Network Coding for High-Reliability Low-Latency Wireless Control

Vasuki Narasimha Swamy*, Paul Rigge*, Gireeja Ranade†, Anant Sahai*, Borivoje Nikolić*
*University of California, Berkeley, CA, USA
†Microsoft Research, Redmond, WA, USA

***Abstract*—The Internet of Things (IoT) envisions simultaneous sensing and actuation of numerous wirelessly connected devices. Emerging human-in-the-loop applications demand low-latency high-reliability communication protocols, paralleling the requirements for high-performance industrial control. This paper introduces a wireless communication protocol based on network coding that in conjunction with cooperative communication techniques builds the necessary diversity to achieve the target reliability. The proposed protocol, XOR-CoW, is analyzed by using a communication theoretic delay-limited-capacity framework and compared to different realizations of previously proposed protocols without network coding. The results show that as the network size or payload increases, XOR-CoW gains advantage in minimum SNR to achieve the target latency. For a scenario inspired by an industrial printing application with 30 nodes in the control loop, total information throughput of 4.8 Mb/s, 20MHz of bandwidth and cycle time under 2 ms, the protocol can robustly achieve a system probability of error better than 10^{-9} with a nominal SNR less than 2 dB with Rayleigh fading.**

***Keywords*—Cooperative communication, network coding, low-latency, high-reliability wireless, industrial control, diversity, Internet of Things**

I. Introduction

The Internet of Things (IoT) promises to enable many exciting new applications in health-care, robotics, transportation and entertainment [1]. For IoT applications that are interactive and immersive or involve control, reliable communication protocols with latencies around 1ms are crucial [1].

The work in [2] showed that the techniques used by existing wireless standards are fundamentally ill-suited for low-latency and high-reliability and established the need to attack this problem from the PHY/MAC layers. Proposed wireless architecture focused on low-latency operation through the use of reliable broadcasting, semi-fixed resource allocation, multiple-antennas for diversity and low-rate coding.

In our previous work [3], we introduced a cooperative communication protocol framework, designed to meet the stringent QoS requirements of low-latency (order of 1ms) and high-reliability (order of 10^{-9}). We showed that multi-user diversity can achieve the desired reliability without relying on time or frequency diversity created by natural multipath or frequency selectivity.

This paper integrates network coding into the cooperative communication protocol. We show that the resulting protocol further reduces the SNR required, as compared to the original Occupy CoW approach from [3]. The key idea here is that relays simultaneously broadcast downlink and uplink packets by XORing them as in [4]. In the Occupy CoW scheme any node which can potentially help unreachable nodes tries to help. By contrast, in the new XOR-CoW scheme only those nodes which already have good links to the nodes in need will help. This ensures efficient use of resources in the network as well as a reduction of noise.

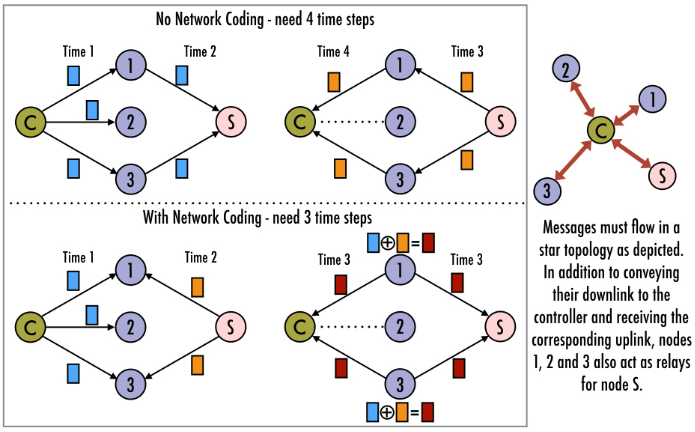


Fig. 1: Illustration of network coding along with simultaneous retransmissions where the C and S nodes have information to convey to each other through 3 relays 1 through 3. The bold lines indicate active links and the dotted lines indicate inactive links. The blue packets are the downlink packets, the orange packets are the uplink packets and the maroon packets are the XORed packets. The XOR scheme can communicate the same amount of information in a shorter time because the uplink and downlink demands are satisfied simultaneously.

We first briefly review some of the recent trends in wireless communications. We then briefly review the evolution of communication for industrial control as well as cooperative communication, wireless diversity and network coding techniques. Section II describes our multi-user-diversity and network coding based protocol. Section III presents how our protocol performs and compares it to hypothetical frequency-diversity-based scheme. Section IV shows that the protocol is robust to uncertainty in fading models.

A. *Recent development in 5G protocols*

Current vision of the 5G wireless not only focuses on increasing capacity and energy efficiency but also on reducing latency while also using mmWave frequencies [5] for enabling tactile applications. Very short RTT latencies, of the order of 1ms allow tactile feedback to wireless users, enabling immersive virtual-reality applications [6].

