Real-Time Cooperative Communication for Automation Over Wireless

Vasuki Narasimha Swamy, Sahaana Suri, Paul Rigge, Matthew Weiner, Gireeja Ranade, Anant Sahai, and Borivoje Nikolić, *Fellow, IEEE*

***Abstract*— High-performance industrial automation systems rely on tens of simultaneously active sensors and actuators and have stringent communication latency and reliability requirements. Current wireless technologies, such as Wi-Fi, Bluetooth, and LTE are unable to meet these requirements, forcing the use of wired communication in industrial control systems. This paper introduces a wireless communication protocol that capitalizes on multiuser diversity and cooperative communication to achieve the ultra-reliability with a low-latency constraint. Our protocol is analyzed using the communication-theoretic delay-limited-capacity framework and compared with baseline schemes that primarily exploit frequency diversity. For a scenario inspired by an industrial printing application with 30 nodes in the control loop, 20-B messages transmitted between pairs of nodes and a cycle time of 2 ms, an idealized protocol can achieve a cycle failure probability (probability that any packet in a cycle is not successfully delivered) lower than 10^{-9} with nominal SNR below 5 dB in a 20-MHz wide channel.**

***Index Terms*— Cooperative communication, low-latency, high-reliability wireless, industrial control, diversity, IoT.**

I. Introduction

HIGH-SPEED ultra-reliable wireless communication networks are critical for developing near-real-time machine-to-machine networks and applications such as industrial automation, immersive virtual reality (VR) and the "tactile internet." An interactive Internet-of-Things (IoT) will require both ubiquitous sensing and simultaneous actuation of numerous connected devices. The latency requirements on the control loop for these applications will be in the low tens of milliseconds, with reliability requirements of one-in-a-million errors [1]. These specifications mirror those of industrial automation today [2], [3]. However, these are unattainable using WiFi, Bluetooth and Long-Term Evolution (LTE), and as a result industrial automation relies heavily on wired connections.

This paper[1] introduces a communication protocol framework for industrial control and IoT applications that is designed to meet stringent Quality-of-Service (QoS) requirements. The protocol relies on multi-user diversity to achieve reliability without relying on time or frequency diversity created by natural motion, multipath or frequency selectivity. Furthermore, combining this with simultaneous relaying allows strict latency requirements to be met at practically achievable signal-to-noise ratios (SNR).

In Section II we first review communication for industrial control, then discuss cooperative communication and wireless diversity techniques, and finally place this work within the context of 5G research and contrast it with the complementary research on the co-design of control and communication systems. During this review, we also argue why time diversity cannot be reliably harnessed for low-latency communication, why frequency diversity cannot be counted on when ultra-reliability is desired, and why randomized carrier sense multiple access with collision avoidance (CSMA/CA) style approaches cannot be used reliably. Then, Section III describes our multi-user-diversity-based protocol framework. Section IV compares the performance of our protocol to hypothetical frequency-diversity-based schemes as well as to schemes that do not leverage simultaneous transmissions. Finally, Section V examines the impact of fine-tuning the protocol parameters and explores duty-cycling to reduce power consumption and what this suggests for implementation. All the formulas used to generate the plots are included in an online techreport [5].

A. Main Results

The protocol in this paper ("Occupy CoW,[2]") targets a local wireless domain where nominally all nodes are in range, but fading might cause a pair of nodes to be unable to hear each other. The traffic patterns (what we deem the "information topology") of interest consist of steady streams of messages, each originating at possibly different nodes within the network, and each stream subscribed to by some (possibly different) subset of nodes within the network. Within a short period of

Manuscript received July 31, 2016; revised January 12, 2017 and June 20, 2017; accepted August 1, 2017. Date of publication August 25, 2017; date of current version November 9, 2017. This work was supported by the NSF for a Graduate Research Fellowship under Grant 0932410, Grant 1321155, Grant 1343398, and Grant 144078. The associate editor coordinating the review of this paper and approving it for publication was T. M. Lok. *(Corresponding author: Vasuki Narasimha Swamy.)*

V. Narasimha Swamy, P. Rigge, A. Sahai, and B. Nikolić are with the University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: vasuki@berkeley.edu).

S. Suri is with Stanford University, Stanford, CA 94305 USA.

M. Weiner is with RF Pixels, Fremont, CA 94538 USA.

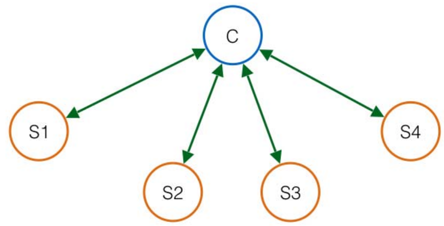
G. Ranade is with the University of California at Berkeley, Berkeley, CA 94720 USA, and also with Microsoft Research, Redmond, WA 98052 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

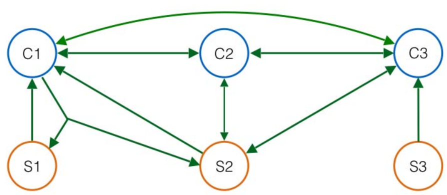
Digital Object Identifier 10.1109/TWC.2017.2741485

[1]This paper expands upon a conference version [4] that contained early forms of these results.

[2]OCCUPYCOW is an acronym for "Optimizing Cooperative Communication for Ultra-reliable Protocols Yoking Control Onto Wireless." The name also evokes the similarity between our scheme and the "human microphone" implemented by demonstrators during the "Occupy Wall Street" movement.



(a) Star message flow topology



(b) An example of a generic non-star message flow topology. Notice that one of the message streams originating at C1 has two subscribers: S1 and S2.

Fig. 1. Information flow topologies.

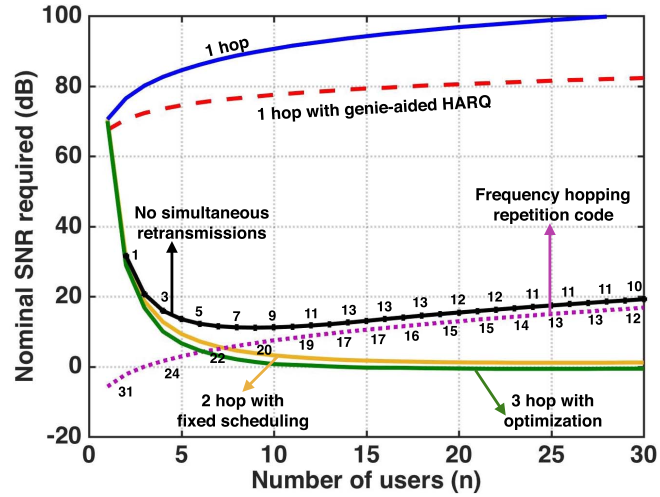


Fig. 2. The performance of Occupy CoW for a star information topology compared with reference schemes for $m = 160$ bit messages, $n = 30$ active nodes, and a 2ms cycle time, aiming at 10^{-9} probability of failure for a 20MHz channel. The numbers next to the frequency-hopping scheme show the frequency diversity needed and those next to the non-simultaneous retransmission scheme show the optimal number of relays per message stream.

time, deemed a "cycle time," every stream needs to deliver one packet reliably to its subscribers. The information topology can be arbitrary – something naturally centralized like a star topology as shown in Fig. 1a (e.g. with a central controller talking to many sensor/actuators collecting streams of measurements and sending streams of commands) or something more generic as in Fig. 1b.

The potential for deep fading is what frustrates the simple strategy of just giving each message stream its own time-slot. For example in Fig. 1a, if the communication link from 'C' to 'S3' were deeply faded, we would have a failure. We combat this by employing cooperative communication – if any of the other nodes can hear C, it can relay the relevant message streams. As reviewed in Section II-B, cooperative communication has been well studied in the wireless literature. In this paper, we specifically adapt it to the ultra-reliability low-latency regime.

We define the cycle failure probability as the probability that any packet transmitted during the cycle was unsuccessful. Our key findings are shown in Fig. 2, where the minimum SNR required to achieve a cycle failure probability of 10^{-9} is used to compare different protocols as the size of the network grows. We see that a one-hop scheme, one that does not use cooperative communication (the top blue line), requires an unreasonably high nominal SNR. Even idealized hybrid automatic repeat request (HARQ — the dashed red line) cannot eliminate the need to use high power to overcome a one-in-a-billion fading event. Harnessing diversity is required to do better. The purple dotted curve shows what idealized frequency hopping could achieve, assuming that nature has guaranteed sufficient frequency diversity (the number of independently faded subchannels required is labeled along the curve). For large enough networks, the black line in Fig. 2 shows that only slightly worse performance is available by using cooperative communication where a subset of the nodes (the number of active relays is marked on the curve) take turns to relay messages that they have heard. This does not count on the multipath environment guaranteeing a lot of frequency diversity and instead harnesses the spatial diversity that independently located nodes bring. A further 20dB of gain is possible by moving to the OccupyCoW protocol described in this paper that combines relaying with simultaneous transmission of messages by relays, and this is what is shown by the yellow and green curves. The difference between the yellow and green curves is what can be gained by optimizing the protocol parameters, and is not as significant by comparison. We will revisit this in detail in Sec. IV-F.

II. Related Work

A. Industrial Automation

Industrial control systems have traditionally been wired. Following trends in networking more broadly, proprietary point-to-point wired systems were replaced by *fieldbus* systems such as SERCOS, PROFIBUS and WorldFIP that provide reliable real-time communication [6]–[8]. The desire to move to wireless communications to reduce bulk and installation costs [9] led to an examination of wireless extensions of fieldbus systems [10], [11]. Unfortunately, these do not work in low-latency high-reliability settings since these wireless fieldbuses are largely derivative of wireless designs for non-critical consumer applications and incorporate features such as CSMA or ALOHA that can induce unbounded delays [12]. The IWLAN standard, which is based on a combination of PROFINET with 802.11n WLAN, attempts to resolve this issue by adding proprietary scheduled polling called iPCF (industrial Point Coordination Function) [13]–[16]. To deal with a deep fade, IWLAN has to rely on handing over the faded node to a redundant access point, and this handover is sped up by using proprietary industrial-automation oriented enhancements to the 802.11n protocol. Even with these enhancements, handover can only occur at the time-scale of tens of milliseconds [13].

Ideas from Wireless Sensor Networks (WSNs) [17]–[19] that provide high-reliability monitoring also cannot be easily adapted for very tight control loops because they inherently tolerate large latencies [20]. The Wireless Interface for Sensors and Actuators (WISA) [21] attempts to meet stringent real-time requirements, but the reliability of WISA ($\approx 10^{-4}$) does not meet the desired specifications [22]. ZigBee PRO [23] also fails to deliver high enough reliability [24]. Both ISA 100 [25] and WirelessHART [26] provide secure and reliable communication, but cannot meet the latency bounds as *each* packet is 10ms long. The median latency for a successful delivery is in the order of 100ms and the protocols are heavily optimized for power consumption. On the other hand, we are interested in packet sizes on the order of 10μs with maximum delivery latency of 2ms [24], [27]. There is a need for a fundamentally faster and more reliable protocol framework if we want to have a drop-in replacement for existing wired fieldbuses like SERCOS III, which simultaneously provide a reliability of 10^{-8} *and* latency of 1 ms when communicating among tens of concurrently active nodes.

