Wireless Communication for High-reliability Low-latency Control - Part I

Vasuki Narasimha Swamy*, Sahaana Suri*, Paul Rigge*, Gireeja Ranade†, Anant Sahai*, Borivoje Nikolić*

*University of California, Berkeley, CA, USA

†Microsoft Research, Redmond, WA, USA

Abstract

High-performance industrial control systems with tens to hundreds of sensors and actuators have stringent latency and reliability requirements. Current wireless technologies like WiFi, Bluetooth, LTE, etc., are unable to meet these requirements, forcing the use of wired systems. This paper introduces a wireless communication protocol framework, dubbed "Occupy CoW," based on cooperative communication among nodes in the network to build the diversity necessary to deliver the target reliability. Simultaneous retransmission by many relays achieves this without significantly decreasing throughput or increasing latency. The key difficulty to overcome is the common knowledge of who needs to speak what and when.

The protocol is analyzed using the communication theoretic delay-limited-capacity framework and compared to baseline schemes that primarily exploit frequency diversity (including the practically employed WISA). For a scenario inspired by an industrial printing application with 30 nodes in the control loop, total information throughput of 4.8 Mb/s, and cycle time under 2 ms, an idealized protocol can achieve a system probability of error better than 10^{-9} with nominal SNR below 5 dB. We also derive the probability of system failure for all cases.

Index Terms

Cooperative communication, low-latency, high-reliability wireless, industrial control, diversity, Internet of Things

I. Introduction

The Internet of Things (IoT) envisions to enable a large number of globally distributed, embedded, computing devices to communicate with each other and interact with the physical

world. This interaction includes not just sensing but also simultaneous actuation of numerous connected devices. For truly immersive applications, the latency requirements on the control loop are in the tens of milliseconds. This pushes the demand on the communication link latency to the order of a millisecond, while demanding very high-reliability. These requirements parallel those of modern industrial automation [1], with a round-trip delay of approximately 1 ms [2] and reliability of 10^{-8} [3], as achieved with wired connections.

This paper is the first in a trilogy about cooperative communication for low-latency high-reliability applications. This paper[1] introduces "Occupy CoW[2]," a communication protocol framework for today's industrial control and future IoT applications, designed to meet these stringent QoS requirements. The second paper integrates network coding into the cooperative communication protocol dubbed "XOR-CoW" and shows that under ideal conditions, XOR-CoW requires lesser SNR compared to Occupy CoW to meet the same requirements. The third paper analyzes how robust these protocols are to channel-modeling assumptions. the impact of channel models on the performance of both Occupy CoW and XOR-CoW. We challenge knowledge of fading distributions, independent fading across channels, channel reciprocity and quasi-static nature.

Our main goal is to facilitate a plug-and-play transition from wired to wireless. This work builds crucially on [1], which established the need to attack this problem from the PHY/MAC layers and proposed a preliminary wireless architecture that focused on low-latency operation through the use of reliable broadcasting, semi-fixed resource allocation, and low-rate coding. The key point of this paper is that multi-user diversity can achieve the desired reliability without relying on time or frequency diversity created by natural multipath or frequency selectivity.

To motivate our protocol from the industrial control context, we first review the evolution of communication for industrial control and then briefly review cooperative communication and wireless diversity techniques in Section II. After that review, Section III describes our multi-user-diversity-based protocol in detail. Section IV presents how it performs, how its internal parameters are optimized, and compares it to hypothetical frequency-diversity-based schemes. All the formulas used to generate the plots are derived in the Appendix.

[1]A conference version of this paper [4] was published at IEEE ICC 2015. This paper expands on the results of [4]

[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 during the "Occupy Wall Street" movement [5].