Recent works like [7] concentrate on the proposed 5GETLA radio interface and show that latencies below 1 ms for payloads of size 50kb are achievable provided a bandwidth of 100MHz is available. Though the targeted latency is similar to ours, they do not consider reliability guarantees or retransmissions. [8], [9] focus on efficient communication of short packets in the information-theoretic context. The key insight of [7]–[9] is that when short packets are transmitted, it is crucial to take into account the communication resources that are invested in the transmission of metadata. In this paper on XOR-CoW, we do not take metadata into consideration and plan to address this in later work.

B. Industrial Control

Communication in industrial control is supported by wired fieldbus systems like HART, PROFIBUS, WorldFIP, Foundation Fieldbus, and SERCOS [10] meet these requirements. Several wireless extensions of these fieldbus systems such as [11], [12] (as well as WirelessHART [13] and ISA100 [14]) which are based on wireless sensor network (WSN) techniques have been developed. They have worst case latencies on the order of hundreds on milliseconds [15] making them unsuitable for high-performance control applications. WISA [16] targeted wireless control by employing frequency hopping techniques but it achieves latency on the order of 10ms with a reliability of 10^{-4} [17], which fails to meet the reliability achieved by wired fieldbuses.

C. Cooperative communication, multi-user diversity and network coding

In our prior work [3], we discussed some of the relevant references on cooperative communication and multi-user diversity in detail. Low-latency applications like ours cannot use time diversity since the cycle time can be shorter than the coherence time. Hence commonly used time and frequency diversity techniques in WSN-inspired technologies [18] like channel hopping and contention-based MACs aren't sufficient to obtain the required diversity. As there are multiple nodes in the system, the key idea is to harvest cooperative and multi-user diversity. Cooperation amongst distributed antennas can provide full diversity without the need for physical arrays [19]. Even with a noisy inter-user channel, multi-user cooperation increases capacity and leads to achievable rates that are robust to channel variations [20]. [21], [22] use relays and a TDMA-based scheme to bring sender-diversity techniques to industrial control but unfortunately, TDMA-based schemes do not scale well with network size.

The contribution of this paper is in bringing together ideas from cooperative communication and network coding to design a wireless communication system that achieves QoS similar to wired fieldbus systems. The seminal work [23] showed that regarding information to be multicast as a "fluid" to be routed or replicated in general is not optimal and employing coding at nodes can lead to efficient use of bandwidth. This idea was further studied in [24], which proposes a forwarding architecture for wireless mesh networks to improve throughput by introducing a coding layer in between the IP and MAC layers. They provide a practical implementation of network coding into the current network stack, addressing the common case of unicast traffic, and dynamic and potentially bursty flows. Recent results by [4] show that using randomized space time block coding (RSTBC) in two way relay networks improves throughput by exchanging data through a bidirectional relay network. Like most works using network coding, we aim to increase throughput which translates to lower latency. Fig. 1 illustrates how we use network coding combined with simultaneous retransmissions in our work.

The proposed wireless communication system brings together ideas from cooperative communication and network coding and achieves the desired QoS requirements by exploiting multi-user diversity and distributed space-time codes (such as those in [25]–[27], so that each receiver can harvest a large diversity gain) to achieve high-reliability and low latency.

II. Protocol Design

A. XOR-CoW Protocol

We begin with a detailed description of our XOR-CoW protocol. The XOR-CoW protocol exploits multi-user diversity and network-coding by using simultaneous relaying to enable ultra-reliable two-way communication between a central controller (C) and a set of n slave nodes (S) within a "cycle" of length T. Distinct messages (size b bits) flow in a star topology from the central controller to individual nodes (downlink), and in the reverse direction from the slave nodes to the controller (uplink).

We assume that while normally, the controller and all nodes are in-range of each other, bad fading events can cause transmissions to fail. Successful nodes that have received both the downlink (from controller) and uplink (from the node in need) messages for a slave node in need, XOR the messages together to form a single packet. They then broadcast the XORed packet simultaneously. The controller uses the XORed packet as well as the downlink information that it already has to decode the uplink packet. The slave node uses the XORed packet as well as the uplink information that it already has to decode its downlink packet.

This scheme has three phases: downlink (length T_D), uplink (T_U) and XOR (T_X) such that $T_D + T_U + T_X = T$ which is the total time. We will describe the protocol with the aid of Fig. 2 where the network consists of one controller and 4 nodes (S1 - S4). To the left of the figure are the downlink buffers at each node and to the right of the figure are the uplink buffers at each node. They get populated as messages are decoded. Initially, the controller's downlink buffer is full as it is the origin of all downlink messages (shown by the striped buffers) and its uplink buffer is empty. S1 - S4 start with their corresponding uplink buffer full (shown by the striped buffers) and downlink buffers empty. The starred messages are those that each node (controller or others) is interested in receiving. The controller is interested in the uplink messages of slave nodes and the slave nodes are interested in receiving the specific downlink message intended for them.

Resource assumptions: These assumptions are the same as the ones made in [3] for "Occupy CoW" protocol. All the nodes share a universal addressing scheme and order, and messages contain their destination address.

Errors are caused only by bad fades. The round trip time is in the order of milliseconds and such short cycle times result in the non-ergodic flat-fading regime. Hence, time diversity cannot be used. All nodes are half-duplex but can switch instantly from transmit mode to receive mode.

