

Finite Block Length Coding for Low-latency High-Reliability Wireless Communication

Leah Dickstein*, Vasuki Narasimha Swamy*, Gireeja Ranade†, Anant Sahai*

*University of California, Berkeley, CA, USA

†Microsoft Research, Redmond, WA, USA

Abstract—This paper takes a step towards making practical cooperative protocols for wireless low-latency high-reliability communication. We consider the effect of finite block-length error correction codes and the main message is that the demands on the error-correcting code are different in different phases of a diversity-seeking cooperative protocol: In the first hop where messages must reach potential relays, the code only has to achieve a moderate probability of error. The final hop from relays to the destination is where the code must be ultrareliable. The results are illustrated in the context of a simple concatenated Hamming+Reed Solomon code.

I. INTRODUCTION

In [1], we introduced a wireless protocol framework that harvests diversity needed for high reliability by employing cooperative communication. Cooperating nodes relay messages simultaneously using a space-time code so that spatial diversity is harvested while meeting low latency requirements. In [2], this protocol framework was extended to include network coding that exploits the bi-directional traffic (both uplink and downlink) and channel reciprocity. Both of those works ignored the effects of realistic error-correcting codes — instead assuming the existence of perfect capacity-achieving codes.

In this paper, we analyze the effect of finite-block-length codes on such cooperative communication schemes. Our naive guess would be that we simply must add the error-correcting code’s gap-to-capacity at the desired probability of error to the transmit SNR predicted by using a perfect capacity-achieving code. The main finding of this paper is that this naive guess is too conservative. We can do a few dB better.

The main insight is that the demands on the error-correcting code are different in different phases. In the initial phases of the cooperative protocols proposed in [1], [2], the key is to recruit as many relays as possible and for this, the error-correcting code does not have to be ultrareliable. Moderate reliability is fine. However, when the messages are finally delivered to their ultimate destination, there is no diversity with respect to the additive noise and it is vital that the error-correcting code be ultrareliable. Because multiple relays were very likely to have been recruited earlier in the protocol, there is less of a fear of simultaneous deep fades.

The main challenge we encounter while analyzing such protocols is the computational complexity of actually evaluating what SNRs would suffice. Since we want very high

reliability (10^{-9}), simulations would take far too long. Numerical integration could be used, but there is a curse of dimensionality since the number of independent fades grows quadratically with the number of nodes in the network. In Section III, this paper presents a simpler way of analyzing the impact of finite-block-length codes in cooperative communication before showing numerical results in Section IV.

II. BACKGROUND

A useful comm-theoretic perspective is to decompose the required SNR into three parts: (1) capacity: how much does the rate fundamentally require? (2) gap-to-capacity: given the target reliability and the specific code being used, how many extra dB do we need beyond capacity? (3) fading-margin: how many dB do we need to absorb bad wireless fades?

Although it is useful to be able to think about these separately, they clearly interact with each other at the system level. For example, if overall “goodput” is what is desired and the higher layers will use ARQ to achieve high reliability, then lowering the target reliability on a link comes at the cost of more retransmissions and hence less overall rate. In [3] the authors propose that links which fail about 10% of the time (allowing more aggressive code rates) result in the best goodput. A similar finding is reported in [4] from a channel dispersion perspective. The core question in this paper is whether a similar story holds when we have a diversity-oriented cooperative communication protocol with a low latency requirement. In this section, we briefly provide pointers into the relevant background underlying diversity-oriented cooperative communication, finite block length effects on performance of error correction codes and the specific low-latency oriented protocols that we are considering.

A. Cooperative Communication

As discussed in [1], exploiting virtual multiple antennas are at the heart of cooperative communication [5]–[7]. The schemes used for cooperation can be broadly divided into: a) coded-cooperation; b) distributed space-time coding; and c) delay-diversity approaches. In coded-cooperation relays decode the message, then re-encode it and transmit a new helper code word [8]–[14]. Laneman et al. [5] proposed a simple distributed space-time block coding (DSTBC) scheme in which each relay transmits a different column of the space-time block code (STBC) matrix. A randomized strategy was addressed in [15] where each relay node transmits an

independent random linear combination of the codewords that would have been transmitted by all the elements of a multi-antenna system thus eliminating the need for a centralized code (or antenna) allocation procedure. Meanwhile, works like [16], [17] employ a flooding strategy where relays repeat the signal effectively acting as active scatterers — effectively increasing the number of taps in the channel and creating guaranteed delay-diversity. All these strategies rely very crucially on the design of good error correction codes.

B. Finite-Block-Length Coding

Most of the theoretical results about cooperative communication assume infinite blocklength codes. However finite block-lengths matter [4], [18]. In the wireless context, the impact on the diversity multiplexing tradeoff was studied in [19], [20], the effect of outdated CSI was studied in [21], the effects of queue constrains were studied in [22], and effect of coherence time on diversity was studied in [23]. There are other notable works which have focussed on interesting aspects of finite block length coding [24]–[29]. The recent paper [30] looked at a very similar problem involving the effects of finite-block-length codes as well as CSI estimation-overhead in the context of low-latency applications and presented simulation results where the finding was that the average PER increases in the number of participating terminals unless the terminals also act as potential relays.

C. Our Low-Latency Protocols: Occupy CoW and XOR-CoW

As in [1], [2], the context in which our protocols exist was adapted from the actual communications requirements for a control system for an industrial printer that uses a wired network to achieve the required performance [31]. We consider a network with a central controller (C) that wishes to send and receive separate messages to and from each node in a set of n nodes, denoted by the set \mathcal{N} (see Fig. 1). Distinct messages (for our plots here each of size $m = 160$ bits, i.e. 20B) flow in a star topology from the central controller to individual nodes, and in the reverse direction from the nodes to the controller within a “cycle” of length T (here $T_c = 2$ ms). This cycle of communication must be achieved with a very small outage probability P_e (on the order of 10^{-9}). To imagine a wireless drop-in protocol, we set the available bandwidth to be 20MHz (the size of a standard WiFi channel) for our plots. We assume that all channels are independently Rayleigh faded.