B. Cooperative Communication and Multi-User Diversity

The key to getting reliability in wireless communication is to harness diversity [28]. Highly-reliable WSNs use techniques like channel hopping and contention-based medium access control (MAC) to harvest time and frequency diversity, and multi-path routing as an indirect way to harvest spatial diversity [9]. Unfortunately, low-latency applications like ours cannot use time diversity since the cycle times of single-digit milliseconds could very well be shorter than channel coherence times of tens of milliseconds. Techniques like Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) also do not provide much advantage in the face of fading [29]. Later in this paper, we demonstrate that frequency-diversity based techniques also fall short, especially when the required throughput pushes us to increase spectral efficiency. Even beyond the issue of poor performance, there is the issue of trust — exploiting frequency diversity requires us to trust that nature will provide enough multipaths with a large enough delay spread to actually create frequency diversity [28]. Consequently, our protocol leverages spatial diversity instead.

The size of the networks targeted in this paper is moderate (10 - 100 nodes active at once). Therefore, there is an abundance of antennas in the system and we can harvest some resulting diversity. Multi-antenna diversity is mainly of two types: a) sender diversity where multiple antennas transmit the same message through independent channels and b) receiver diversity where multiple receive antennas harvest copies of the same signal received via independent channels. Researchers have studied these techniques in great detail; so our treatment here is limited. Cooperation among distributed antennas can provide full sender-diversity without the need for physical arrays [30]. Even with a noisy inter-user channel, multi-user cooperation increases capacity and leads to achievable rates that are robust to channel variations [31]. The prior works in cooperative communication tend to focus on the asymptotic regimes of high SNR. By contrast, we are interested in low - moderate SNR regimes (around 10 dB).

Multi-antenna techniques have been widely implemented in commercial wireless protocols like IEEE 802.11. Sender-diversity harvesting techniques using relaying coupled with a time division multiple access (TDMA) based scheme have been explored for industrial control [29], [32]. Unfortunately, as we discuss later and can see in Fig. 2, strict TDMA for relays can scale poorly with network size since relaying for one message consumes many slots to get enough diversity to attain high reliability. To scale better with network size, our protocol uses simultaneous transmission by many relays, using some distributed space-time codes (DSTCs) such as those in [33]–[35], so that each receiver can harvest a large diversity gain. While we do not discuss the specifics of space-time code implementation, recent work by Rahul et al. [36] demonstrates that it is possible to implement schemes that harvest sender diversity by using concurrent transmissions.

C. Recent Developments in Proposed 5G Protocols

Latency and reliability have risen in importance as 5G wireless is discussed, taking their place alongside a focus on increasing capacity and energy efficiency while also using mmWave frequencies [37]. One important driver for very short round-trip time (RTT) latencies, of the order of 1ms, is to support tactile feedback to wireless users, enabling immersive VR applications [38].

Levanen et al. [39] concentrate on the proposed 5GETLA radio interface and show that latencies below 1ms for payloads of size 50kb are achievable provided a bandwidth of 100MHz is available. Though the targeted latency is of the same order as required by industrial control, they do not consider reliability guarantees or retransmissions. A discussion of the feasibility, requirements, and design challenges of an OFDM based 5G radio interface suitable for mission-critical machine type communication (MTC) concluded that multiple receive antennas are critical for interference mitigation [40]. In similar spirit, coverage and capacity aspects concerning both noise-limited and interference-limited operations for MTC were considered in [41]. Various PHY and MAC layer solutions for mMTC (massive MTC) and uMTC (ultra-reliable MTC) are discussed in [42] where they conclude that higher-layer considerations ensure lean signaling by enabling longer sleep cycles and other techniques. Efficient communication of short packets in the communication-theoretic context was discussed in [39], [43], and [44], where the key insight is that when packets are short, the resources needed for metadata (like preamble, header, etc.) transmission should be considered. In this paper, we do not consider either metadata or interference and plan to address these in later work.

D. Control and Communication Co-Design

This paper's approach to enabling wireless industrial automation is to maximize the reliability of communication while simultaneously reducing latency. For completeness, we mention that a second approach towards achieving successful wireless industrial automation would be to adapt control algorithms to (the reliability and latency guarantees provided by) wireless communication and to co-design the two modules. Fundamental limits for control and estimation of systems over both noiseless rate-limited channels [45], [46] and noisy channels [47] have been established. A series of works [48]–[50] established the limits of control and estimation over packet dropping networks and it was recently shown that control and communication co-design could provide unbounded performance gains in such settings [51].

The literature on adapting control to wireless communication has generally focussed on leveraging the optimization paradigms of control-theoretic synthesis. Works like [52]–[54] combine data rate and quantization with performance optimization and dealing with packet drops. A holistic view of network parameters including the placement of controller functionality has been studied in [55] and [56]. Finally, there are even more completely integrated approaches like the wireless control network idea proposed in Pajic et al. [57] wherein the wireless network itself is modeled as the controller with the network topology providing an implementation constraint and the unreliability of the links viewed through the lens of robust control techniques [58].

Our paper focuses exclusively on improving communication. This has two motivations. First, it follows the principle of layering as it allows unmodified control laws (which might not have been developed using any particular synthesis methodology or even stated performance criteria) to operate with a new communication layer [59]. Second, it establishes a baseline upon which we can study the gains achievable through co-design, which warrants further investigation.

III. Protocol Design

The Occupy CoW protocol exploits multi-user diversity by using simultaneous relaying (i.e. using diversity-oriented distributed space-time codes (DSTC)) to enable low-latency ultra-reliable communication between a set of nodes (say n of them) within a "cycle" of length T. As described in the introduction, we assume that all nodes are in-range of each other and have a given nominal SNR. However, a deep fading event can cause transmissions to fail. One could in principle wait for the channel condition to improve to a good fade. However, due to the coherence times being longer than the cycle time, channels do not change quickly enough. Therefore to reliably (meaning with low probability of failure) deliver packets, multiple paths to the destination need to be found.

The protocol is information-flow-centric rather than node-centric. There is an information topology (i.e. a list of streams having sources and subscribers; where each stream generates one fresh fixed-size data packet at the start of each cycle that must reliably reach all its subscribers during that cycle) that is known to everyone in advance. Each packet gets dedicated time slots for transmission as well as relay retransmissions. We have two main versions of the protocol, as summarized in Algorithm 1:

1) Fixed Schedule: Once an initial schedule (or order) of packets has been determined, all packets are transmitted once. Then, in the same order, all nodes that have the corresponding packet simultaneously retransmit it. This is a two hop version of the protocol. For three hops, all nodes that now have the corresponding packet simultaneously retransmit it again. We have restricted the number of hops to three. This is because in local networks where nominally nodes are in-range and thus presumably connected to each other, there is negligible improvement in performance (say SNR reduction) from going to higher hop counts. However when the networks are fundamentally wide (some flows need at least 2 hops to reach their destinations under nominal channel conditions), then going to higher hop counts would be necessary.

The inefficiency in the fixed schedule is that it dedicates slots to retransmit packets that were already successful. This forces all the retransmission slots to be shorter than they could have been. To avoid this, it seems like a good idea to adapt the retransmission schedule to concentrate only on the messages that need relaying. However, to achieve this, all the potential relays need to agree on which messages need to be retransmitted. This requires the reliable dissemination of scheduling information throughout the network, and acknowledgments (ACKs) from all of the network nodes are required before the retransmission of data packets.

2) Adaptive Schedule: Once an initial schedule (or order) of messages has been determined, all messages are transmitted once. All nodes then take turns broadcasting their own ACK packets where they indicate the messages that they have heard. These ACK packets are rebroadcast using the Fixed Schedule scheme above so that all nodes' ACK information gets reliably disseminated to everyone. Once all the ACKs are known, the data retransmission schedule is recomputed to include only those messages that have not yet reached all their subscribers and each data packet is in turn rebroadcast simultaneously by the nodes that have it. The data rates for retransmissions adapt so that the full cycle time is used.

The protocol itself is information-topology independent, but star-topology examples will be used to explain fixed schedules in Sec. III-B and adaptive schedules in Sec. III-C.

A. Resource Assumptions

We make a few assumptions regarding the hardware and environment to focus on the conceptual framework of the protocol.

- All the nodes share a universal addressing scheme and order. Each node knows the initial order of messages being transmitted so there is no confusion about what is transmitted next.
- 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 [60]. Thus we can schedule time slots for specific packets and nodes can simultaneously transmit if so desired.

Algorithm 1 Occupy CoW Protocol

procedure DETERMINE SCHEDULE
  $\mathcal{S} \leftarrow$ set of all nodes
  $\mathcal{G} \leftarrow$ ordered key-value pair table. Messages are the keys and lists of their subscribers are the values. Messages are transmitted as per their order in the table.
  $scheme \leftarrow$ fixed or adaptive
  $hops \leftarrow 2$ or 3
  if $scheme$ = fixed **then**
    procedure FIXED SCHEDULE
      Phase 1:
      for packet $g \in \mathcal{G}$ **do** g is broadcast in g's pre-assigned slot. All other nodes listen.
      Phase 2:
      for packet $g \in \mathcal{G}$ **do** All nodes with g simultaneously broadcast during g's pre-assigned slot using a diversity-oriented DSTC. All others listen.
      if $hops = 3$ **then**
        Phase 3:
        for packet $g \in \mathcal{G}$ **do** All nodes with g simultaneously broadcast during g's pre-assigned slot using a diversity-oriented DSTC. Interested subscribers listen.
  else
    procedure ADAPTIVE SCHEDULE
      Message Phase 1:
      for packet $g \in \mathcal{G}$ **do** g is broadcast in g's allocated slot. All other nodes listen.
      for node $s \in \mathcal{S}$ **do** $a_s \leftarrow$ ACK packet indicating the $g \in \mathcal{G}$ that s has received.
      Scheduling Phase 1:
      for node $s \in \mathcal{S}$ **do** s broadcasts a_s in its pre-assigned slot. All others listen.
      Scheduling Phase 2:
      for $s \in \mathcal{S}$ **do** All nodes with ACK packet a_s simultaneously retransmit it in its pre-assigned slot using a diversity-oriented DSTC. All others listen.
      if $hops = 3$ **then**
        Scheduling Phase 3:
        for $s \in \mathcal{S}$ **do** All nodes with ACK packet a_s simultaneously retransmit it in its pre-assigned slot using a diversity-oriented DSTC. All others listen.
      $\mathcal{G}_s \leftarrow \emptyset$. This is the new adaptive schedule to be populated and is a subset of $\mathcal{G}$.
      for packet $g \in \mathcal{G}$ **do** If g has not reached all its subscribers (as indicated by various a_s) then $\mathcal{G}_s \leftarrow \mathcal{G}_s \bigcup g$.
      Message Phase 2:
      for packet $g \in \mathcal{G}_s$ **do** All nodes with g simultaneously broadcast using a diversity-oriented DSTC during g's slot according to the new schedule. All others listen.
      if $hops = 3$ **then**
        Message Phase 3:
        for packet $g \in \mathcal{G}_s$ **do** All nodes with g simultaneously broadcast using a diversity-oriented DSTC during g's slot according to the new schedule. All interested subscribers listen.