II. Related Work

A. Industrial control

Communication in industrial control systems has 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 [6]–[8]. The main objective of fieldbus systems is to provide reliable real-time communication. There is a further desire to move to wireless communications for industrial control environments to reduce bulk and installation costs [9], and several wireless extensions of fieldbus systems have been examined [10], [11]. Unfortunately, these do not work in high-reliability settings since present designs for 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 transmission delays [12]. On the other hand, ideas from wireless communication in Wireless Sensor Networks (WSNs) [13]–[15] that provide high-reliability monitoring also cannot be easily adapted for tight control loops because they inherently tolerate large latencies [16].

The current generation of leading wireless technologies for industrial control are all based on successful WSN ideas. The Wireless Interface for Sensors and Actuators (WISA) [17] attempts to meet stringent real-time requirements, but fails to achieve interoperability and multi-path routing. The reliability of WISA ($\approx 10^{-4}$) does not work as a drop-in replacement for control [18]. ZigBee PRO [19] also fails to deliver high enough reliability [20]. Both ISA 100 [21] and WirelessHART [22] provide secure and reliable communication, but have relaxed latency bounds since they focus on non-time-critical applications. These schemes are unable to hit the 2ms requirement we consider here. [20], [23]

There is a need for a faster and more reliable protocol if we want to have a drop-in replacement for existing wired fieldbuses like SERCOS III, which provide a reliability of 10^{-8} and latency of 1 ms when communicating among tens of nodes. We now review some wireless communication techniques which can aid in designing a protocol which can meet the stringent requirements.

B. Cooperative communication and multi-user diversity

Wireless sensor networks are highly reliable and use many techniques like channel hopping, contention-based MACs and multi-path routing to harvest time and frequency diversity [9]. However, most strategies for WSNs or industrial control networks do not exploit spatial diversity

from multiple antennas or user cooperation, except implicitly through higher-layer approaches. Low-latency applications like ours cannot use time diversity since the cycle time is shorter than the coherence time. Techniques like Forward Error Correction and Automatic Repeat Request (ARQ) also do not provide much advantage [24]. 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. Consequently, our protocol leverages spatial diversity instead.

The size of the networks targeted in this paper is moderate (say 10 - 100 nodes). Therefore there is an abundance of antennas in the system and we can take advantage of it by harvesting cooperative and multi-user diversity. Multi-antenna diversity are mainly of two types: a) sender diversity where multiple antennas transmit the same message through independent channels and b) receiver diversity where the receiver has multiple antennas to harvest multiple copies of the same message received via independent channels. Many researchers have studied these techniques in great detail; so our treatment here is limited. Laneman et al. [25] showed that cooperation amongst distributed antennas can provide full sender-diversity without the need for physical arrays. Even with a noisy inter-user channel, multi-user cooperation increases capacity and leads to achievable rates that are robust to channel variations [26]. The prior works in cooperative communication tends to focus on the asymptotic regimes of high SNR. By contrast, we are interested in moderate SNR regimes.

Multi-antenna techniques have been widely implemented in commercial wireless protocols like IEEE 802.11. [24], [27] use relays and a TDMA-based scheme to bring sender-diversity techniques to industrial control. Unfortunately, TDMA can scale badly with network size. To scale better with network size, our protocol uses simultaneous transmission by many relays, using some distributed space-time codes such as those in [28]–[30], so that each receiver can harvest a large diversity gain. This allows the protocol to achieve ultra-high-reliability without greatly decreasing throughput or increasing latency. While we do not discuss the specifics of space-time code implementation, recent work by Katabi *et al*. demonstrates that it is possible to implement schemes that harvest sender diversity using concurrent transmissions [31].

III. Protocol Design

The Occupy CoW protocol exploits multi-user diversity 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.

The network can be visualized as in the bottom right diagram in Fig. 1. All messages must flow in a star topology from the central controller to the individual nodes, and in the reverse direction from the nodes to the controller. As seen in in Fig. 1, there exists a central controller (C) that must transmit m distinct bits of information to each of the n nodes. This is the downlink stage of the protocol. Each of the n nodes in $\mathcal{S}$ must then transmit its unique m bits of information to the controller. This is the uplink stage of the protocol. 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, or both.