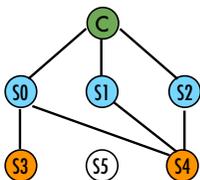


Fig. 1: Topology schematic for the protocols.

1) *Occupy CoW*: The Occupy CoW protocol (which we abbreviate as the CoW-protocol) uses multi-user diversity to overcome bad fading events. The basic idea is to use a

flooding strategy where the controller broadcasts a packet that includes messages for all nodes. As this is a broadcast message, the intended set of recipients are all nodes. Nodes with good channels act as relays for other nodes. Details of the protocol are discussed in [1]. In this work we consider a fixed schedule 2-hop variant of the protocol. The path to any node from the controller is restricted to 2-hops and all nodes get two shots at succeeding irrespective of whether they’ve already succeeded in the first trial. The time available T_c is divided for 4 phases namely downlink phase 1 (T_{D_1}), uplink phase 1 (T_{U_1}), downlink phase 2 (T_{D_2}) and uplink phase 2 (T_{U_2}) such that $T_{D_1} + T_{U_1} + T_{D_2} + T_{U_2} = T_c$. In downlink phase 1, the controller puts the messages it has for each node into a single packet and broadcasts it. This is followed by uplink phase 1 where the nodes transmit their uplink information in a time divided fashion such that each node gets T_{U_1}/n time to transmit its message. This is followed by downlink phase 2 where all nodes that have successfully decoded the controller’s message simultaneously broadcast the entire message using a space-time code. Uplink phase 2 is similar except that the individual messages are being sent to the controller and the time slot for each node’s message/repetition is T_{U_2}/n long.

2) *XOR-CoW*: The XOR-CoW protocol [2] (which we abbreviate as the XOR-protocol) follows the same key ideas as the CoW-protocol described above, except that it also uses network coding inspired by works such as [32], [33]. The protocol increases efficiency by dividing the cycle length into only three phases. The first downlink phase and the second uplink phase follow the same broadcast ideas as the previous protocol. In the third “XOR” phase, strong nodes that have *both* uplink and downlink messages for other nodes broadcast the XOR of the two messages associated with each individual node being helped, thus simultaneously serving as an uplink relay for the controller and as a downlink relay for that node.

In both the XOR-protocol and the CoW-protocol considered here, every message effectively gets exactly two chances to be heard correctly. The first time it is transmitted and then again when relays simultaneously try to help it get to its destination. The core question here is how do the realities of error-correcting codes change what needs to happen in the protocol and whether there is any difference in how we use the error correcting code in the first vs the second transmission.

III. CoW WITH FINITE BLOCK LENGTH CODING

A. Idealized analysis of the 2-hop downlink CoW-protocol

Before considering the situation with finite block-length codes, we first review how the probability of success for downlink is derived in [1], [2]. The uplink derivations are similar, albeit a bit more complicated. A note about the notation: we use calligraphic script to denote sets; the random variable associated with the size of a set is in upper case and the instantiation being considered is denoted by lower case.

Denote the set of nodes with direct controller links by \mathcal{A} . Other nodes may connect to the controller through these nodes in a two-hop fashion. At its essence, our simple

analysis fundamentally examines how the size A of this set \mathcal{A} changes what happens in the second phase. Here, the transmission rate in the downlink phases is $R = \frac{mn}{T/4}$ and hence the probability of a single link outage due to fading (assuming Rayleigh fading and a Shannon capacity-achieving code) is $p = 1 - \exp(-\frac{2^R-1}{SNR})$. Then A follows a Binomial distribution.

Under the ideal conditions that the channels do not change during a cycle & are reciprocal, the probability of cycle failure is the probability that at least one of the nodes in the set $\mathcal{N} \setminus \mathcal{A}$ does not connect to \mathcal{A} . Essentially hearing the loudest relay is enough as it is not necessary to hear all of them. Hence, we call this model ‘the loudest talker’ model. Thus we have:

$$P(\text{fail}|A = a) = 1 - (1 - p^a)^{n-a}$$

Thus, the probability of cycle failure is given by:

$$\begin{aligned} P(\text{fail}) &= \sum_{a=0}^{n-1} P(A = a) \cdot P(\text{fail}|A = a) \\ &= \sum_{a=0}^{n-1} \left(\binom{n}{a} (1-p)^a p^{n-a} \right) (1 - (1 - p^a)^{n-a}) \end{aligned} \quad (1)$$

(2)

B. Effect of additive noise at receivers

Practical receivers introduce some sort of additive noise to the signal. To guard against this, we use error correction codes. The descriptions of the noise models, modulation schemes, etc. can be found in [34]. For this paper, the main quantity of interest is the probability of incorrect decoding denoted by $F_S(r_P)$ where \mathcal{S} is the coding scheme under consideration (including the block length) and r_P is the power received at the receiver.

C. Loudest talker analysis

The challenge here is a curse of dimensionality — we want to be able to say something interesting when there are tens of nodes in the system. We call our approach the ‘loudest talker model’ and analyze downlink, uplink and XOR-CoW protocols using this approach.

1) *Downlink*: Let the coding scheme used be \mathcal{S} and the rate of coding is given by $R = \frac{mn}{T/4}$ where m is the message size and n is the number of nodes in the network. If the instantaneous fade from the controller to a node i was h_i , then the probability of declaring a decoding error is given by

$$P(\text{error}|h_i) = F_S(|h_i|^2 SNR). \quad (3)$$

The fade h_i is Rayleigh faded so the probability of declaring a decoding error (or the probability of a link failing in downlink phase 1) is given by

$$P(\text{single}) = \int_0^\infty P(\text{error}|h_i) f(|h_i|^2) d(h_i^2) \quad (4)$$

where $f(|h_i|^2)$ is the pdf of an exponential random variable. Let the nodes which succeeded in downlink phase 1 be called \mathcal{A} (with cardinality A). Then probability that $A = a$ is a