- We assume that if k relays simultaneously (with consciously introduced jitter[3] or some other DSTC) transmit, then all receivers can extract signal diversity k without having to know in advance who is relaying or how many simultaneous transmissions they are receiving.
- (Only for adaptive-schedules) Nodes are capable of decoding variable-rate transmissions [61].

B. Fixed Schedule Example

For simplicity we focus on a simple star information topology as in Fig. 1a. A central controller (C) that must transmit m distinct bits (downlink messages) to each of the four nodes. Each of nodes (S1-S4) must transmit m distinct bits (uplink messages) to the controller. We define a cycle failure to be the event that at least one node fails to receive its downlink message, the controller fails to receive an uplink message from any of the nodes, or both. While there is no qualitative or quantitative difference between downlink and uplink packets, we use this terminology for ease of exposition.

We will now run through a fixed schedule two-hop version of Occupy CoW on this network using Fig. 3. Column (1) in the figure has two components in it — the top figure shows the available communication links depicted by the dotted lines (the rest are faded out) and the bottom comprises two tables for the downlink and uplink information of each node. The table on the left is the downlink information state of each

[3]This jitter or explicit delay transforms spatial diversity into frequency-diversity as shown in [35]. This scheme does not require a relay to know who else is relaying alongside it, and having a long enough OFDM symbol suffices.

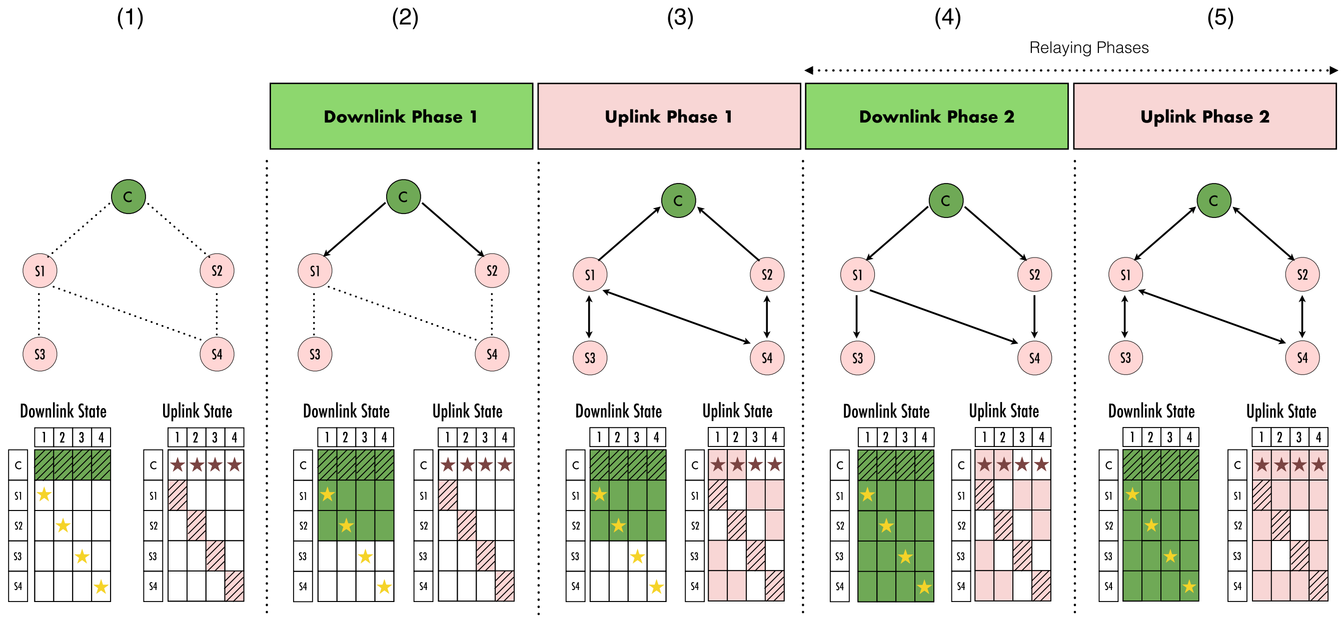


Fig. 3. Simple example with one controller and 4 nodes. The graph illustrates which links are active during that phase. The downlink and uplink tables at each stage represent the information each node has at the end of that phase. Striped cells indicate message origins and starred cells indicate message destinations. Explained in detail in Sec. III-B.

node (including the controller) and the table on the right is the uplink information state. Striped cells indicate message origins and starred cells indicate message destinations. For instance, since S1 is interested in downlink message 1 from the controller, the corresponding box in the downlink table is starred, and similarly for S2-S4. On the uplink, the controller is interested in the uplink packets from nodes S1 to S4, but these nodes do not care about each others packets, leading to stars only in the top row.

Columns (2)-(4) indicate phases of the protocol. The graph shows directional links on which information is actively transmitted during the phase. As the nodes successfully hear different packets, the cells in the table are colored in. Initially, the cells corresponding to the controller's downlink state and S1 to S4's own uplink states are filled.

Phase I: In Phase I each node transmits its messages in a predetermined order. In the schedule shown here, the controller first transmits the downlink packets for S1 through S4 in that order, and then S1 to S4 take turns transmitting their uplink packets. For illustration, we divide this Phase I into two parts: Downlink Phase I (Column (2)) and Uplink Phase I (Column (3)). Since the controller can only reach S1 and S2 the links from C $\rightarrow$ S1 and C $\rightarrow$ S2 are active (bold directed lines) and the rest remain inactive. These two nodes hear downlink messages for all four nodes as shown in the Downlink table. S1 and S2 are thus possible relays for S3 and S4's downlink messages. Then, in Uplink Phase I, S1 transmits its message and C, S3 and S4 hear the message. When S2 transmits, C and S4 are able to hear the message. When S3 transmits only S1 is able to hear the message. When S4 transmits S1 and S2 are able to hear the message. The graph illustrates these links and cells corresponding to these received messages are filled.

Phase II: In this phase (also divided into Downlink and Uplink), nodes *simultaneously* transmit packets to help other nodes[4]

In Downlink Phase II (column (4)), the first message to be relayed is the downlink packet of S1. Since C, S1 and S2 have this packet, they broadcast it using a DSTC. S3 and S4 can now decode S1's downlink packet. Similarly, S2's downlink packet is decoded by S3 and S4.

The key point here is that S3's downlink packet is retransmitted by both C and S1 using a DSTC. Since S1 has a good channel to S3 (and S4) they both can decode this. The same ensues for S4's downlink packet. At this stage, all nodes (S1 to S4) have received their downlink packets.

The final phase is Uplink Phase II (column (5)). S1's uplink packet is retransmitted by C, S1, S3 and S4 simultaneously using a DSTC and S2 is able to decode it. A similar procedure happens for S2's uplink packet and S1 and C are able to decode it. Again, S1 helps to transmit S3's uplink packet by simultaneously broadcasting it along with S3. C and S4 are able to decode the message. A similar procedure happens for S4's packet. Once S4's uplink packet has been transmitted, the round is complete. In this instance, all messages have reached their subscribers since all the starred cells are filled. Notice however that S2 and S3 never hear each others' uplink messages.

C. Adaptive Schedule Example

We again consider a star information topology for this example. There is one controller and 10 nodes (S0 - S9). The controller has m bits of information for each node and

[4]Section IV-D discusses a version of the protocol where relays take turns instead of simultaneously relaying.

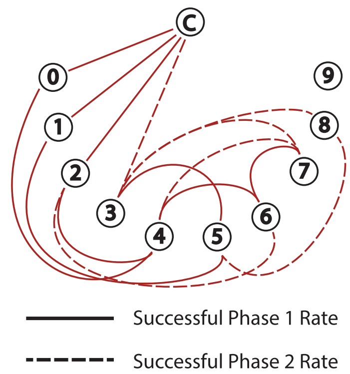


Fig. 4. Network realization of the adaptive schedule example. The links that are present under different rates are depicted.

each node in turn has m bits of information for the controller. In this example, we will consider an adaptive schedule three-hop protocol. The link realization of the network is shown in Fig. 4. The controller (C) has direct links to nodes S0 - S2 at the rate of Phase I. The rates of other phases depend on the number of nodes that succeeded in Phase I – thus links that were bad under the initial rate could be good under the new rate (for example the link between C and S4). Fig. 5 walks through this example step-by-step and shows the information state at each node. For compactness, we have merged the two uplink and downlink tables for each node into a single table. During the downlink phases, the downlink part of the table is shown and during the uplink phases, the uplink part of the table is shown. For this example, we allocate time equally for all message phases (Downlink Phases I, II and III and Uplink Phases I, II and III) and by reciprocity assume that links present in Downlink Phase I are present in Uplink Phase I and so on.

Phase I: This phase is just like its counterpart in the fixed-schedule case — all messages get transmitted for the first time in their allotted slots. Phase I is divided into two phases – Downlink Phase I (length of T_{D_1})) and Uplink Phase I (length of T_{U_1})). In Downlink Phase I, the controller transmits the downlink packets of each of the nodes. One can further optimize this by combining multiple packets from a single node into one larger packet for practical purposes (as shown in Fig. 5). The controller combines the individual messages into a single packet and broadcasts it at the rate $R_{D_1} = \frac{m \cdot n}{T_{D_1}}$. In the instance depicted in Fig. 5, Column 1, only S0, S1, and S2 successfully receive and decode the controller's packet. Note that these three "direct links" to the controller are also depicted in Fig. 4. At the end of Downlink Phase I, S0, S1, and S2 have decoded both their individual messages as well as the messages intended for all of the other nodes. This is followed by Uplink Phase I. In this phase the individual nodes transmit their uplink messages in separate packets in their assigned slots.

In Fig. 5, Column 2, again only S0, S1, and S2 successfully transmit their messages to the controller. When a node is not transmitting, it is trying to listen for other messages – thus S4 and S0 are able to hear each other, and so on. In Fig. 4, we can also see the nodes which can hear each other even though they do not have anything to say to each other. All successes thus far have been due to *direct* connections between nodes and the controller. Due to this, we refer to these types of successes as "one-hop" successes.

