving channel coding for fountain-coded multicast transmission in packet radio systems," *IEEE Trans. Wireless Communications*, vol. 12, no. 12, pp. 6514–6526, 2013.

[9] M. Nakagami, "The m-distribution – A general formula of intensity distribution of rapid fading," in *Statistical Methods in Radio Wave Propagation*, W. C. Hoffman (ed.), pp. 3–36, Pergamon Press, London, 1960.

[10] M. A. Juang and M. B. Pursley, "Finite-state Markov chain models for the intensity of Nakagami fading," *Int. J. Wireless Inf. Networks*, vol. 20, no. 2, pp. 95–102, 2013.

[11] Advanced Hardware Architectures, Inc., Product Specification for AHA4501Astro 36 Mbits/sec Turbo Product Code Encoder/Decoder. Available: http://www.aha.com.