binomial with probability of failure given by Eq. (4) as seen earlier. We now consider a node (say j) that hasn’t heard its downlink message from the controller. In downlink phase 2, the relays \mathcal{A} will simultaneously broadcast the packet and the node j can only reap the benefits of the loudest link. Let the random variable associated with the channel fade of the loudest link from \mathcal{A} to node Y be denoted by $H_a^{max} = \max(|h_{1,j}|^2, |h_{2,j}|^2, \dots, |h_{a,j}|^2)$. The pdf of H_a^{max} (denoted by $f_a^{max}(h^2)$) is given by

$$f_a^{max}(h^2) = a(1 - \exp(-h^2))^{a-1} \exp(-h^2)$$

Now the probability of declaring a decoding error at node j in downlink phase 2 given all the instantaneous fades between the node j and the relay nodes \mathcal{A} is

$$P(\text{error}|h_{1,j}, \dots, h_{a,j}) = F_S \left((h_a^{max}(j))^2 SNR \right) \quad (5)$$

where $h_a^{max}(j) = \max(|h_{1,j}|, \dots, |h_{a,j}|)$. Thus the probability of declaring a decoding error in downlink phase 2 is given by

$$P(\text{loudest}|A = a) = \int_0^\infty P(\text{error}|h_{1,j}, \dots, h_{a,j}) f_a^{max}(h^2) dh^2 \quad (6)$$

Thus, the probability of failure for 2-hop downlink under the loudest talker model is given by

$$\begin{aligned} P(\text{fail}) &= \sum_{a=0}^n P(A = a) (1 - (1 - P(\text{ec}|A = a))^{n-a}) \\ &= \sum_{a=0}^n \left\{ \binom{n}{a} (1 - P(\text{single}))^a P(\text{single})^{n-a} \right\} \\ &\quad \times \{1 - (1 - P(\text{loudest}|A = a))^{n-a}\} \end{aligned} \quad (7)$$

This style of analysis yields an exact and tractable calculation for downlink reliability. But it doesn’t give insight into what the dominant effects are and where coding reliability is required.

We address this by approximating the waterfall curve of an error-correcting code with a threshold cliff. We consider that a link is ‘bad’ in two ways a) if the received power is too low due to fading; or b) the additive noise at the receiver was too much despite good enough receive power. Let the transmit power (in dB) be t_P and the threshold to declare inadequate receive power (in dB) be r_{th} . The probability that actual received power r_P is less than r_{th} is the probability of bad fade (denoted by p_{fade}). Thus we have,

$$p_{fade} = P(r_P < r_{th}) = 1 - \exp(-10 \frac{r_{th} - t_P}{10}) \quad (8)$$

The probability that additive noise is too high is approximated by the probability of decoding error when the received power is r_{th} . Thus we get the probability of high additive noise (denoted by p_{add}) under coding scheme \mathcal{S} is given by,

$$p_{add} = F_S(r_{th}) \quad (9)$$

This threshold of reliability (p_{add}) that divides acceptable from unacceptable is an internal parameter of the analysis

that can be optimized to get the best overall bound. Thus, the probability of failure of a link is given by

$$p_{link} = p_{fade} + (1 - p_{fade})p_{add}. \quad (10)$$

The combination of looking at the loudest talker (max SNR) and approximating the waterfall curve with a threshold cliff enables the analysis to decompose and become scalable with the number of nodes. We get nested sums, but the number of nested sums scales with the number of phases of the protocol rather than the number of nodes. Downlink analysis now simplifies to:

$$P(A = a) = \binom{n}{a} (1 - p_{link})^a (p_{link})^{n-a} \quad (11)$$

Conditioned on the cardinality of \mathcal{A} , the probability of downlink failure is then the probability that at least one of the remaining $n - a$ nodes did not hear the message. Thus we get,

$$P(\text{fail}|A = a) = 1 - (1 - p_{fade}^a - (1 - p_{fade}^a) p_{add})^{n-a} \quad (12)$$

Combining Eq. (11) and (12) we get,

$$P(\text{fail}) = \sum_{a=0}^n P(A = a) P(\text{fail}|A = a) \quad (13)$$

Uplink proceeds similarly, but for reasons of space, we omit that analysis.

D. XOR-CoW Loudest Talker Analysis

We must analyze the XOR-CoW protocol slightly differently as the successes in downlink and uplink are coupled. This coupling makes it impossible to decouple all the integrals representing each of the independent fades — leaving us with a curse of dimensionality for numerical integration, which must be in turn be done to high precision to resolve probabilities of error around 10^{-9} . Fortunately, the bounding approach taken above can be made tractable.

We assume that the transmit power is the same at all nodes and in all phases. However, we allow the receive power threshold (implying a different fade tolerance in different phases) to be different for the downlink-uplink and XOR phases. For simplicity, we set the thresholds to be the same for downlink and uplink but demand a higher receive power for the XOR phase. Essentially, a link with a good fade in downlink & uplink phases does not imply that the link has sufficient capacity for the XOR phase. The reason for setting different thresholds is to capture the importance of the relaying phase. If a message did not succeed in the first trial (downlink or uplink phase), then it has only one more chance to succeed. By requiring the receive power to be higher, we essentially try to combat the effect of additive noise at the receiver. This makes sense because the maximum energy in the loudest talker is indeed higher than it would be for a single talker. The current analysis is for a set of threshold and we search over the threshold values that minimizes the transmit power required. The rates in the different phases are determined by the block lengths allocated for the phases (T_D , T_U and T_X), the number of nodes in the network (n) and

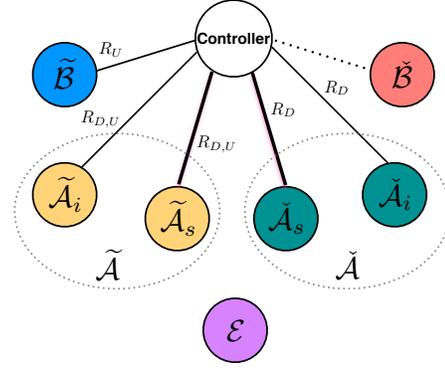


Fig. 2: The figure shows the different sets and their connectivity to the controller. The interconnection between the sets needed for success are not shown. The rates annotating the links are the rates in which the links to the controller are present. The bold links belong to the superior set which has links to the controller during the XOR phase.

the payload sizes (m). Thus we get that the downlink, uplink and XOR rates are $R_D = \frac{m \cdot n}{T_D}$, $R_U = \frac{m \cdot n}{T_U}$ and $R_X = \frac{m \cdot n}{T_X}$ respectively.