Scheduling Phases: The scheduling phases are the key component in the adaptive scheduling scheme, since it is essential that all the nodes are aware of the packets that require retransmission. This allows them to compute the schedule according to which relays can help using the DSTC.

During three scheduling phases (total length T_S and each sub-phase of length $T_S/3$) the controller and the other nodes transmit short acknowledgments of $2n$ bits corresponding to n downlink packets and n uplink packets. Each phase is divided equally among the $n+1$ nodes, resulting in a scheduling rate of $R_S = \frac{2n\cdot\ (n+1)}{T_S/3} = \frac{6n\cdot\ (n+1)}{T_S}$. In Scheduling Phase I, all nodes take turns transmitting their acknowledgment (ACK) packets. For example, the controller could go first, then S0, then S1 and so on. In this example, the controller's ACK packet would be 11111111111110000000 with the first 10 ones corresponding to the downlink packets (known by assumption), the next 3 ones indicate that the controller has the uplink packets of S0 - S2 and the rest of them are zero to indicate that the controller doesn't have those packets. Similarly S0's ACK packet is 11111111111000100000 with the first 10 ones corresponding to the downlink packets (as S0 has decoded all downlink packets), the next one is for its own uplink packet, the next three zeros for the uplink packets of S1 - S3 are followed by a one for S4's uplink packet and the rest are zeros corresponding to S5's - S9's uplink packets. After all ACK packets have been transmitted once, Scheduling Phase I ends.

In Scheduling Phases II and III, these short ACK packets are retransmitted in a round-robin fashion by the nodes which have heard them using a DSTC in a fashion identical to fixed-schedule Occupy CoW Phase II. For example, C's ACK packet is simultaneously transmitted by C, S0, S1, S2 and S3, S0's ACK packet is transmitted by C, S0 and S4 and so on. These ACK packets are relayed once again in Scheduling Phase III so that all packets reach all nodes. At the end of Scheduling Phase III, all nodes have 'global ACK information' with high probability and are ready to adapt the retransmission schedule so that slots are not wasted on already successful data packets in Phases II and III. Fig. 5, Column 3 shows the information state of the nodes after the end of the scheduling phases. All nodes except S9 have received every ACK packet and know the schedule for the rest of the cycle. However, S9 has not received the scheduling information and therefore does not transmit anything for the rest of the cycle in order to avoid any interference to other packets.

Phase II: After the Scheduling Phases, we have Phase II of data transmission. The messages that have already succeeded are the downlink and uplink packets of S0 - S2. Thus, the retransmission schedule only allocates time for the downlink packets of S3 - S9 and the uplink packets of S3 - S9. For illustrative purposes, we divide this phase further into two sub-phases – Downlink Phase II (length T_{D_2}) and Uplink Phase II (length T_{U_2}). In general, if a_D packets

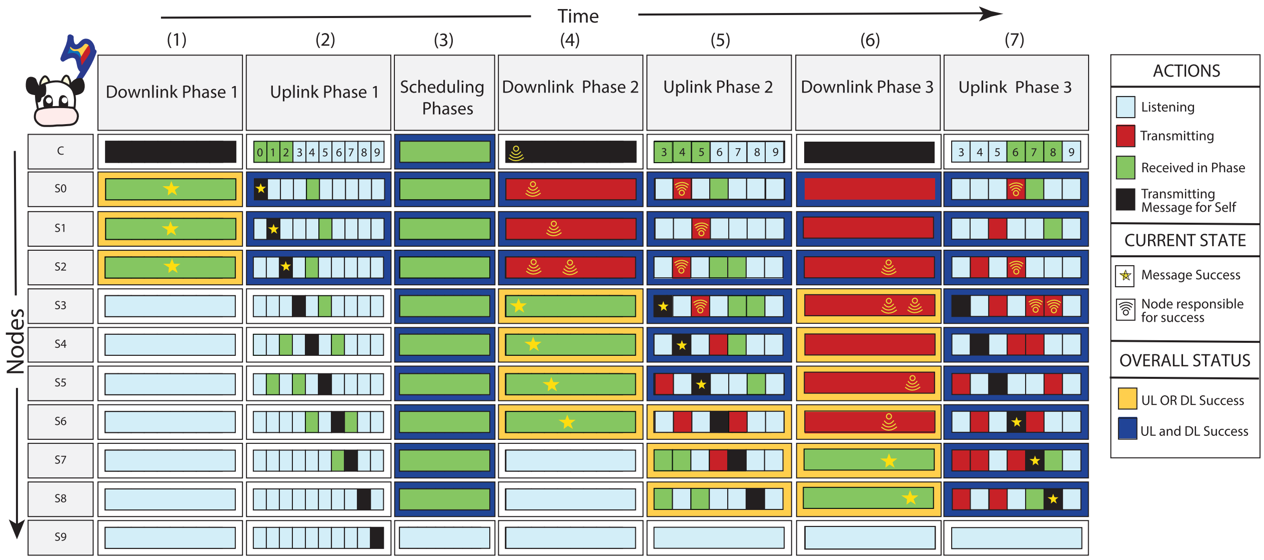


Fig. 5. The seven phases of the Occupy CoW protocol illustrated by a representative example. The table shows a variety of successful downlink and uplink transmissions using 0, 1 or 2 relays. S9 is unsuccessful for both downlink and uplink.

have succeeded in Downlink Phase I and a_U have succeeded in Uplink Phase I, then the rates of transmission in these phases are: $R_{D_2} = \frac{m \cdot (n-a_D)}{T_{D_2}}$ and $R_{U_2} = \frac{m \cdot (n-a_U)}{T_{U_2}}$. The relaying in these phases is the same as the relaying in Phase II of the fixed schedule protocol – except with a modified schedule. Because the schedule has adapted, it is possible that nodes that were initially unable to directly connect to the controller may now be able to, *if* the rate during any of these phases is lower than that of the first. This may occur if enough nodes are successful in the first phase since fewer messages must now be sent or if the time allocated for the phases T_{D_2} or T_{U_2} is greater than T_{D_1} or T_{U_1} respectively resulting in a lower rate.

Downlink Phase II is depicted in Fig. 5, Column 4. We see that node S3 gets its downlink message directly through the controller (due to reduced rate), and this is reflected in the dashed representation of the connection between node S3 and the controller in Fig. 4. As S0 and S2 are able to reach S4, it successfully receives the controller's message in two hops via S0 and S2 and so on. At the end of Downlink Phase II, nodes S0, S1, S2, S3, S4, S5, and S6 have successfully received their downlink messages. Uplink Phase II is similar and is depicted in Fig. 5, Column 5 and the same set of nodes' uplink packets have successfully been delivered to the controller.

Phase III: Again, this phase is divided into Downlink Phase III (length T_{D_3}) and Uplink Phase III (length T_{U_3}) with rates $R_{D_3} = \frac{m \cdot (n-a_D)}{T_{D_3}}$ and $R_{U_3} = \frac{m \cdot (n-a_U)}{T_{U_3}}$ respectively. In these phases, 'three-hop' successes occur. For example, in Fig. 5, Column 6, S8 successfully receives its downlink packet through S5 (the full path is C $\rightarrow$ S1 $\rightarrow$ S5 $\rightarrow$ S8). The uplink counterpart is similar to downlink and at the end of Phase III, all nodes except S9 have received their downlink packet and have successfully relayed their uplink message to the controller. The example depicted in Fig. 4 and 5 is a failed instance of the protocol since the node S9 has not received its downlink message and the controller has not received S9's uplink packet.

D. Information Topology-Dependent Optimization

The adaptive schedule scheme can be optimized for reduced implementation complexity when the information topology is a star. In particular, the scheduling phase can be shortened.

For example, each node can piggyback a one bit ACK for their downlink packet onto their uplink message. Then the extra scheduling phases can be simplified to a single phase where the controller processes all the ACKs (received as well as not received) into a single packet that just lists which messages require retransmission. Then, all the nodes that can hear the controller get to know the schedule. These nodes can then modify the downlink packet (culling already successful messages and appending the global schedule to it) and simultaneously broadcast it. The nodes that can hear this first set of relays can then not only decode the downlink messages (despite not knowing the schedule) but also figure out the schedule itself so that they can help in the next phase. At this stage, nodes only reachable via three hops do not have the schedule and to propagate the information to them, we switch the order of Uplink Phase II and Downlink Phase III (Downlink Phase III directly follows Downlink Phase II). The nodes reachable by two hops broadcast the downlink messages (with embedded schedule) again so that it can be heard by the nodes only reachable by three hops. Thus, even though all the nodes did not know the schedule at the beginning of Downlink Phase II, they do get to know it by the end of Downlink Phase III and that is sufficient for enabling Uplink Phases II and III. As you can see, this optimization exploits the star nature of the information topology to shorten the scheduling phase and furthermore, the total traffic dedicated to scheduling is substantially reduced. This optimized variation of the protocol is explored in detail in [4].

IV. Analysis of Occupy CoW

We explore the Occupy CoW protocol with parameters in the neighborhood of a practical application, the industrial printer case described in Weiner et al. [2]. In this particular scenario, the SERCOS III protocol [62] supports the printer's required cycle time of 2 ms with reliability of 10^{-8}. Consequently, we target a 10^{-9} probability of failure for Occupy CoW. The printer has 30 moving printing heads that move at speeds up to 3 m/s over distances of up to 10 m. Every 2 ms cycle, each head's actuator receives 20 bytes from the controller and each head's sensor transmits 20 bytes to the controller. The amount of information transmitted by the controller in a single cycle is $20 \times 30 = 600$ bytes. The total amount of sensor information transmitted by the heads to the controller in a single cycle is also $20 \times 30 = 600$ bytes. Therefore a total of 1200 bytes or 9600 bits of information is sent during a cycle of 2 ms which corresponds to a desired goodput of $\frac{9600}{(2\times10^{-3})}$ bit/sec = 4.8 Mbit/sec. If we assume access to a single dedicated 20 MHz wireless channel, this 4.8 Mbit/sec corresponds to an overall net spectral efficiency of $\frac{(4.8\times10^6)}{(20\times10^6)} = 0.24$ bits/sec/Hz.

A. Behavioral Assumptions for Analysis

The following behavioral assumptions are added to the resource assumptions in Sec. III-A.