We assume that while normally, the controller and all nodes are in-range of each other, bad fading events can cause transmissions to fail. The protocol uses different nodes as relays to overcome this. On the downlink side, nodes that have received messages from the controller act as *simultaneous* relays to deliver messages to their destinations in a multi-hop fashion. A similar idea is applied for the uplink. When they are not transmitting, all nodes are listening. Nodes that have successfully decoded messages act as simultaneous relays for that message. This protocol is implemented by dividing every communication cycle into three phases each for downlink and uplink, with a small (but critical) scheduling and acknowledgment phase mixed in.

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, and messages contain their destination address.

Fundamentally, errors are caused by deep fades. Since the short cycle time puts us in the non-ergodic flat-fading regime, time diversity cannot be used. All nodes are assumed to be capable of instantly decoding variable-rate transmissions [32]. 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 [33]. Thus we can schedule time slots for specific nodes without any overhead. The protocol relies on time/frequency synchronization to achieve simultaneous retransmission of messages by multiple relays. We assume that if k relays

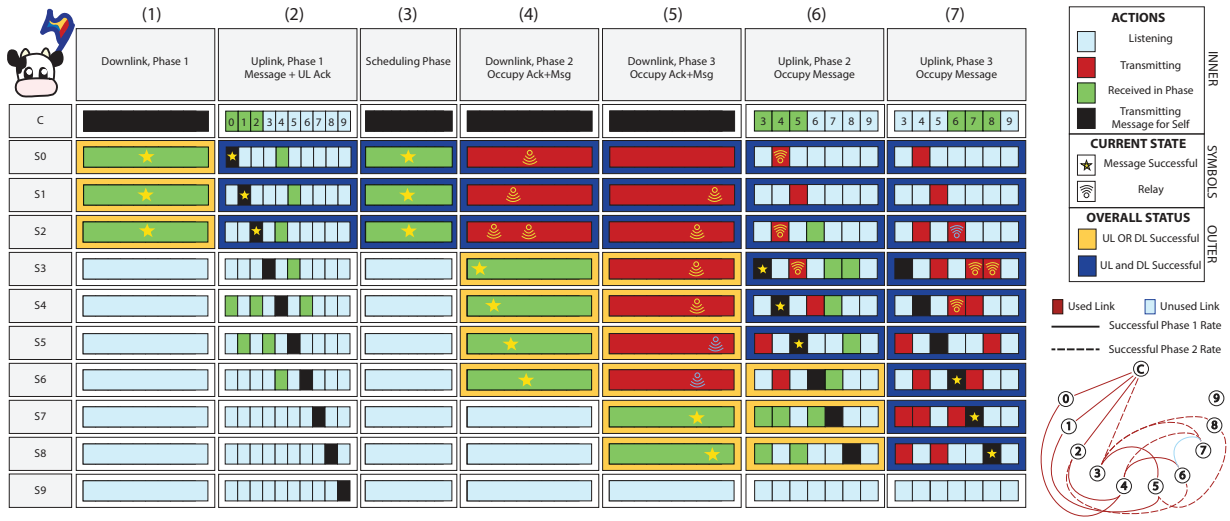


Fig. 1: The seven phases of the Occupy CoW protocol illustrated by a representative example. The table shows a variety of successful downlink and uplink transmissions using 0, 1 or 2 relays. S9 is unsuccessful for both downlink and uplink. The graph on the right shows the underlying link-strengths for the network.

simultaneously (with consciously introduced jitter[3]) transmit, then all receivers can extract signal diversity k.

A. Downlink and Uplink Phase I

Downlink Phase I (length T_{D_1}) is used by the controller to broadcast all m-bit messages to all n slave nodes at rate $R_{D_1} = \frac{b \cdot n}{T_{D_1}}$. At this point, each of the nodes are listening for information. In the instance depicted in Fig. 1, 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 the bottom right diagram in the figure. At this point, $S0$, $S1$, and $S2$ have decoded both their individual messages as well as message intended for all of the other nodes.