Clocks on each of the nodes are perfectly synchronized in both time and frequency. This could be achieved by adapting techniques from [28]. The protocol relies on time/frequency synchronization to achieve simultaneous retransmission of messages by multiple relays. We assume that if k relays simultaneously (with consciously introduced timing dither[1]) transmit the exact same information, then all receivers can extract signal diversity k.

Downlink and Uplink Phase:

During these phases, all the nodes are listening whenever they are not transmitting. The cycle starts with the downlink phase in which the controller broadcasts a single packet consisting of all b-bit messages to all n slave nodes at rate $R_D = \frac{b \cdot n}{T_D}$. In Fig 2a, S1 and S2 successfully decode the entire downlink message. Their starred buffers are filled along with the downlink buffers corresponding to other nodes.

This is followed by the uplink phase, in which the individual nodes transmit their messages (including one bit for an ACK/NAK to the downlink message) to the controller one by one according to a predetermined schedule at rate $R_U = \frac{b+1}{T_U/n} = \frac{(b+1) \cdot n}{T_U}$ by evenly dividing the time slots among all slave nodes.

In Fig 2b, the controller successfully decodes the uplink messages of S1 and S2 and the starred uplink buffers of the controller corresponding to these nodes are filled. Since all nodes are listening whenever they are not transmitting, S1 also receives the uplink messages of S3 and S4. S2 receives the uplink message of S4. The nodes which have successfully received the downlink message as well as successfully transmitted their uplink message to the controller are referred to as **strong nodes**. In Fig. 2, S1 and S2 are the strong nodes.

Scheduling Phase:

In this phase the controller transmits acknowledgments to the strong nodes. This is just 2 bits of information per slave node for downlink and uplink. The common-information about the system's state enables the strong nodes to share a common schedule for relaying messages for the remaining nodes.

XOR phase:

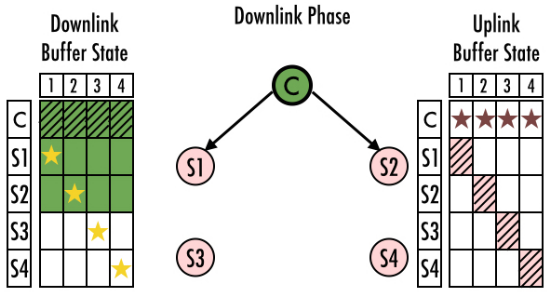
In this phase, each unsuccessful slave node has a slot allocated. The strong nodes XOR the downlink and uplink messages of each of the unsuccessful nodes they've heard. During the slot of an unsuccessful node (say node Y), all the strong nodes that have successfully heard node Y act as simultaneous broadcast relays and transmit the XORed packet using a distributed-space-time-code.

In Fig. 2c and Fig. 2d, S3 and S4 are the unsuccessful nodes. In the XOR slot allocated for S3, S1 XORs the downlink and uplink packet of S3 (represented by the maroon packet) and broadcasts it. Using the downlink packet of S3, the controller can now recover the uplink packet. Using its own uplink packet, S3 can now recover the downlink packet. The process for S4 is similar and the difference lies in the fact that S1 and S2 simultaneously transmit the XORed packet for S4.

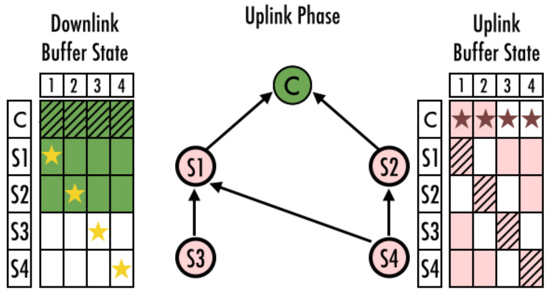
Actually, there are two kinds of schedule for this phase: predetermined or adaptive.

Predetermined (no adaptive) schedules:

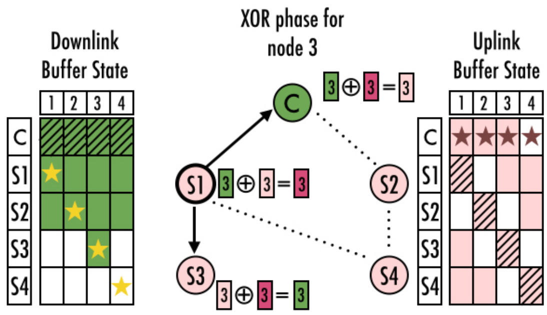
[1]To transform spatial diversity into frequency-diversity [27].



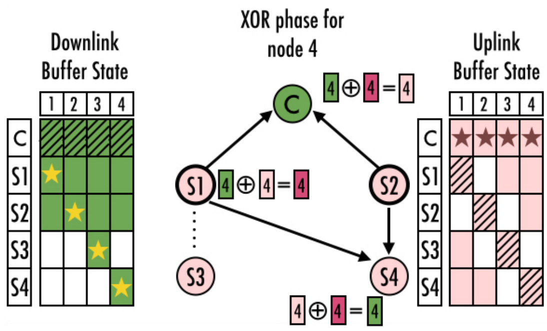
(a) Controller broadcasts downlink packet and nodes 1 and 2 successfully decode it.