- We assume a fixed nominal SNR and independent Rayleigh fading on each link. We assume that each node has perfect local receiver CSI knowledge of the 'good' channels to itself i.e., those net channels on which messages may be decoded. No assumptions are made for knowing the CSI of deeply faded channels. No global CSI knowledge is assumed. We defer the issue of the overhead associated with acquiring this local knowledge to future work.
- We assume a single tap channel — performance would improve if we reliably had more taps/diversity. Because the cycle-time is so short, the channels' coefficients do not change in a cycle and hence we use the delay-limited-capacity framework [63], [64].
- A link with complex fade h and bandwidth W is deemed good (no errors or erasures) if the rate of transmission R is less than or equal to the link's capacity $C = W\log(1 + |h|^2 SNR)$. Consequently, the probability of link failure is defined as

$$p_{link} = P(R > C) = 1 - \exp\left(-\frac{2^{R/W} - 1}{SNR}\right). \quad (1)$$

  From the above equation we see that if R decreases, then the probability that the capacity C is less than R also decreases (the capacity C did not change, only R did). In other words, a channel which was unable to support a given rate might be able to support a lower rate.
- We also assume channel reciprocity – if a channel has fade h between node A to B, then it is also h from B to A as well.
- If there are k simultaneous transmissions,[5] then each receiving node harvests perfect sender diversity of k. For analysis purposes this is treated as k independent tries for communicating the message that only fails if all the tries fail.
- We do not consider any real implementation effects on decoding to abstract away some of the nuanced effects. This is partially justified in spirit by Yang et al. [66] where they show that the channel dispersion is zero for quasi-static fading channels. However, we have explored finite-blocklength effects in another work [67]. At the most basic level, we can think about a code in terms of its gap to capacity at the desired reliability. If a code is 3dB away from capacity, then one can add 3dB to all power requirements calculated assuming infinite-blocklength and the protocol should work. However, the effects are more nuanced. We found in [67] that the demands are different in different phases of a diversity-seeking cooperative protocol. For infinite-blocklengths, codes either work or don't work depending on the rate and channel capacity. In the case of finite-blocklength the performance degrades more smoothly. Even a link that cannot deliver the final target reliability for the error correcting code under consideration still might be enough to allow a node to decode and become a potential relay. More potential relays mean that we can more easily count on getting higher receiver power in the relaying phase – thus getting higher reliability in the relaying phase. This tradeoff creates "partial credit" which allows us to perform better than what a conservative Shannon-capacity-plus-gap-based analysis would suggest. We are unable to include these effects in this paper in further detail due to page limitations. A related assumption is that no transmission or decoding errors are undetected [68] — a corrupted packet can be identified (say using a 40 bit hash) and is then completely discarded.

We derive the probability of failure for a two-hop downlink scheme in Sec. IV-B, a union bound of the failure probability for a generic information topology in Sec. IV-C, a similar bound on the probability of failure for relaying with non-simultaneous transmissions in Sec. IV-D and the probability of failure for frequency hopping repetition coding in Sec. IV-E. These equations are used to derive results in Sec. IV-F. Additional derivations can be found in [5].

B. Two-Hop Downlink (Star Information Topology)

In a two-hop scheme, there are two shots at getting a message across. We derive the probability of protocol failure for both the fixed and adaptive schedule scheme. In both schemes, failure is the event that at least one of the n nodes in the set $\mathcal{S}$ has not received its message by the end.

1) Fixed Schedule Scheme: In the fixed schedule two-hop scheme each message gets sent twice whether or not it was successful in the first try. The first phase's rate $R_{D_1} = \frac{n \cdot m}{T_{D_1}}$ and the corresponding probability of link failure is p_1. The second phase rate $R_{D_2} = \frac{n \cdot m}{T_{D_2}}$ (as all messages get sent two times) and the corresponding probability of link failure is p_2. Let the nodes successful in Phase I be in a set $\mathcal{A}$ (with cardinality

[5] The cyclic-delay-diversity space-time-coding schemes we envision make the effective channel response longer. This can push the PHY into the "wideband regime", and a full analysis must account for the required increase in channel sounding by pilots to learn this channel [65]. We defer this issue to future work but preliminary results suggest that it will only add $2-3$dB to the SNRs required at reasonable network sizes.

represented by the random variable A and a representing a specific size). The nodes in the set $\mathcal{S} \setminus \mathcal{A}$ succeed in Phase II only if they connect to either the controller or at least one of the nodes in $\mathcal{A}$. The Rayleigh fading assumption tells us that the probability that a link fails in Phase II given it failed in Phase I is given by $p_c = \min\left(\frac{p_2}{p_1}, 1\right)$. Then, the probability of not connecting to {controller $\bigcup \mathcal{A}$} in Phase II is $p_2^a \cdot p_c$. The probability of 2-phase downlink system failure is thus:

$$\mathrm{P(fail)} = \sum_{a=0}^{n-1} \left\{ \left(\binom{n}{a}(1-p_1)^a (p_1)^{n-a}\right) \times \left(1 - \left(1 - p_2^a \cdot p_c\right)^{n-a}\right) \right\}. \quad (2)$$

2) Adaptive Schedule Scheme: In the adaptive schedule two-hop scheme only the messages that were unsuccessful in Phase I get sent again in Phase II. The first phase is exactly like the fixed-schedule scheme. The time allocated for Phase II and the number of first phase successes a dictate the Phase II rate $R_{D_2} = \frac{(n-a)\cdot m}{T_{D_2}}$. The corresponding probability of link failure is denoted $p_2(a)$ (the (a) is used to indicate that it is a function of a). As in the fixed-schedule case, the probability that the controller to node link fails in Phase II given it failed in Phase I is given by $p_c(a) = \min\left(\frac{p_2(a)}{p_1}, 1\right)$. Then, the probability of not connecting to {controller $\bigcup \mathcal{A}$} in Phase II is $(p_2(a))^a \cdot p_c(a)$. The probability of 2-hop downlink system failure is thus:

$$\mathrm{P(fail)} = \sum_{a=0}^{n-1} \left\{ \left(\binom{n}{a}(1-p_1)^a (p_1)^{n-a}\right) \times \left(1 - \left(1 - (p_2(a))^a \cdot p_c(a)\right)^{n-a}\right) \right\}. \quad (3)$$

Notice that in the above derivation, we omitted the role of scheduling information even though it is actually crucial for adapting the rate of transmission in Phase II. This is because we assume that the scheduling phases are allocated sufficient time such that the scheduling phase rate is lower than the rates of transmission in any of the other phases. Therefore any scheduling error in the protocol is also going to manifest as a delivery failure for a message packet. A property of ACK packets is that they want to reach all the nodes in the network. Therefore, a more detailed analysis for scheduling failure is as derived in the next subsection discussing the union bound, which is how we can upper bound the probability of failure for a generic topology (Sec. IV-C).

C. The Union Bound and Generic Information Topologies

Consider a generic network with n nodes and s message streams. Let's say that each stream has one origin and on average d subscribers. For simplicity, the rates for all transmissions are kept constant at some rate R with a corresponding probability p of link failure. Consider a single message-destination pair. Let each message get two shots at reaching its subscribers – directly from the source or through relays (say j of them). Then the probability of the message reaching any specific destination is

$$\begin{aligned} q_s &= \mathrm{P(direct\ link)} \times \mathrm{P(success|direct\ link)} \\ &\quad + \mathrm{P(no\ direct\ link)} \times \mathrm{P(success|no\ direct\ link)} \\ &= (1-p) + p\sum_{j=1}^{n-2}\binom{n-2}{j}(1-p)^j p^{n-2-j}\left(1-p^j\right) \quad (4) \end{aligned}$$

Then the union bound on the probability of failure that even one of the s messages did not reach one of its subscribers is:

$$\mathrm{P(failure)} = s \times d \times (1 - q_s). \quad (5)$$

As mentioned in Sec. IV-B2, ACK information from each node has to disseminate throughout the network – therefore if there are n nodes in the network, there are n ACK messages, and the number of subscribers for each is $n-1$. Consequently, the probability of ACK dissemination failure can be bounded by the union bound derived in this section. Equations for other error probabilities are derived similarly and can be found in the extended version of this paper [5].

D. Non-Simultaneous Relaying

To tease apart the impacts of relaying and simultaneous transmission, it is useful to analyze relaying without simultaneous transmissions. To have relays taking turns within a fixed-schedule scheme, the basic requirement is making the time-slots shorter. Suppose that we have r potential relays for every data packet, designated in advance for every message stream. This means that if there are k (either 2 or 3) hops, then each data packet will have a footprint of $1 + (k-1)r$ time slots. This means that for s message streams, if the total cycle time is T and each data packet is m bits long, then the link data rate is $R = \frac{s\cdot m\cdot\ (1+(k-1)r)}{T}$.

Consider a single message stream and let $q_s(p, r)$ denote the probability of success to a single destination where p is the probability of link error given the rate and SNR. The analysis for the two-hop case follows the union-bound case in (4) with the number of potential relays r playing the role of $n-2$ above. Consequently:

$$q_s(p,r) = (1-p) + \left(p \cdot \left(\sum_{j=1}^{r}\binom{r}{j}(1-p)^j p^{r-j}(1-p^j)\right)\right). \quad (6)$$

The union bound argument applies and so (5) continues to bound the probability of error, just with the slightly revised expression in (6) for q_s.

E. Frequency-Hopping Schemes

In Occupy CoW, during each message's transmission, the entire available bandwidth W is used for coding at a link rate of R. In a frequency-hopping scheme, the available bandwidth W is broken into k sub-channels ($k > 1$) and each sub-channel carries the entire packet at the higher rate of $R_{sc}(k) = k \times R$. We assume that each sub-channel fades independently. The analysis of the frequency hopping repetition coding scheme is very similar to the non-simultaneous relaying scheme. Let the probability of failure of a single sub-channel link at rate $R_{sc}(k)$ be $p_{sc}(k)$. Then the probability that a message was not successful is the probability that each of the sub-channels failed to deliver the message i.e., $(p_{sc}(k))^k$. Therefore, the failure probability of a frequency hopping repetition based scheme with s streams each with a single destination is given by

$$\mathrm{P}(\text{fail}, k) = 1 - \left(1 - (p_{sc}(k))^k\right)^s. \quad (7)$$

F. Results and Comparison

Following Weiner et al. [2] and the communication-theoretic convention, we use the minimum SNR required to achieve 10^{-9} reliability as our metric to compare fixed-schedule 2-hop and adaptive-schedule 3-hop Occupy CoW to four other baseline schemes. We calculate the minimum SNR required by various protocols to meet the specs in the following fashion. Assuming a fixed nominal SNR, we calculate the probability of failure for the protocol under consideration. Then, we search for the smallest value of nominal SNR that meets the reliability requirement of 10^{-9}. Fig. 2 looks at performance (the minimum SNR required on the y-axis) for a star information topology with a central node sending $m = 160$ bit messages to n other nodes and receiving the same size messages from them. All this has to be completed within 2ms and a bandwidth of 20MHz. Initially the minimum required SNR for Occupy CoW decreases with increasing n, even through the required throughput increases as $m \cdot n$, but the curves then flatten out. The gains of multi-user diversity eventually give way and the required SNR starts to increase for large n as the required spectral efficiency increases.