Let the transmit power be t_P (in dB) and the received power threshold for downlink and uplink phases be r_{DU} and r_X (in dB). The probability of having a good fade in the downlink and uplink phases is then given by

$$p_{fade_{DU}} = 1 - \exp(-10^{\frac{r_{DU} - t_P}{10}}). \quad (14)$$

Let p_{add_D} and p_{add_U} be the probability of failure due to additive noise (despite having had enough receive power) in the downlink and uplink phase respectively. The probabilities are different because the blocklengths of the messages in these phases are different. We partition the set of nodes \mathcal{N} (as shown in Fig. 2) into different sets for ease of analysis:

- Let the set of nodes which have a good fade to the controller in the downlink and uplink phase be \mathcal{G} . This set is further divided into disjoint sets $\tilde{\mathcal{A}}$, $\tilde{\mathcal{A}}$, $\tilde{\mathcal{B}}$ and $\tilde{\mathcal{B}}$ such that $\mathcal{G} = \tilde{\mathcal{A}} \cup \tilde{\mathcal{A}} \cup \tilde{\mathcal{B}} \cup \tilde{\mathcal{B}}$.
- $\tilde{\mathcal{A}}$ is the set of nodes which were successful in both downlink and uplink phases.
- $\tilde{\mathcal{A}}$ is the set of nodes which were successful in downlink phase only (no uplink).
- $\tilde{\mathcal{B}}$ is the set of nodes which were successful in uplink only (no downlink).
- $\tilde{\mathcal{B}}$ is the set of nodes which were not successful in neither downlink nor uplink.

In order to act as a relay in the XOR phase, a node *must* have the downlink information. Hence only nodes in $\tilde{\mathcal{A}} \cup \tilde{\mathcal{A}}$ can act as relays in the XOR phase. As we have further restricted the receive power needed to overcome the additive noise threshold in the XOR phase, only a subset of the nodes in $\tilde{\mathcal{A}} \cup \tilde{\mathcal{A}}$ can help. Let the subset of the nodes in $\tilde{\mathcal{A}}$ with “superior” links to the controller be $\tilde{\mathcal{A}}_s$ (the rest form $\tilde{\mathcal{A}}_i$) and the subset of nodes in $\tilde{\mathcal{A}}$ with “superior” links to the controller be $\tilde{\mathcal{A}}_s$ (the rest form $\tilde{\mathcal{A}}_i$).

We enumerate the ways in which nodes can succeed.

- Nodes in $\tilde{\mathcal{A}}$ successfully receive their downlink information in the downlink phase and successfully transmit their uplink information in the uplink phase.
- Nodes in $\tilde{\mathcal{A}}$ successfully receive their downlink information in the downlink phase. Nodes in $\tilde{\mathcal{A}}_s$ successfully transmit their uplink information if the additive noise wasn't too much at the controller during their slot in the XOR phase. Nodes in $\tilde{\mathcal{A}}_i$ don't have a superior link to the controller. They can successfully transmit their uplink information to the controller only if a node in the set $\mathcal{A}_s = \tilde{\mathcal{A}}_s \cup \tilde{\mathcal{A}}_s$ successfully heard its uplink message *and* the additive noise at the controller during its slot in the XOR phase wasn't too much.
- Nodes in $\tilde{\mathcal{B}}$ successfully transmit their uplink information in the uplink phase. They can successfully receive their downlink information either directly from the controller if the controller has a superior link to the node or if they connect to $\mathcal{A} = \tilde{\mathcal{A}} \cup \tilde{\mathcal{A}}$ in both uplink and XOR phase (thus having a superior link).
- Nodes in $\tilde{\mathcal{B}} \cup \{\mathcal{N} \cap \mathcal{G}\}$ succeed by connecting to \mathcal{A}_s in the uplink phase (to have a path to the controller in the XOR phase). They succeed in getting their downlink information by either having a superior link to \mathcal{A}_s in the XOR phase or by connecting to $\mathcal{A}_i = \tilde{\mathcal{A}}_i \cup \tilde{\mathcal{A}}_i$ in uplink as well as XOR phase. Additionally, the additive noise at both the controller and the node must be low enough in the XOR phase.

Notation:

In order to effectively present the derived expressions, we provide a guide to the notation that will be used in the following sections. A binomial distribution with n independent experiments, probability of success $1 - p$, and number of success m will be referred to as

$$B(n, m, p) = \binom{n}{m} (1 - p)^m p^{n-m}. \quad (15)$$

Failure is the event that even one of the nodes did not get its downlink information or wasn't able to transmit its uplink information. We will calculate the probability of failure by unraveling the state space. As mentioned earlier, the probability of having a bad fade in the downlink and uplink phases is then given by Eq. (14)

$$pfade_{DU} = 1 - \exp(-10^{\frac{r_{DU}-t_P}{10}})$$

where r_{DU} is the receive power threshold and t_P is the transmit power.

Therefore the probability of $G = g$ nodes having a good link to the controller is given by $P(G = g) = B(n, g, pfade_{DU})$. Conditioned on the event of having $G = g$ good fade nodes, let us look at the distribution of different sets $\tilde{\mathcal{A}}, \tilde{\mathcal{A}}, \tilde{\mathcal{B}}$ and $\tilde{\mathcal{B}}$.

We denote by $\mathcal{A} = \tilde{\mathcal{A}} \cup \tilde{\mathcal{A}}$ the set of nodes that succeed in downlink. Thus, we get that in addition to having good links, the additive noise at these receivers were low enough to allow decoding. The probability of failing due to additive noise despite having enough receive power in the downlink

phase is $pfade_D$ which depends on the block length and the coding rate as already discussed earlier. Thus, we get that the probability that $A = a$ conditioned on $G = g$ is given by $P(A = a|G = g) = B(g, a, pfade_D)$.