(b) Nodes 1 through 4 send their uplink packets. Nodes 1 and 2 are successful and inter-node links are realized.



(c) Node 1 XORs the DL and UL packets of node 3 (resulting in the green packet) and broadcasts it. Node 3 decodes the DL packet and the controller decodes node 3's UL packet.



(d) Nodes 1 and 2 XOR the DL and UL packets of node 4 (resulting in the green packet) and broadcast it. Node 4 decodes the DL packet and the controller decodes node 4's UL packet. This terminates the XOR phase and all nodes have the desired packets.

Fig. 2: Instance of XOR-CoW with a controller and 4 nodes. Links that are activated are shown in solid bold line.

In this scheme, the time allocated for the XOR scheme is evenly divided among **all** the slave nodes regardless of whether they succeeded and the rate of transmission is $R_X = \frac{b \cdot n}{T_X}$. During the slot of an unsuccessful node, the strong nodes simultaneously relay the XORed packet as described earlier. During the slot of a successful node (say node Z), there are no transmissions. Note that for this scheme the scheduling phase is optional. Since the strong nodes have heard the nodes they have a link to in the uplink phase, they can determine using the ACK in the uplink message if a slave node needs help. By the channel reciprocity assumption, this is also when the controller needs help. Since the schedule is predetermined, the time at which the message of a particular node is to be transmitted is also known.

Adaptive schedules:

In this scheme, the time allocated for the XOR scheme is evenly divided among the unsuccessful nodes only (say n_1 of them) and the rate of transmission is $R_X = \frac{b \cdot n_1}{T_X}$. All nodes are assumed to be capable of instantly decoding variable-rate transmissions [29] so they keep listening until they get the message intended for them.

B. Occupy CoW Protocol

We now briefly summarize Occupy CoW protocol that is used for comparison purposes. For a detailed description, refer to [3]. The Occupy CoW protocol has the same network setup and aims to meets the same requirements as XOR-CoW.

Downlink phase I (length T_{D_1}), uplink phase I (length T_{U_1}) and the scheduling phase are similar to the downlink, uplink and scheduling phases of XOR-CoW described above.

In downlink phase II (length T_{D_2}) and III (length T_{D_3}), the controller and strong nodes alter the broadcast message to remove already-successful messages for the strong nodes and simultaneously broadcast the adapted packet. The unsuccessful slave nodes are listening. At the end of this phase, the nodes who received their messages from the controller have also received the global ACK information. This allows these nodes to participate as relays in the uplink phases since they can calculate the uplink transmission schedule.

The calculated schedule from earlier phases allocates a slot for each unsuccessful node from uplink phase I in uplink phases II (length T_{U_2}) and III (length T_{U_3}). Time slots are again divided evenly among all unsuccessful slaves. In the slot for each failed slave node, the slave node and everyone who heard that slave node in an earlier uplink phase will simultaneously transmit the relevant message using a distributed-space-time-code. The protocol can either have 2 hops or 3 hops.

III. Analysis of XOR-CoW

We explore XOR-CoW with parameters taken from today's practical application, the industrial printer case described in [2]. The SERCOS III protocol [30] supports the printer's cycle time of 2 ms with reliability of 10^{-8}. We target the following system requirements for the industrial printing application: 30 moving printing heads that move at speeds up to 3 m/s over distances of up to 10 m. Every cycle lasts 2ms and in each cycle the controller transmits 20 bytes of actuation data to each head and each of the 30 sensors transmit 20 bytes of sensory data to the controller. Assuming access to a single 20MHz wireless channel, this 4.8 Mbit/sec throughput corresponds to an overall spectral efficiency of approximately 0.25 bits/sec/Hz. SERCOS supports a reliability of 10^{-8} and for our protocol we target a reliability of 10^{-9}.

A. Behavioral assumptions for analysis

Our analysis depends on the following behavioral assumptions in addition to the resource assumptions in Sec. II-A. We assume a fixed nominal SNR on all links with independent Rayleigh fading on each link. We also assume channel reciprocity. Our model assumes a single tap channel[2] (hence flat-fading). Because the cycle-time is so short, we use the delay-limited-capacity framework [31], [32].

A link with fade h and bandwidth W is deemed good (thus no errors or erasures) if the rate of transmission R is less than or equal to the link's capacity $C = W \log(1 + |h|^2 \mathit{SNR})$. Consequently, the probability of link failure is defined as $p_{link} = P(R > C) = 1 - \exp\left(-\frac{2^{R/W}-1}{\mathit{SNR}}\right)$.

As in [3], if there are k simultaneous transmissions, then each receiver harvests perfect sender diversity of k. For analysis, this is treated as k independent tries that only fails in communicating the message if all the tries fail.

As in [3], we do not consider any dispersion-style finite-block-length effects on decoding. This can be justified in spirit by [33]. As in [3] we assume that transmission related errors are always detected [34].