The topmost blue solid curve in Fig. 2 shows performance of the protocol restricted to just the first hop of Occupy CoW with one slot per message. The required SNR shoots off the figure for two reasons: (a) because the throughput increases linearly with the number of nodes and (b) to have the system probability of failure stay controlled with more messages to transmit, each individual message must be that much more reliable. The second scheme (red dashed curve) is purely hypothetical. It allows each message to use the entire 2ms time slot for its own uplink and downlink message but without any relaying and thus also no diversity. This bounds what could possibly be achieved by using adaptive HARQ techniques and shows why harnessing diversity is essential. This is rising only because of effect (b) above.

The third reference scheme is the non-simultaneous relaying scheme described in Sec. IV-D and plotted in Fig. 2 by the black curve with markers. We see that this curve is always above the Occupy CoW lines — showing the quantitative importance of simultaneous relaying. The curve is annotated with the best number of relays r that minimizes the SNR required. As r increases, the available spatial diversity increases, but the added message repetitions force the link data rates higher.

Fig. 6 explores the effect of the number of relays allocated on the required SNR for the scheme in Sec. IV-D. For a network size of $n = 30$, a payload size of 60B per message would select $r = 6$ as the optimal number of relays. Reducing

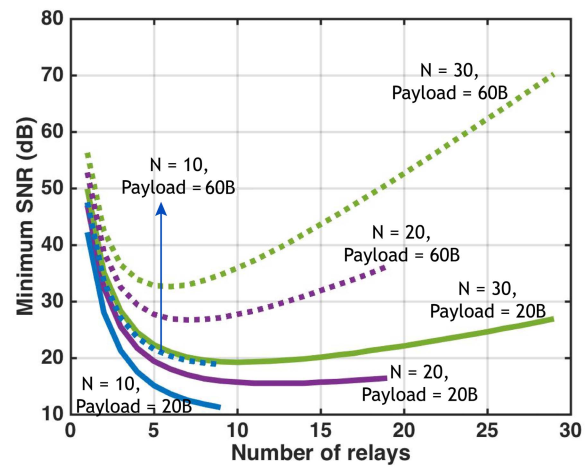


Fig. 6. For non-simultaneous relaying, the minimum SNR required to achieve a 10^{-9} probability of system failure for different network and payload sizes as the number of nominated relays vary.

the network size to 10 makes $r = 9$ be the optimal number of relays. Compare this to a payload size of 20B and $n = 30$ — not only is the optimal number of relays the same, the entire curve is very close to that for $n = 10$ with payload $60B$.

This is because for the same number of relays, the link data rates are the same and the factor of 3 difference in the number of message streams demands a factor 3 reduction in the probability of error per message – which for nine relays is accomplished for less than 1dB. Given a large enough network, the optimum number of relays seems to depend primarily on the aggregate rate. For a high-aggregate rate, we choose a smaller number of relays and for a lower one, we pick more relays.

The last reference scheme (the purple dotted line in Fig. 2) represents the hypothetical frequency-hopping described in Sec. IV-E. As the number k of frequency hops increases, the available diversity increases, but the added message repetitions force the instantaneous link rates higher, just as additional relays do for non-simultaneous relaying. For low n we prefer more frequency hops because of the diversity benefits. The SNR cost of doing this is not so high because the throughput is low enough (requiring a spectral efficiency less than 1.5bits/s/Hz) that we are still on the cusp of the energy-limited 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 20 or more guaranteed independently faded sub-channels to repeat across. After 7 nodes, notice the frequency-hopping scheme is paralleling the non-simultaneous relaying scheme in Fig. 2. However, frequency hopping is optimized with more diversity and lower SNR because harnessing multiuser diversity requires the first-hop to actually reach enough relays to be able to use the reserved slots while frequency-diversity is just assumed to always be available. Fig. 2 also compares the fixed-schedule two-hop Occupy CoW protocol with equal phase lengths to an adaptive three-hop scheme optimized to minimize SNR. We see that these are very close to each other

Aggregate Rate (b/s/Hz) \ Number of slave nodes (n)	1	2	3	4	5	10	15	20	25	30
3	96	65	49.5	42.5	38	29	25.5	24	22.5	22
2	93.5	59	44	36.5	32	23	19.5	18	16.5	16
1	89.5	52.5	37.5	30	25.5	16.5	13	11	10	9
0.75	88	50.5	35	28	23.5	14.5	11	9	8	7
0.5	86	48	33	25.5	21	12	8.5	6.5	5.5	4.5
0.25	83	44	29.5	22	17	7.5	4	2.5	1	0.5
0.1	78.5	39.5	25	17	12.5	3	-0.5	-2.5	-3.5	-4.5
0.05	75.5	36.5	22	14	9	-0.5	-3.5	-5.5	-7	-7.5
0.01	68.5	29.5	14.5	6.5	2	-7.5	-11	-13	-14	-15

3 hops
2 hops
1 hop

Fig. 7. The number of hops and minimum SNR to be operating at to achieve a high-performance of 10^{-9} as aggregate rate and number of users are varied. Here, the time division within a cycle is unoptimized. Uplink and downlink have equal time, 2-hops has a 1:1 ratio across phases, and 3-hops has a 1:1:1 ratio for the 3 phases. The numbers here are for a star information topology but as the next figure shows, they would not be much different for generic topologies.

and the choice between these is not as important as harnessing diversity and taking advantage of simultaneous transmissions. This is discussed in detail in the next Section V.

It turns out that the aggregate goodput required (overall spectral efficiency considering all users) is the most important parameter for choosing the number of relay hops in our scheme. This is illustrated clearly in Fig. 7. This table shows the SNR required and the best number of hops to use for a given n. With one node, clearly a 1 phase scheme is all that is possible. As the number of nodes increases, we transition from 2-phase to 3-phase schemes being better. For $n \geqslant 5$, aggregate rate is what matters in choosing a scheme, since 3-phase schemes have to deal with a $3\times$ increase in the instantaneous rate due to each phases' shorter time, and this dominates the choice. In principle, at high enough aggregate rates, even the one-hop scheme will be best with enough users. But when the target reliability is 10^{-9}, this is at absurdly high aggregate rates.[6] In the practical regime, diversity wins.

We now consider the case of a generic non-star topology using (5). Figure 8 considers the SNR required for a varying number of destinations for different network sizes. The number of destinations per message ranges from 1 to $n-1$. For comparison purposes, at "0" destinations, we have plotted the SNR required for the star information topology. There is an SNR 'penalty' for each message having multiple destinations but even when everyone wants to hear everything, this penalty is quite modest. The case of $n-1$ destinations is similar to simply reducing the tolerable probability of failure by a factor of $1/n$. The extra SNR required is on the order of 1dB for medium to large network sizes because of the ample diversity available.

V. Optimization of Occupy CoW

A. Phase-Length Optimization

The protocols we have described come with the choice of the number of phases (2 or 3) and a choice of fixed or adaptive schedule. Furthermore, there is the choice of the time

[6]We estimate this is around aggregate rate 40 — that would correspond to 40 users each of which wants to simultaneously achieve a spectral efficiency of 1.

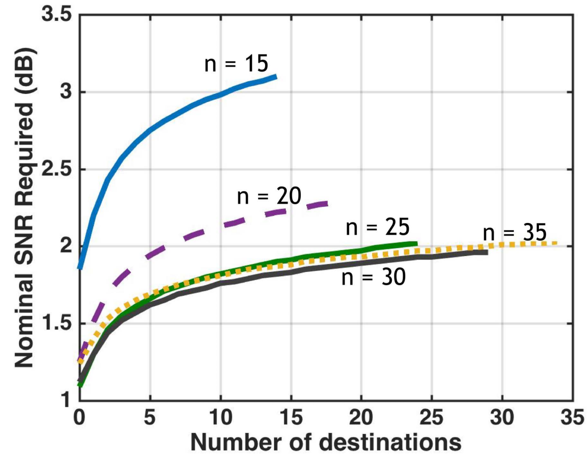


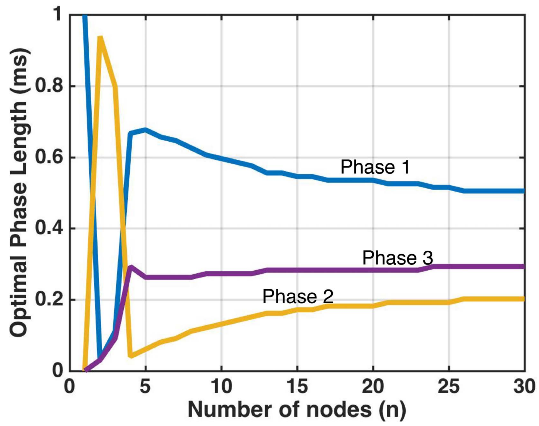
Fig. 8. Number of destinations vs SNR required for different network sizes for $m = 160$ bit messages and $n = (15, 20, 25, 30, 35)$ nodes with 20MHz and a 2ms cycle time, aiming at 10^{-9} probability of failure. The SNR values at "0" destinations represents the SNR required for a star information topology.

allocated for different phases. How does one pick the 'right' parameters? Does it matter? To answer that, we compare the performance (minimum SNR required to achieve the specs) of a simple 2-hop fixed schedule scheme where the time available is equally divided among the phases and a 3-hop adaptive protocol with optimal phase lengths minimizing the SNR required.

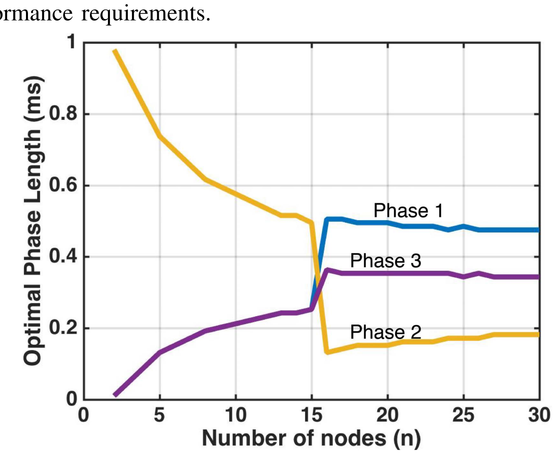
We focus on a star information topology because it has both extremes of downlink (one source with separate messages for many destinations) and uplink (the vice-versa). We consider downlink and uplink separately and look at the optimal allocation of time for a three-hop protocol which minimizes the SNR required to meet the performance specifications. Here we used a simple brute force search over time allocations. We find that the optimal phase-length allocations are far from even. We also find that the SNR savings that we achieve by having different lengths is minimal and believe that the implementation complexity of building a system which can code and decode at variable rates is a bigger deal and ultimately negates out the small SNR savings achieved by phase-length optimization and dealing with all the ACK information.