Out of the nodes in the set \mathcal{A} , only the set $\tilde{\mathcal{A}}$ succeed in uplink as well. The probability of having low enough additive noise to enable decoding in the uplink phase is given by $pfade_U$. Thus conditioned on $G = g$ and $A = a$, we get that the probability of $\tilde{\mathcal{A}} = \tilde{a}$ is given by $P(\tilde{\mathcal{A}} = \tilde{a}|A = a, G = g) = B(a, \tilde{a}, pfade_U)$. In addition to $\tilde{\mathcal{A}}$, the nodes in $\tilde{\mathcal{B}}$ also succeed in the uplink phase (though they did not succeed in the downlink phase). Conditioned on $G = g$, $A = a$ and $\tilde{\mathcal{A}} = \tilde{a}$, we get the probability of $\tilde{\mathcal{B}} = \tilde{b}$ is given by $P(\tilde{\mathcal{B}} = \tilde{b}|G = g, A = a, \tilde{\mathcal{A}} = \tilde{a}) = B(g - a, \tilde{b}, pfade_U)$.

We'll now calculate the probability of the 'superior' sets $\tilde{\mathcal{A}}_s$ and $\tilde{\mathcal{A}}_s$. We already know that the fades between the nodes in the set \mathcal{A} and the controller has a minimum receiver power of r_{DU} . In the XOR phase, the receiver power required is $r_X \geq r_{DU}$. Conditioned on the links being good enough for the downlink and uplink phases, the probability that they are *not good* enough for the XOR phase is given by

$$p_{XDU} = P(\text{link not good for XOR} | \text{link good for DU}) \quad (16)$$

$$= 1 - \exp(10^{\frac{r_{DU}-t_P}{10}} - 10^{\frac{r_X-t_P}{10}})$$

Therefore, we get the probability of $\tilde{\mathcal{A}}_s = \tilde{a}_s$ and $\tilde{\mathcal{A}}_s = \tilde{a}_s$ conditioned on $\tilde{\mathcal{A}} = \tilde{a}$ and $\tilde{\mathcal{A}} = \tilde{a}$ is given by $P(\tilde{\mathcal{A}}_s = \tilde{a}_s, \tilde{\mathcal{A}}_s = \tilde{a}_s | \tilde{\mathcal{A}} = \tilde{a}, \tilde{\mathcal{A}} = \tilde{a}) = B(\tilde{a}, \tilde{a}_s, p_{XDU}) \cdot B(\tilde{a}, \tilde{a}_s, p_{XDU})$.

We now calculate the probability of success of each set in the XOR phase. Set $\tilde{\mathcal{A}}$ has already succeeded in the downlink and uplink phases, so their probability of success is 1. Therefore

$$P(\text{success of } \tilde{\mathcal{A}}) = 1.$$

The next set under consideration is $\tilde{\mathcal{A}}_s$ which succeeds as long as the additive noise at the controller was low enough in the XOR phase (this happens with probability $pfade_X$ which depends on the block length, coding rate and r_X). Therefore

$$P(\text{success of } \tilde{\mathcal{A}}_s) = (1 - pfade_X)^{\tilde{a}_s}.$$

The next set under consideration is $\tilde{\mathcal{A}}_i$ which succeeds if the nodes have a connection to \mathcal{A}_s in the uplink phase and the additive noise at the controller was low enough in the XOR phase. Therefore

$$P(\text{success of } \tilde{\mathcal{A}}_i) = ((1 - (p_U)^{a_s})(1 - pfade_X))^{\tilde{a}_i}$$

where $p_U = pfade_{DU} + (1 - pfade_{DU})pfade_U$.

Consider the set $\tilde{\mathcal{B}}$ which succeeds if they have a 'superior' link to the controller (with probability $1 - p_{XDU}$) or they connect to \mathcal{A} in uplink phase and have a superior link to the set in the XOR phase. Let the probability of success for a node (before considering the effect of thermal noise at the receiver) in $\tilde{\mathcal{B}}$ be $q_{\tilde{\mathcal{B}}}$. Then we have

$$q_{\tilde{\mathcal{B}}} = \left(1 - p_{XDU} + p_{XDU} \left(\sum_{k=1}^a (1 - p_U)^k p_U^{a-k} (1 - p_{XDU}^k) \right) \right)$$

where $p_U = pfade_{DU} + (1 - pfade_{DU})padd_U$. Thus we have

$$P(\text{success of } \tilde{B}) = (q_{\tilde{B}}(1 - p_{XDU}))^{\tilde{b}}.$$

Lets consider the nodes in $\mathcal{N} \setminus \mathcal{G}$ and \tilde{B} . They succeed in transmitting their uplink information by connecting to \mathcal{A}_s in the uplink phase (to have a path to the controller in the XOR phase). They succeed in getting their downlink information by either having a superior link to \mathcal{A}_s in the XOR phase or by connecting to $\mathcal{A}_i = \tilde{\mathcal{A}}_i \cup \tilde{\mathcal{A}}_i$ in uplink as well as XOR phase. We calculate the probability of not getting a path to success f_e (not counting thermal noise).

$$f_e = p_U^{a_s} + \left\{ \left[\sum_{k_s=1}^{a_s} \left(B(a_s, k_s, p_U) p_{XDU}^{k_s} \right) \right] \times \left[\sum_{k_i=0}^{a_i} \left(B(a_i, k_i, p_U) p_{XDU}^{k_i} \right) \right] \right\} \quad (17)$$

where $a_s = \tilde{a}_s + \tilde{a}_s$ and $a_i = \tilde{a}_i + \tilde{a}_i$. Thus we get

$$P(\text{fail of node in else}) = f_e + (1 - f_e) (1 - (1 - padd_X)^2).$$

$$P(\text{success of else}) = (1 - P(\text{fail of node in else}))^{n-g+\tilde{b}}.$$

Combining the success equations above we get,

$$P(\text{success}|\text{states}) = P(\text{success of else}) \cdot P(\text{success of } \tilde{B}) \cdot P(\text{success of } \tilde{\mathcal{A}}_i) P(\text{success of } \tilde{\mathcal{A}}_s).$$