B. XOR-CoW probability of failure

We will analyze and calculate the probability of failure for an allocation of time for the downlink and uplink phases such that $R_D = R_U = \frac{b \cdot n}{T_D}$. We will denote this by R_{DU}. We will explore how smartly allocating lengths for downlink + uplink and XOR phases can help the performance. Let the set of nodes that successfully decoded the downlink transmissions be A (cardinality a). By channel reciprocity, the controller decodes the uplink transmissions of the same set of nodes A. The rest of the nodes in the set $S \backslash A$ need help to succeed.

The rate in the XOR phase depends on the schedule. So we get $R_X = \frac{b \cdot (n-a)}{T_X}$ if we choose adaptive schedule or we get $R_X = \frac{b \cdot n}{T_X}$ if we choose predetermined schedule. Irrespective of whether we choose to do adaptation, we are faced with two possible scenarios. The rate of transmission in the XOR phase R_X can either be higher or lower than the rate of transmission in the downlink and uplink phases R_{DU}. Failure is the event in which at least one of the n slave nodes has not received its downlink message by the end of the cycle or the controller has not received one of the n uplink messages by the end of the cycle.

Case 1: $R_{DU} \geq R_X$

As the rate in the XOR phase remains the same or gets better, all the links which existed in the downlink and uplink phases stay intact. In this case, failure is the event that at least one of the nodes in the set $S \backslash A$ did not reach the set A during the uplink phase.

[2]Performance would improve if we reliably had more taps/diversity.

Let $p_{DU} = P(R_{DU} > C)$ be the probability of link failure in the downlink and uplink phases. Then,

$$P\big(|A| = a\big) = \binom{n}{a}(1 - p_{DU})^a (p_{DU})^{n-a}. \quad (1)$$

This probability does not depend on whether the rate decreases or increases (it is the same for both), hence we do not condition on it. Then conditioned on the number of nodes which are strong, the probability of failure of the system in Case 1 is given by

$$P\big(\text{fail} \mid |A| = a, \text{Case 1}\big) = 1 - \big(1 - (p_{DU})^a\big)^{n-a}. \quad (2)$$

Putting Eq. (1) and Eq. (2) together we get

$$P\big(\text{fail}, |A| = a \mid \text{Case 1}\big) = \left\{\binom{n}{a}(1 - p_{DU})^a (p_{DU})^{n-a}\right\} \times \left\{1 - \big(1 - (p_{DU})^a\big)^{n-a}\right\} \quad (3)$$

Notice that the probability of system failure only depends on the probability of link failure in the downlink or uplink phases (p_{DU}). Hence, in an adaptive scheduling scheme, lowering the rate in the XOR phase (R_X) below R_{DU} will not bring any benefits. Keeping the rate R_X at least as much as R_{DU} (thus reducing the time spent in the XOR phase) is the right thing to do. In section III-C, we will see how allocating less time for XOR phase and doing adaptive scheduling brings in SNR benefits.

Case 2: $R_{DU} < R_X$

As the rate in the XOR phase gets worse than the rate in downlink and uplink phase, some of the links that previously existed could vanish. This would mean that some nodes in the set A can not help in transmitting information anymore. Let the nodes in set A that can act as relays in the XOR phase be $\widetilde{A}$ and those that can not be $\widehat{A}$. In order to succeed, the nodes in $S \backslash A$ need to have a link to at least one node in the set $\widetilde{A}$ in the XOR phase (having a link in the XOR phase implies that the link existed in the downlink and uplink phases too). Let $p_{DU} = P(R_{DU} > C)$ be the probability of link failure in the downlink and uplink phases and $p_X = P(R_X > C)$ be the probability of link failure in the XOR phase. Let the probability of a link existing in the downlink and uplink phases and vanishing in the XOR phase be denoted by q. Then we have

$$q = P\big(R_X > C \mid C > R_{DU}\big) = \frac{p_X - p_{DU}}{1 - p_{DU}}. \quad (4)$$

Eq. (1) continues to hold for $P\big(|A| = a\big)$ and the probability that $\widetilde{a}$ remain effective in the XOR phase is given by,

$$P\big(|\widetilde{A}| = \widetilde{a} \mid |A| = a\big) = \binom{a}{\widetilde{a}}(1 - q)^{\widetilde{a}}(q)^{a-\widetilde{a}}. \quad (5)$$

Then conditioned on the number of successes in the downlink and uplink phases and the number of nodes which are strong in the XOR phase, the probability of failure is given by

$$P\big(\text{fail} \mid |A| = a, |\widetilde{A}| = \widetilde{a}, \text{Case 2}\big) = 1 - \Big(1 - (p_X)^{\widetilde{a}}\Big)^{n-a} \quad (6)$$

Putting Eq. (1), Eq. (5) and Eq. (6) together we get

$$\begin{aligned} &P\big(\text{fail}, |A| = a \mid \text{Case 2}\big) \\ &\quad= \sum_{\widetilde{a}=0}^{a} P\big(\text{fail} \mid |A| = a, |\widetilde{X}| = \widetilde{a}, \text{Case 2}\big) \\ &\quad\times P\big(|\widetilde{A}| = \widetilde{a} \mid |A| = a\big) \times P\big(|A| = a\big) \end{aligned} \quad (7)$$

Putting Eq. (3) and Eq. (7) together we get

$$\begin{aligned} P\big(\text{fail}\big) = \sum_{a=0}^{n-1} \Big\{ &P\big(\text{fail}, |A| = a \mid \text{Case 1}\big) \mathbb{1}\{R_{DU} > R_X\} \\ &+ P\big(\text{fail}, |A| = a \mid \text{Case 2}\big) \mathbb{1}\{R_{DU} < R_X\} \Big\}. \end{aligned} \quad (8)$$

Before we discuss how XOR-CoW performs when compared to Occupy CoW and frequency hopping techniques, we first state two theorems that are useful for explaining the results. We omit the proofs due to space limitations and will be including them in the extended version.