Let us consider the adaptive scheduling protocol with a 2ms cycle divided equally between Uplink and Downlink. How should we divide the times across phases for this? Assume that the ACK information is reliably delivered for free. Let the times allocated for Phase I, II and III of downlink and uplink be T_{D_1}, T_{D_2} and T_{D_3} and T_{U_1}, T_{U_2} and T_{U_3}respectively such that $T_{D_1} + T_{D_2} + T_{D_3} = 1$ms and $T_{U_1} + T_{U_2} + T_{U_3} = 1$ms. Similarly, let the times allocated for phase I, II and III of uplink be T_{U_1}, T_{U_2} and T_{U_3} respectively such that $T_{U_1} + T_{U_2} + T_{U_3} = 1$ms.

1) Downlink: Figure 9a shows the optimal allocation of time for phase I, II and III for downlink. The optimization suggests that phase I should be the longest, phase II the shortest and phase III in between (except for network sizes 1 and 2 where the optimal strategies are 1 hop and 2 hop respectively). Phase III is longer than Phase II to make sure that the messages reach everyone possible as more links



(a) Optimal fraction of time allocated for downlink phase I, II and III in the three-hop protocol at the smallest SNR which meets the performance requirements.



(b) Optimal fraction of time allocated for uplink phase I, II and III in the three-hop protocol at the smallest SNR which meets the performance requirements.

Fig. 9. Optimal phase allocation for three-hops with 160 bit messages, 30 users, 2×10^4 total cycle length.

open up during phase III. Phase I is longest to ensure that the messages are initially successfully decoded by enough nodes to ensure maximal spread.

2) Uplink: Figure 9b shows the optimal allocation of time for phase I, II and III for uplink. In uplink, the critical paths are the ones connecting to the controller rather than the inter-node links. Hence, phase III is allocated more time than in the downlink Phase.

How much SNR does optimization save?

For concreteness, let us consider the downlink side. (Uplink is similar.) Figure 10 considers different phase length allocations including the optimal phase length allocation and several other suboptimal allocations. For a star network of 30 nodes, we see that the difference between the various allocations and schemes is minimal. Note that these results are for a star information topology which is the 'best' case in terms of the SNR required. As the benefits of optimization are marginal in the best case, the benefits in a generic topology are even more negligible.

Furthermore, adaptive scheduling is a harder problem in a non-star topology as one cannot mostly piggyback the relevant ACK or scheduling information onto packets that would be sent anyway, as discussed in Section III-D. Consequently, we conclude that though we have many knobs to turn which can optimize the performance of the protocol in terms of required SNR, the benefits for that metric are not going to be that substantial. This is not to say that there might not be other reasons for wanting to use adaptive scheduling — e.g. to support additional best-effort traffic by harvesting time-slots that are not needed for relaying time-critical packets. However, that is beyond the scope of this paper.

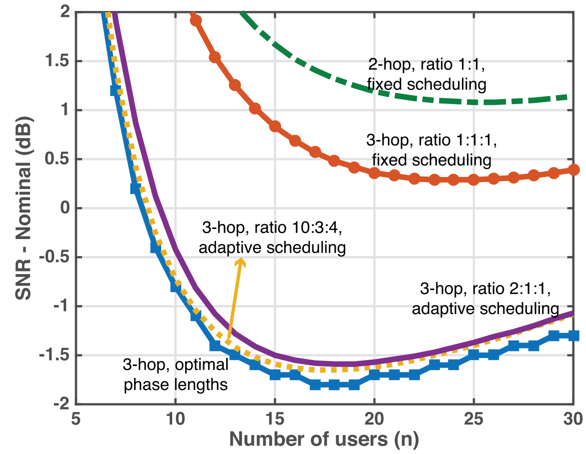


Fig. 10. Comparing the SNR required for optimum downlink phase length allocation and a few non-optimal allocations.

B. Power Consumption and the Effect of Duty Cycling

The Occupy CoW protocol as described so far relies on all nodes being awake and listening at all times (when not transmitting). However, in most practical wireless systems, nodes are asleep often to conserve energy, even during active periods. If such duty-cycling is to be introduced, what percentage of time should the nodes be put to sleep? To answer this question, we first modify the protocol to handle duty-cycling by using the ideas used to understand non-simultaneous relaying.

We dedicate a percentage of nodes per message as pre-allocated potential relays (say $x\%$). These wake up during the message's transmission – they either listen for the message or simultaneously re-transmit the message if they have it. The equation (4) can be modified so that the maximum number of relays is not $n-2$ but $r = \left\lceil \frac{x \times (n-2)}{100} \right\rceil$. Thus the probability of success of a single message-destination pair $q_{ds}(r)$ is:

$$q_{ds}(r) = (1-p) + \left(p \cdot \left(\sum_{j=1}^{r} \binom{r}{j} (1-p)^j p^{r-j} (1-p^j) \right) \right) \tag{8}$$

Thus we have that the duty-cycled protocol's probability failure with s message streams and d average subscribers per stream is bounded by

$$\text{P(failure)} \leq s \times d \times (1 - q_{ds}(r)). \tag{9}$$

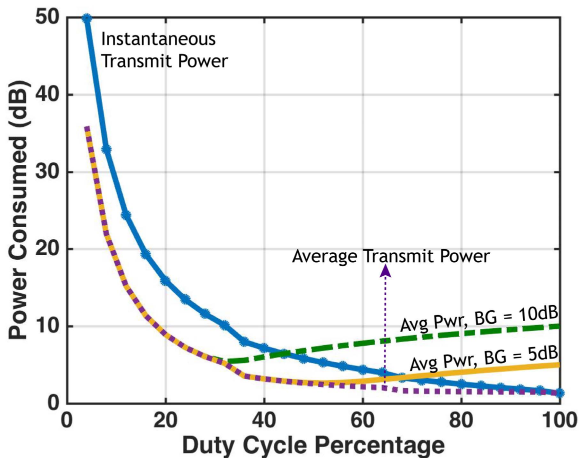


Fig. 11. Effect of duty cycling percentage (i.e. time awake) on the power required for different on-time percentages for $m = 160$ bit messages and $n = 30$ nodes with 20MHz and a 2ms cycle time, aiming at 10^{-9} probability of failure.

Fig. 11 shows the power consumed to reach the target reliability as a function of the time awake (duty cycle percentage). The blue curve plots the power consumed by a node when awake (in units of received SNR) in order to meet the required reliability. The purple dotted line takes into account the percentage of time the node is asleep, and plots the average transmit power used. Because this is minimized at 100% duty cycle, if transmit power consumption were all that mattered, it would not be worth going to sleep at all. To get a more refined answer, we recognize that there is some level of background power consumption in the wireless circuitry which accounts for listening and encoding/decoding processes whenever the node is awake [69]. The green line is the average total power consumed assuming a background power consumption of 10dB (i.e. the background power is the same as what the transmit power would be to give a 10dB SNR.) and the yellow line is for 5dB background power consumption. These plots reveal an easy rule of thumb about the desired operating point – operate with a duty-cycle percentage such that the transmit power required is equal to the background power.

VI. Conclusions & Future Work

This paper proposed a wireless communication protocol framework for high-performance industrial-automation systems that demand ultra-high reliability and low-latency for many message streams within a network with many active nodes. The protocol framework is targeted to a single wireless local domain where all nodes are nominally in range of each other, but can handle any arbitrary information topology in terms of which node is subscribed to which message stream. Harnessing significant diversity is absolutely essential for ultra-reliability and cooperative communication using relaying can access multiuser diversity. To achieve low-latency, simultaneous transmission using a diversity-oriented distributed space-time code is important, especially when the payload sizes are such that spectral efficiency is a concern. This gives a significant SNR advantage over pure frequency-hopping approaches while also not demanding that nature guarantee a lot of frequency diversity. Time diversity is also not viable when the tolerable latency is shorter than the coherence time, leaving multiuser diversity as the only real choice. When the background power used for having the wireless subsystem turned on is significant, it is beneficial to have subsets of nodes go to sleep while relying on others to listen and relay messages. Although this increases the transmit power required, it reduces overall network power consumption. Simple phase length allocations and a fixed schedule are suffice to achieve our target reliability are reasonable SNR; optimized scheduling and phase lengths only provide marginal savings.

Acknowledgment

The authors would like to Venkat Anantharam for the useful discussions. The authors would also like to thank the BLISS and BWRC students, staff, faculty, and industrial sponsors.

References

[1] G. P. Fettweis, "The tactile Internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[2] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolié, "Design of a low-latency, high-reliability wireless communication system for control applications," in *Proc. IEEE Int. Conf. Commun. (ICC)* Sydney, NSW, Australia, Jun. 2014, pp. 3829–3835.

[3] *SERCOS News, the Automation Bus Magazine*. Accessed on Jan. 2014. [Online]. Available: http://www.sercos.com/literature/pdf/sercos_news_0114_en.pdf

[4] V. Narasimha Swamy *et al.*, "Cooperative communication for high-reliability low-latency wireless control," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 4380–4386.

[5] V. Narasimha Swamy *et al. Appendix to Real-Time Cooperative Communication for Automation Over Wireless*. Accessed on Jan. 2017. [Online]. Available: https://www.eecs.berkeley.edu/~vasuki/papers/OccupyFullAppendix.pdf

[6] R. Zurawski, *Industrial Communication Technology Handbook*. Boca Raton, FL, USA: CRC Press, 2005.

[7] S. K. Sen, *Fieldbus and Networking in Process Automation*. Boca Raton, FL, USA: CRC Press, 2014.

[8] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proc. IEEE*, vol. 93, no. 6, pp. 1130–1151, Jun. 2005.

[9] P. Zand, S. Chatterjea, K. Das, and P. J. M. Havinga, "Wireless industrial monitoring and control networks: The journey so far and the road ahead," *J. Sensor Actuator Netw.*, vol. 1, no. 2, pp. 123–152, 2012.

[10] A. Willig, "An architecture for wireless extension of PROFIBUS," in *Proc. 29th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, vol. 3. Nov. 2003, pp. 2369–2375.

[11] P. Morel, A. Croisier, and J.-D. Decotignie, "Requirements for wireless extensions of a FIP fieldbus," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom. (EFTA)*, vol. 1. Nov. 1996, pp. 116–122.

[12] G. Cena, A. Valenzano, and S. Vitturi, "Hybrid wired/wireless networks for real-time communications," *IEEE Ind. Electron. Mag.*, vol. 2, no. 1, pp. 8–20, Mar. 2008.

[13] *Basics of Setting Up an Industrial Wireless LAN, Version 3*, Siemens AG, Munich, Germany, 2013.

[14] H. Trsek, "Isochronous wireless network for real-time communication," in *Industrial Automation* (Technologien für die intelligente Automation). Berlin, Germany: Springer-Verlag, 2016.

[15] A. Frotzscher *et al.*, "Requirements and current solutions of wireless communication in industrial automation," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2014, pp. 67–72.