Finally,

$$P(\text{failure}) = \sum_{\text{states}} P(\text{states}) \times P(\text{success}|\text{states}).$$

where

$$\begin{aligned} P(\text{states}) &= P(\tilde{\mathcal{A}}_s = \tilde{a}_s, \tilde{\mathcal{A}}_s = \tilde{a}_s | \tilde{\mathcal{A}} = \tilde{a}, \tilde{\mathcal{A}} = \tilde{a}) \\ &\cdot P(\tilde{B} = \tilde{b} | G = g, A = a) \cdot P(\tilde{\mathcal{A}} = \tilde{a} | A = a) \quad (19) \\ &\cdot P(A = a | G = g) \cdot P(G = g) \end{aligned}$$

IV. NUMERICAL RESULTS

In this section we present numeric results so that the relative quality of the bounds can be seen. The individual node message payload size used for all these plots is 20B, the latency requirement is 1.5ms and the available bandwidth is 20MHz. The total blocklength given for each phase is thus 10000 symbols. Before presenting the results, we briefly discuss the very simple coding scheme that we consider in this paper to showcase finite blocklength effects.

Concatenated Hamming+Reed-Solomon code

A short Hamming code is used to fix isolated bit flips with a Reed-Solomon code wrapper to clean up the rest. In particular, we consider a (7, 4) code, and each of the $16 = 2^4$ Hamming codewords forms a symbol in the Reed-Solomon alphabet. When we need a field size of more than 16, we just group two Hamming codewords together and so up to 256 RS symbols can be obtained by putting two together, and so on. We then generate RS parity symbols such that the coding rate is close to $R = \frac{m \times n}{T/4}$ for any given m, n and

T . The exact expressions for decoding error $F_S(SNR)$ as a function of the SNR at the receiver can be computed and we have used the half-minimum-distance decoding expressions for an underlying BPSK signaling assumption for our plots.

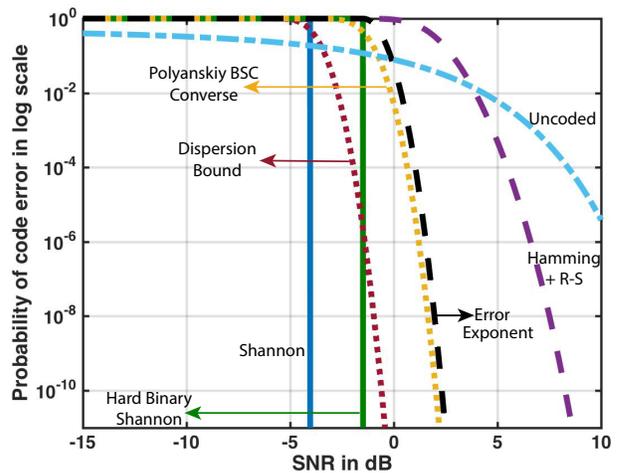


Fig. 3: Waterfall curves with a block-length of 333 symbols per codeword at a coding rate of $R = 0.48$.

We begin by looking at the waterfall curves (in Fig. 3) for various coding techniques for $n = 30$ nodes which corresponds to a coding rate of $R = \frac{30 \times 160}{10000} = 0.48$ at an 'uplink' blocklength of $10000/30 = 333$ symbols per codeword. As expected, the performance for the simple concatenated Hamming+Reed-Solomon code is much worse than the channel dispersion-based bound [4]. We observe that once in the waterfall region, the block error probability for the Hamming+Reed-Solomon scheme falls rapidly from 10^{-2} to 10^{-10} in a matter of 4dB.

We first look solely at the performance of the downlink 2-hop protocol with uplink-like blocklengths. The reason for considering this particular scenario is because downlink is the simplest to analyze because of the independence of the links used in various phases. This allows us to get a better understanding of various issues that need to be considered due to ultra-reliability requirement. The uplink blocklengths are used because they are shorter and hence more vulnerable to additive noise. In Fig. 4 we consider the following curves: a) the Shannon code curve which gives us the lower bound on the power required; b) the AWGN-dispersion curve derived using the integral model as described in Eq. (7) in Sec.III; c) the dispersion code curve for the fade + additive noise model from Eq. (10) with p_{add} being set to 10^{-10} ; d) the concatenated Hamming+Reed-Solomon code curve using the integral model as described in Eq. (7); and e) the concatenated Hamming+Reed-Solomon code curve for the fade + additive noise model from Eq. (10) with p_{add} being set to 10^{-10} . We heuristically set p_{add} to 10^{-10} since the target probability of cycle failure is 10^{-9} .

The underlying reason for the gaps in Fig. 4 between the downlink curve calculated with the full integral (as described in Eq. (4)) and the one calculated using the $p_{add} = 10^{-10}$

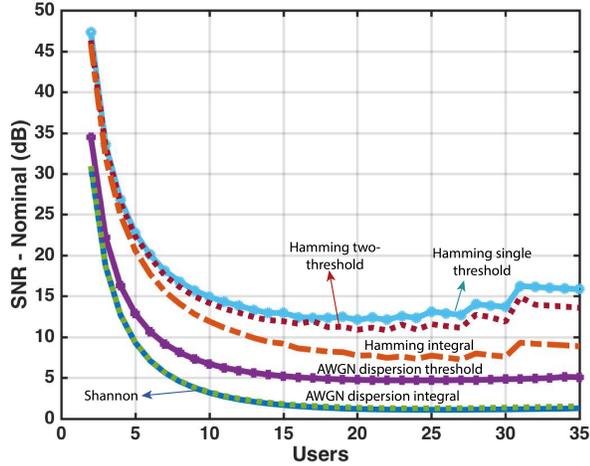


Fig. 4: For 2-hop downlink, the SNR required for a perfect Shannon capacity code versus the SNR required for using various coding schemes is shown. The reliability is 10^{-9} .