Theorem 1: If an instance of no-rate-adaptation 2-hop Occupy CoW protocol with equal downlink and uplink phases ($T_{D_1} = T_{U_1} = T_{D_2} = T_{U_2} = T_M$) succeeds, then there is a common downlink and uplink success path for each node in the network.

If a node successfully decoded the downlink message in 1 hop, its uplink message also gets through successfully to the controller in one hop. If a node successfully decoded the downlink message in 2 hops via a relay Z, then the same relay helps uplink as well.

Theorem 2: If an instance of no-rate-adaptation 2-hop Occupy CoW protocol with equal downlink and uplink phase 1 ($T_{D_1} = T_{U_1} = T_M$) and a given SNR succeeds, then XOR-CoW with downlink and uplink phase lengths both equal to T_M and XOR phase length also equals to T_M succeeds at the same SNR.

A corollary of Theorem 2 is that while 2-hop Occupy CoW would require time $4 \times T_M$ *to succeed, XOR-CoW succeeds in time* $3 \times T_M$. From Theorem 1 we know that if we allocate equal time (T_M) to all phases with no rate adaptation, then downlink and uplink success have a common path. This common path allows the XOR-CoW scheme to succeed.

C. Results and comparison

Following [2], [3] and the communication-theoretic convention, we use the minimum SNR required to achieve 10^{-9} reliability as our metric to compare XOR-CoW to other schemes.

Fig. 3 looks at performance with fixed payload size $b = 160$ bits and $b = 480$ bits as the number of nodes, n, varies. We compare 2-hop Occupy CoW with predetermined schedule and adaptive schedule XOR-CoW.

Let us first consider the four curves in the bottom with no markers which correspond to $b = 160$ bits. The first solid curve in the set is the 2-hop Occupy CoW protocol with phase lengths equally allocated to the four phases. Initially

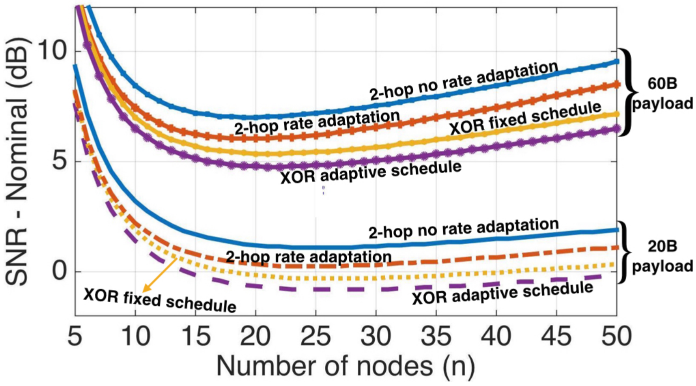


Fig. 3: The performance of XOR-CoW as compared with 2-hop Occupy CoW for $b = 160$ bit messages and $b = 480$ bit messages and varying network size with 20MHz bandwidth and a 2ms cycle time, aiming at 10^{-9}.

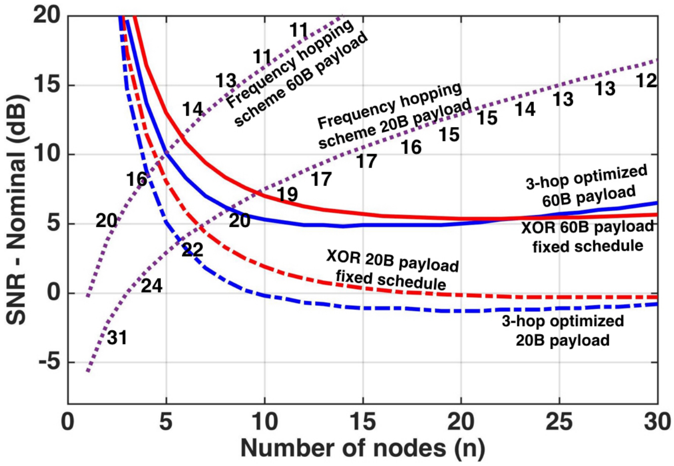


Fig. 4: The performance of XOR-CoW, optimized 3 hop Occupy CoW and frequency-hopping for $b = 160$ bit and $b = 480$ bit messages and varying network size with 20MHz bandwidth and a 2ms cycle time, aiming at 10^{-9}.

the minimum required SNR for Occupy CoW decreases with increasing n, even through the throughput increases as $b \cdot n$, but the curve rises back again. This is because the impact of multi-user diversity eventually gives way and the required SNR would start to increase for very large n.