[16] D. K. Lam *et al.*, "A novel wireless LAN protocol for factory automation control," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2014, pp. 1248–1253.

[17] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[18] A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, and A. Sangiovanni-Vincentelli, "System level design for clustered wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 3, no. 3, pp. 202–214, Aug. 2007.

[19] M. A. Yigitel, O. D. Incel, and C. Ersoy, "QoS-aware MAC protocols for wireless sensor networks: A survey," *Comput. Netw.*, vol. 55, no. 8, pp. 1982–2004, Jun. 2011.

[20] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Trans. Ind. Informat.*, vol. 4, no. 2, pp. 102–124, May 2007.

[21] G. Scheible, D. Dzung, J. Endresen, and J. E. Frey, "Unplugged but connected [Design and implementation of a truly wireless real-time sensor/actuator interface]," *IEEE Ind. Electron. Mag.*, vol. 1, no. 2, pp. 25–34, Jul. 2007

[22] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[23] *ZigBee PRO Specfication*, ZigBee Alliance Standard, Davis, CA, USA, Oct. 2007.

[24] A. N. Kim, F. Hekland, S. Petersen, and P. Doyle, "When HART goes wireless: Understanding and implementing the WirelessHART standard," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Autom.*, Sep. 2008, pp. 899–907.

[25] *An update on the Process Automation Applications Wireless Standard*, document ISA100, ISA100.11a, ISA Seminar, Orlando, FL, USA, 2008.

[26] *Industrial Communication Networks-Fieldbus Specifications, WirelessHART Communication Network and Communication Profile*, Int. Electrotech. Commission, British Standards Inst., London, U.K., 2009.

[27] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Proc. 9th IEEE Int. Conf. Ind. Inform.*, Jul. 2011, pp. 410–415.

[28] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[29] A. Willig, "How to exploit spatial diversity in wireless industrial networks," *Annu. Rev. Control*, vol. 32, no. 1, pp. 49–57, Apr. 2008.

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

[31] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. Part I. System description," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.

[32] S. Girs, E. Uhlemann, and M. Bjorkman, "Increased reliability or reduced delay in wireless industrial networks using relaying and Luby codes," in *Proc. IEEE 18th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2013, pp. 1–9.

[33] F. Oggier, G. Rekaya, J.-C. Belfiore, and E. Viterbo, "Perfect space–Time block codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3885–3902, Sep. 2006.

[34] P. Elia, B. A. Sethuraman, and P. V. Kumar, "Perfect space–time codes for any number of antennas," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 3853–3868, Nov. 2007.

[35] G. Wu, Z. Li, H. Wang, and W. Zou, "Selective random cyclic delay diversity for HARQ in cooperative relay," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2010, pp. 1–6.

[36] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: A distributed wireless architecture for exploiting sender diversity," in *Proc. ACM SIGCOMM Conf.*, New York, NY, USA, Sep. 2010, pp. 171–182.

[37] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[38] Nokia Siemens Networks Corporation, "2020: Beyond 4G radio evolution for the gigabit experience," Nokia Siemens Netw. Corp., Espoo, Finland, Tech. Rep., 2011.

[39] T. Levanen, J. Pirskanen, T. Koskela, J. Talvitie, and M. Valkama, "Low latency radio interface for 5G flexible TDD local area communications," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2014, pp. 7–13.

[40] O. N. Yilmaz, Y.-P. E. Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf, and J. Sachs, "Analysis of ultra-reliable and low-latency 5G communication for a factory automation use case," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 1190–1195.

[41] N. Brahmi, O. N. Yilmaz, K. W. Helmersson, S. A. Ashraf, and J. Torsner, "Deployment strategies for ultra-reliable and low-latency communication in factory automation," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.

[42] C. Bockelmann *et al.*, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, Sep. 2016.

[43] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Aug. 2016.

[44] P. Popovski, "Ultra-reliable communication in 5G wireless systems," in *Proc. 1st Int. Conf. 5G Ubiquitous Connectivity (5GU)*, Feb. 2014, pp. 146–151.

[45] W. S. Wong and R. W. Brockett, "Systems with finite communication bandwidth constraints. I. State estimation problems," *IEEE Trans. Autom. Control*, vol. 42, no. 9, pp. 1294–1299, Sep. 1997.

[46] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Trans. Autom. Control*, vol. 49, no. 7, pp. 1056–1068, Jul. 2004.

[47] A. Sahai and S. Mitter, "The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link—Part I: Scalar systems," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3369–3395, Aug. 2006.

[48] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.

[49] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proc. IEEE*, vol. 95, no. 1, pp. 163–187, Jan. 2007.

[50] S. Y. Park and A. Sahai, "Intermittent Kalman filtering: Eigenvalue cycles and nonuniform sampling," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2011, pp. 3692–3697.

[51] G. Ramnarayan, G. Ranade, and A. Sahai, "Side-information in control and estimation," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 171–175.

[52] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith, "Joint optimization of communication rates and linear systems," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 148–153, Jan. 2003.

[53] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith, "Joint optimization of wireless communication and networked control systems," in *Switching and Learning in Feedback Systems*. Berlin, Germany: Springer-Verlag, 2005, pp. 248–272.

[54] V. Gupta, A. F. Dana, J. P. Hespanha, R. M. Murray, and B. Hassibi, "Data transmission over networks for estimation and control," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1807–1819, Aug. 2009.

[55] P. Park, J. Araújo, and K. H. Johansson, "Wireless networked control system co-design," in *Proc. Int. Conf. Netw., Sens. Control (ICNSC)*, Apr. 2011, pp. 486–491.

[56] C. L. Robinson and P. R. Kumar, "Optimizing controller location in networked control systems with packet drops," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 661–671, May 2008.

[57] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam, "The wireless control network: A new approach for control over networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2305–2318, Oct. 2011.

[58] N. Elia, "Remote stabilization over fading channels," *Syst. Control Lett.*, vol. 54, no. 3, pp. 237–249, 2005.

[59] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Commun.*, vol. 12, no. 1, pp. 3–11, Feb. 2005.

[60] Q. Huang, M. Ghogho, J. Wei, and P. Ciblat, "Practical timing and frequency synchronization for OFDM-based cooperative systems," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3706–3716, Jul. 2010.

[61] S. Verdú and S. Shamai (Shitz), "Variable-rate channel capacity," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2651–2667, Jun. 2010.

[62] *Introduction to SERCOS III With Industrial Ethernet*. [Online]. Available: http://www.sercos.com/technology/sercos3.htm

[63] S. V. Hanly and D. N. C. Tse, "Multiaccess fading channels. II. Delay-limited capacities," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2816–2831, Nov. 1998.

[64] L. H. Ozarow, S. Shamai (Shitz), and A. D. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE Trans. Veh. Technol.*, vol. 43, no. 2, pp. 359–378, May 1994.

[65] A. Lozano and D. Porrat, "Non-peaky signals in wideband fading channels: Achievable bit rates and optimal bandwidth," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 246–257, Jan. 2012.

[66] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4232–4265, Jun. 2014.

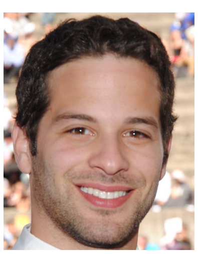
[67] L. Dickstein, V. Narasimha Swamy, G. Ranade, and A. Sahai, "Finite block length coding for low-latency high-reliability wireless communication," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2016, pp. 908–915.

[68] G. D. Forney, Jr., "Exponential error bounds for erasure, list, and decision feedback schemes," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 2, pp. 206–220, Mar. 1968.
[69] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2349–2360, Sep. 2005.

Vasuki Narasimha Swamy received the B.Tech. degree from IIT Madras, Chennai, India, in 2012, and the M.S. degree from the University of California at Berkeley, Berkeley, CA, USA, in 2015, where she is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Sciences. Her research focuses on designing wireless communication protocols for high performance control applications.

Sahaana Suri received the bachelor's degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA. She is currently pursuing the Ph.D. degree with the Stanford InfoLab, with P. Bailis. Her research focuses on building easily accessible data analytics and machine learning systems that scale.

Paul Rigge received the B.S. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 2012. He held internship positions with the NASA Jet Propulsion Laboratory, Pasadena, CA, USA, where he was involved in signal processing. His current research interests include wireless communication system design and agile digital design.



Matthew Weiner received the B.S. degree in engineering from Harvey Mudd College in 2008, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 2010 and 2015, respectively, under the guidance of Prof. B. Nikolić. He is currently affiliated with RF Pixels, Fremont, CA, USA. Previously, he has held positions with Cohere Technologies (2015), Intel Corporation (2011), and Marvell Semiconductor (2010). His research interests include wireless protocols, wireless communication system design, and the efficient implementations of beamforming and error control coding algorithms.

Gireeja Ranade received the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, and the SB degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA. She is currently a Researcher with Microsoft Research, Redmond, WA, USA. Her research interests include information theory, control theory, wireless communications, and other areas.

Anant Sahai received the B.S. degree from the University of California at Berkeley, Berkeley, CA, USA, in 1994, and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1996 and 2001, respectively. Prior to joining the Berkeley faculty in 2002, he spent a year as a Research Scientist with the wireless startup Enuvis, South San Francisco, CA, USA, developing software-radio signal-processing algorithms to enable very sensitive GPS receivers for indoor operation. From 2007 to 2009, he was the Treasurer for the IEEE Information Theory Society. His current research interests are at the intersection of information theory, wireless communication, and decentralized control - including dynamic spectrum sharing and the role of machine learning in it.

Borivoje Nikolić (S'93–M'99–SM'05–F'17) received the Dipl.Ing. and M.Sc. degrees in electrical engineering from the University of Belgrade, Serbia, in 1992 and 1994, respectively, and the Ph.D. degree from the University of California at Davis in 1999. In 1999, he joined the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, where he is currently a National Semiconductor Distinguished Professor of engineering. He is a co-author of the book *Digital Integrated Circuits: A Design Perspective*, Second Edition, (Prentice-Hall, 2003). His research activities include digital, analog, and RF integrated circuit design and the VLSI implementation of communications and signal processing systems. He received the NSF CAREER Award in 2003, the College of Engineering Best Doctoral Dissertation Prize, and the Anil K. Jain Prize for the Best Doctoral Dissertation in Electrical and Computer Engineering at the University of California at Davis in 1999, as well as the City of Belgrade Award for the Best Diploma Thesis in 1992. For work with his students and colleagues he has received best paper awards at the IEEE International Solid-State Circuits Conference, the Symposium on VLSI Circuits, the IEEE International SOI Conference, the European Solid-State Device Research Conference, the European Solid-State Circuits Conference, the S3S Conference, and the ACM/IEEE International Symposium of Low-Power Electronics. In 2014–2015, he was a Distinguished Lecturer of the IEEE Solid-State Circuits Society.