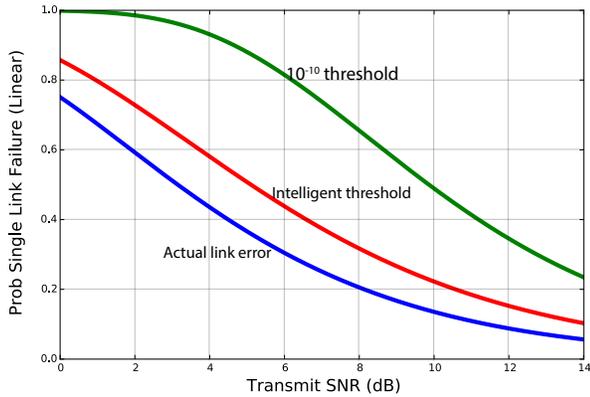


Fig. 5: Comparison between bounds for a single link failure using different models for coding rate corresponding to $R = 0.48$ and blocklength of $1000/3$.

threshold bound (described in Eq. (10)) can be seen by examining Fig. 5. At any transmit SNR, the thresholding bound significantly overestimates the probability of failure which translates to an increase in transmit power needed to achieve the same performance. A third curve is shown that uses an intelligent search over the value of p_{add} to get as close to the actual value of p_{link} as possible.

Fig. 6 shows the transmit SNR required to achieve our target reliability while using the XOR-CoW protocol. As explained earlier, the analysis of the XOR-CoW protocol using the integral approach is computationally intractable and hence not plotted. However we do plot curves with a) p_{add} set to 10^{-10} ; and b) where we search over p_{add} for each phase. The line corresponding to the Shannon capacity code gives us a lower bound on the transmit power required. The dispersion-based line corresponding to the adaptive search over p_{add} gives us a good ballpark lowerbound on how a ‘good’ finite blocklength code can seem to perform using this style of analysis. The performance of the concatenated

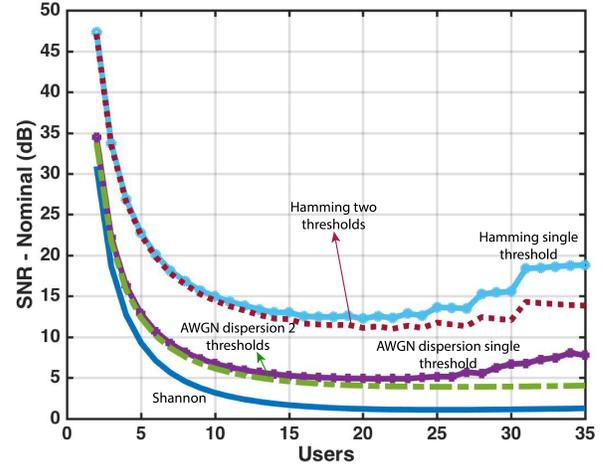


Fig. 6: The SNR required under the assumption of existence of a Shannon capacity code versus the SNR required for using a practical code for XOR-CoW protocol. The target reliability is 10^{-9} .

Hamming+Reed-Solomon code is similar to that in Fig. 4. The close match suggests to us that the downlink integral curves are indeed essentially the right answers even for the XOR case.

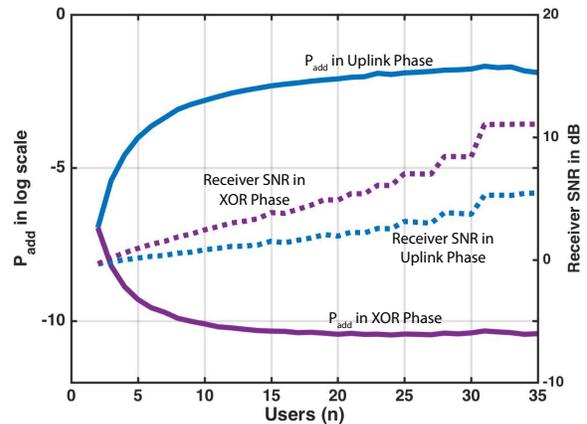


Fig. 7: Optimized Receiver SNR thresholds (and the corresponding additive noise error probability) for the XOR-CoW protocol using the simple Hamming+RS Code.

The more interesting aspect is to look in Fig. 7 at the receive SNR thresholds that are selected for the XOR phase vs the downlink/uplink phase when we allow those thresholds to be optimized. By tolerating a lower receiver power for the downlink-uplink phases, we allow for a potentially larger number of relays for the relaying phase even as the resulting probability of additive noise induced error is higher. More relays means that we can more easily count on getting higher receiver power in the XOR phase – thus getting higher reliability in the relaying phase. This allows for lowering the transmit power required by around 4dB which agrees

with the number from the waterfall curve at 30 nodes. This possibility of “partial credit” is why the naive prediction of simply adding the capacity-gap to the Shannon-style analysis is too conservative. The error-correcting code in the early phases is not called upon to hit probabilities of error of 10^{-9} . That is only required at the final phase.

V. CONCLUSIONS AND FUTURE WORK

Works like [4] and [18] tell us that once blocklengths are short, no code can be perfect. For low-latency communication, short blocklengths are essential. For ultrareliability, multiple phases and the prospect of relaying is essential to harvest the required diversity of fading. The message of this paper is that in the initial phase, the goal is not ultrareliability but reaching the maximum number of relays. The code can therefore be run at a much more moderate error probability — similar to traditional wireless communication systems that will use ARQs to achieve reliability. The relaying phase (in this case the XOR phase) must be made as reliable as possible as it is the last chance to succeed. This means that simply adding together the gap-to-capacity to the Shannon bound is too conservative when thinking about ultrareliable low-latency wireless communication. We can do significantly better.

ACKNOWLEDGEMENTS

This work grew out of an ongoing project with Borivoje Nikolić, Venkat Anantharam, Adam Woliz, and Paul Rigge who we thank for many discussions leading up to the investigations reported in this paper. We thank the NSF for grants CNS-0932410, CNS-1321155, and ECCS-1343398.