The second dash-dotted curve is the 2-hop Occupy CoW protocol with phase lengths equally allocated to the four phases but allowing for adaptive schedule. Clearly this does better than 2-hop Occupy CoW protocol without adaptation.

The third dotted curve is the XOR-CoW protocol with phase lengths equally allocated to the three phases (downlink uplink and XOR) with fixed schedule. From Theorem 2 we know that if we had allocated three-fourths of the available time to XOR-CoW, it would succeed at the same SNR as the topmost curve. This protocol meets the reliability requirement at an SNR below the rate adaptive 2-hop Occupy CoW protocol. For the same SNR and total block length, XOR-CoW protocol can support at least 33% more users as compared to 2-hop Occupy CoW protocol.

The lowest dashed curve is the XOR-CoW protocol with ratio of $T_D : T_U : T_X = 1 : 1 : 2/3$ and allowing adaptive scheduling. We see that allowing adaptive scheduling lowers the SNR further. To be precise, at a constant number of users, it reduces the minimum required SNR by ≈ 0.5dB and at a constant SNR, it allows about 10 users more than XOR-CoW with fixed schedule.

The first four curves with markers are for $b = 480$ bits in the order 2-hop Occupy CoW with no rate adaptation, 2-hop Occupy CoW with rate adaptation, XOR-CoW with fixed schedule, XOR-CoW with adaptive schedule. We see that the gap between the SNR required for the 2-hop scheme and XOR-CoW is even bigger for higher aggregate rates. The SNR gap is about 4dB for $b = 480$ bits at 50 nodes as opposed to 2dB for $b = 160$ bits at 50 nodes.

Fig. 4 compares an optimized 3 hop Occupy CoW, fixed-schedule XOR-CoW and a hypothetical frequency-hopping scheme (this was also looked at in [3]). We see that the optimized 3-hop Occupy CoW and fixed-schedule XOR-CoW perform comparably for $b = 160$ bits. The advantage of XOR-CoW is clear for high aggregate rates and large networks as shown by the top blue and red curves in Fig. 4. We see that XOR-CoW beats the performance of Occupy CoW for $b = 480$ bits and network size > 20 while also being a more simpler scheme.

The dotted purple curves represent a hypothetical (non-adaptive) frequency-hopping scheme that divides the bandwidth $W = 20$MHz into k sub-channels that are assumed to be independently faded, for $b = 160$ bits and $b = 480$ bits. The curves are annotated with the optimal k. As k (and thus hops) increases, the available diversity increases, but the added message repetitions force the instantaneous link data rate higher. For low n the scheme prefers more frequency hops because of the diversity benefits. The SNR cost of doing this is not so high because the throughput is low enough that we are still in the linear-regime of channel capacity. For fewer than 7 nodes, this says that using frequency-hopping is great — as long as we can reliably count on about 20 independently faded sub-channels to repeat across.

IV. Robustness of XOR-CoW

In [3], we addressed whether Occupy CoW relied on the perfect knowledge of deep fading distributions to achieve the extremely low probabilities that we target in a wireless setting. We concluded that Occupy CoW does not rely on perfect knowledge of deep fading distributions and achieves high-reliability by relying on the independence of link failures.

Similarly, we analyze the impact of modeling error and uncertainty on the XOR-CoW protocol. We account for modeling error and interference introduced by the local environment as an extra probability of failure at each link, p_{env}, on top of the probability of error due to nominal fading, p_{fade}. The probability of link failure is the combined effect of both these parts, p_{link}, is $p_{link} = p_{fade} + p_{env}$.

Fig. 5 shows that there is an SNR penalty to be paid in order to attain the same performance under extra modeling error. For really small network sizes ($n < 15$) achieving the probability of error requirement robustly can become impossible (when the desired tolerance offset to modeling

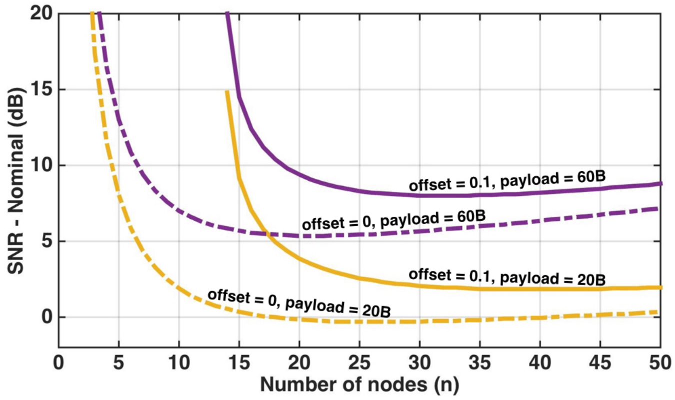


Fig. 5: SNR penalty to achieve performance robustly in XOR-CoW with fixed schedule. For an increasing amount of robustness, a higher SNR is required as well as more nodes.

error is greater than the maximum tolerable probability of link failure). For moderate to large network sizes ($n \geq 25$), the protocol only has a small SNR penalty ($\approx 3dB$). We conclude that similar to the Occupy CoW protocol, XOR-CoW also does not rely on perfect knowledge of deep fading distributions and achieves high-reliability by relying on the independence of link failures.