REFERENCES

- [1] V. Narasimha Swamy *et al.*, “Cooperative communication for high-reliability low-latency wireless control,” in *Communications (ICC), 2015 IEEE International Conference on*, June 2015, pp. 4380–4386.
- [2] V. Narasimha Swamy *et al.*, “Network coding for High-Reliability Low-Latency wireless control,” in *2016 IEEE Wireless Communications and Networking Conference WS 11 : IEEE WCNC’2016 Workshop on The Tactile Internet: Enabling Technologies and Applications (IEEEWCNC2016-TACNET)*, Doha, Qatar, Apr. 2016.
- [3] P. Wu *et al.*, “Coding versus ARQ in fading channels: how reliable should the PHY be?” *Communications, IEEE Transactions on*, vol. 59, no. 12, pp. 3363–3374, 2011.
- [4] Y. Polyanskiy *et al.*, “Channel coding rate in the finite blocklength regime,” *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.
- [5] J. N. Laneman *et al.*, “Cooperative diversity in wireless networks: Efficient protocols and outage behavior,” *Information Theory, IEEE Transactions on*, vol. 50, no. 12, pp. 3062–3080, 2004.
- [6] A. Sendonaris *et al.*, “User cooperation diversity. Part I. System description,” *Communications, IEEE Transactions on*, vol. 51, no. 11, pp. 1927–1938, 2003.
- [7] —, “User cooperation diversity. part II. implementation aspects and performance analysis,” *Communications, IEEE Transactions on*, vol. 51, no. 11, pp. 1939–1948, 2003.
- [8] A. Stefanov *et al.*, “Cooperative coding for wireless networks,” *Communications, IEEE Transactions on*, vol. 52, no. 9, pp. 1470–1476, 2004.
- [9] —, “Cooperative space-time coding for wireless networks,” *Communications, IEEE Transactions on*, vol. 53, no. 11, pp. 1804–1809, 2005.
- [10] M. Janani *et al.*, “Coded cooperation in wireless communications: space-time transmission and iterative decoding,” *Signal Processing, IEEE Transactions on*, vol. 52, no. 2, pp. 362–371, 2004.
- [11] T. E. Hunter *et al.*, “Diversity through coded cooperation,” *Wireless Communications, IEEE Transactions on*, vol. 5, no. 2, pp. 283–289, 2006.
- [12] K. G. Seddik *et al.*, “Design criteria and performance analysis for distributed space-time coding,” *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 4, pp. 2280–2292, 2008.
- [13] Y. Jing *et al.*, “Distributed space-time coding in wireless relay networks,” *Wireless Communications, IEEE Transactions on*, vol. 5, no. 12, pp. 3524–3536, 2006.
- [14] B. Zhao *et al.*, “Distributed turbo coded diversity for relay channel,” *Electronics letters*, vol. 39, no. 10, pp. 786–787, 2003.
- [15] B. Sirkeci-Mergen *et al.*, “Randomized space-time coding for distributed cooperative communication,” *Signal Processing, IEEE Transactions on*, vol. 55, no. 10, pp. 5003–5017, 2007.
- [16] A. Scaglione *et al.*, “Opportunistic large arrays: Cooperative transmission in wireless multihop ad hoc networks to reach far distances,” *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2082–2092, 2003.
- [17] S. Wei *et al.*, “Asynchronous cooperative diversity,” *Wireless Communications, IEEE Transactions on*, vol. 5, no. 6, pp. 1547–1557, 2006.
- [18] Y.-W. Huang *et al.*, “Finite blocklength coding for multiple access channels,” in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 831–835.
- [19] C. P. Lee *et al.*, “Dynamic decode and forward for the multi-access relay channel with finite block length,” in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2011, pp. 1825–1829.
- [20] C. Lee *et al.*, “Diversity-multiplexing tradeoff in the multiaccess relay channel with finite block length,” *CoRR*, vol. abs/1103.3719, 2011. [Online]. Available: <http://arxiv.org/abs/1103.3719>
- [21] Y. Hu *et al.*, “The impact of outdated CSI on the finite blocklength performance of relaying,” *CoRR*, vol. abs/1606.06490, 2016. [Online]. Available: <http://arxiv.org/abs/1606.06490>
- [22] Y. Li *et al.*, “Throughput of two-hop wireless channels with queueing constraints and finite blocklength codes,” in *2016 IEEE International Symposium on Information Theory (ISIT)*, July 2016, pp. 2599–2603.
- [23] W. Yang *et al.*, “Diversity versus channel knowledge at finite blocklength,” in *Information Theory Workshop (ITW), 2012 IEEE*, Sept 2012, pp. 572–576.
- [24] Y. Hu *et al.*, “Blocklength-limited performance of relaying under quasi-static rayleigh channels,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4548–4558, July 2016.
- [25] —, “On the capacity of relaying with finite blocklength,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1790–1794, 2016.
- [26] V. Kostina *et al.*, “Fixed-length lossy compression in the finite blocklength regime,” *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3309–3338, June 2012.
- [27] G. J. Bradford *et al.*, “Rate, reliability, and delay tradeoffs for decode-and-forward relaying.”
- [28] W. Yang *et al.*, “Quasi-static multiple-antenna fading channels at finite blocklength,” *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, July 2014.
- [29] E. MolavianJazi *et al.*, “A finite-blocklength perspective on gaussian multi-access channels,” *CoRR*, vol. abs/1309.2343, 2013. [Online]. Available: <http://arxiv.org/abs/1309.2343>
- [30] Y. Hu *et al.*, “Finite blocklength performance of multi-terminal wireless industrial networks,” *CoRR*, vol. abs/1606.08646, 2016. [Online]. Available: <http://arxiv.org/abs/1606.08646>
- [31] M. Weiner *et al.*, “Design of a low-latency, high-reliability wireless communication system for control applications,” in *IEEE International Conference on Communications, ICC 2014, Sydney, Australia, June 10-14, 2014*, 2014, pp. 3829–3835.
- [32] S. Katti *et al.*, “XORs in the air: practical wireless network coding,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.
- [33] S. Bagheri *et al.*, “Randomized decode-and-forward strategies for two-way relay networks,” *Wireless Communications, IEEE Transactions on*, vol. 10, no. 12, pp. 4214–4225, 2011.
- [34] D. Tse *et al.*, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005.