Acknowledgements

Thanks to Venkat Anantharam and Sahaana Suri for useful discussions. We also thank the BWRC students, staff, faculty and industrial sponsors and the NSF for a Graduate Research Fellowship and grants CNS-0932410, CNS-1321155, and ECCS-1343398.

References

[1] G. Fettweis, "The Tactile Internet: Applications and Challenges," *Vehicular Technology Magazine, IEEE*, vol. 9, no. 1, pp. 64–70, March 2014.

[2] 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.

[3] V. 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.

[4] 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.

[5] J. Andrews *et al.*, "What Will 5G Be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[6] N. S. Networks, "2020: Beyond 4G: Radio Evolution for the Gigabit Experience," 2011.

[7] T. Levanen *et al.*, "Low latency radio interface for 5G flexible TDD local area communications," in *Communications Workshops (ICC), 2014 IEEE International Conference on*, June 2014, pp. 7–13.

[8] G. Durisi *et al.*, "Towards massive, ultra-reliable, and low-latency wireless: The art of sending short packets," *CoRR*, vol. abs/1504.06526, 2015. [Online]. Available: http://arxiv.org/abs/1504.06526

[9] P. Popovski, "Ultra-reliable communication in 5g wireless systems," *CoRR*, vol. abs/1410.4330, 2014. [Online]. Available: http://arxiv.org/abs/1410.4330

[10] R. Zurawski, *Industrial Communication Technology Handbook*. CRC Press, 2005.

[11] A. Willig, "An architecture for wireless extension of PROFIBUS," in *The 29th Annual Conference of the IEEE Industrial Electronics Society*, vol. 3, Nov 2003, pp. 2369–2375 Vol.3.

[12] P. Morel *et al.*, "Requirements for wireless extensions of a FIP fieldbus," in *1996 IEEE Conference on Emerging Technologies and Factory Automation*, vol. 1, Nov 1996, pp. 116–122 vol.1.

[13] International Electrotechnical Commission, *Industrial Communication Networks-Fieldbus Specifications, WirelessHART Communication Network and Communication Profile*. British Standards Institute, 2009.

[14] ISA100, "ISA100.11a, An update on the Process Automation Applications Wireless Standard," in *ISA Seminar, Orlando, Florida*, 2008.

[15] J. Akerberg *et al.*, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*. IEEE, 2011, pp. 410–415.

[16] R. Steigmann *et al.*, "Introduction to WISA: Wireless Interface for Sensors and Actuators ," July 2006.

[17] A. Willig, "Recent and Emerging Topics in Wireless Industrial Communications: A Selection," pp. 102–124, May 2007.

[18] P. Zand *et al.*, "Wireless Industrial Monitoring and Control Networks: The Journey So Far and the Road Ahead," *J. Sensor and Actuator Networks*, vol. 1, no. 2, pp. 123–152, 2012.

[19] J. Laneman *et al.*, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, Dec 2004.

[20] A. Sendonaris *et al.*, "User cooperation diversity. Part I. System description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, Nov 2003.

[21] A. Willig, "How to exploit spatial diversity in wireless industrial networks," *Annual Reviews in Control*, vol. 32, no. 1, pp. 49 – 57, 2008.

[22] S. Girs *et al.*, "Increased reliability or reduced delay in wireless industrial networks using relaying and Luby codes," in *IEEE 18th Conference on Emerging Technologies Factory Automation, 2013*, Sept 2013, pp. 1–9.

[23] R. Ahlswede *et al.*, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.

[24] S. Katti *et al.*, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking (ToN)*, vol. 16, no. 3, pp. 497–510, 2008.

[25] F. Oggier *et al.*, "Perfect spacetime block codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 9, pp. 3885–3902, 2006.

[26] P. Elia *et al.*, "Perfect Space-Time Codes for Any Number of Antennas," *IEEE Trans. Inform. Theory*, vol. 53, no. 11, pp. 3853–3868, 2007.

[27] G. Wu *et al.*, "Selective Random Cyclic Delay Diversity for HARQ in Cooperative Relay," in *IEEE Wireless Communications and Networking Conference (WCNC), 2010*, April 2010, pp. 1–6.

[28] Q. Huang *et al.*, "Practical Timing and Frequency Synchronization for OFDM-Based Cooperative Systems," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3706–3716, July 2010.

[29] S. Verdú *et al.*, "Variable-rate channel capacity," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2651–2667, 2010.

[30] "Introduction to SERCOS III with industrial ethernet." [Online]. Available: http://www.sercos.com/technology/sercos3.htm

[31] S. Hanly *et al.*, "Multiaccess fading channels. II. Delay-limited capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2816–2831, Nov 1998.

[32] L. Ozarow *et al.*, "Information theoretic considerations for cellular mobile radio," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 2, pp. 359–378, May 1994.

[33] 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, 2014.

[34] G. D. Forney, "Exponential error bounds for erasure, list, and decision feedback schemes," *IEEE Transactions on Information Theory*, vol. 14, pp. 206–220, 1968.