This is followed by Uplink Phase I (length T_{U_1}), 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 $R_{U_1} = \frac{b+1}{T_{U_1}/n} = \frac{(b+1) \cdot n}{T_{U_1}}$ by evenly dividing the time slots among all slave nodes. In Fig. 1, Column 2, only $S0$, $S1$, and $S2$ successfully transmit

[3]To transform spatial diversity into frequency-diversity [30].

their messages to the controller. As before, when a node is not transmitting, it is trying to listen for the messages being sent. Here, we see that $S4$ and $S0$ are able to hear each other, as are $S1$ and $S5$, and so on. We can also see the links between these nodes in the bottom right diagram. All successes that have occurred thus far have succeeded due to *direct* connections between nodes $S0$, $S1$, and $S2$, and the controller. Due to this, we refer to these types of successes as "one-hop" successes — the messages only traveled a single hop. Should we terminate the protocol at this point, it would be dubbed a one-hop, or single phase, protocol, as all successes must occur via a single hop.

B. Scheduling information

The scheduling phase (length T_S) is used by the controller to transmit acknowledgments to the strong users (Fig. 1, Column 3). This is just 2 bits of information per slave node for downlink and uplink. The common-information about the system's state transmitted during this phase enables the controller and other nodes to share a common schedule for relaying messages for the remaining nodes. The strong nodes that are able to help must receive this information, and it doesn't matter that other nodes do not have this information at this time since they have nothing useful to say. This common-information is passed on to the remaining nodes in the downlink phases to follow.

If only nodes which have failed in phase I are to be helped in the upcoming phases (i.e., the rates are adapted eliminating the already successful nodes) and we choose to have a flexible schedule, then this phase is absolutely crucial. The common ack information also allows the scheme to use possibly lower rates R_{D_2}, R_{D_3}, R_{U_2} and R_{U_3}, as we will see. If slots are reserved to help *every node*, then this phase is optional as it doesn't help in determining schedule. It might perhaps still be useful in reducing unnecessary transmission for already successful nodes. The strong nodes S0-S2 in Fig. 1 receive the ack information.

C. Downlink Phase II and III

In Downlink Phase II (length T_{D_2}) the controller can choose to alter its broadcast message to remove already-successful messages for the strong nodes; so the packet is sent at an adapted rate, $R_{D_2} = \frac{2n+bn_1}{T_{D_2}}$, where n_1 is the number of nodes that were not successful in Phase I. Alternatively, we can choose to have a fixed schedule where everyone gets another shot at succeeding and

transmit at rate, $R_{D_2} = \frac{(2+b)n}{T_{D_2}}$. At this point, the controller and all nodes that were successful in the first phase broadcast the message they heard *along with the scheduling message*.

It is possible that nodes that were initially unable to directly connect to the controller may now be able to, *if* the rate during this phase is lower than that of the first. This is a very important point to note, and may occur if enough nodes are successful in the first phase since fewer messages must now be sent or if the time allocated for this phase T_{D_2} is greater than T_{D_1} resulting in a lower rate. If we choose to have an adaptive schedule, then in our example the messages for $S0$, $S1$, and $S2$ need not be transmitted again. In Fig. 1, Column 4, node $S3$ gets its message directly through the controller (due to reduced rate), hence the connection between node $S3$ and the controller is dashed in the bottom right diagram. Additionally, in this phase, we effectively use relaying for the first time — introducing the possibility for "two-hop" successes. We refer to them as two-hop successes as the messages must be transmitted via two different nodes before reaching their final destination. In Fig. 1, $S2$ (initially successful) is able to reach $S4$. This means $S4$ successfully receives the controller's message and the scheduling message in two hops via $S2$. In a similar way, $S5$ hears the controller's message via $S1$, and $S6$ could have heard the message from either $S0$ or $S2$. At this point, nodes $S0$, $S1$, $S2$, $S3$, $S4$, $S5$, and $S6$ have all successfully received their messages from the controller.

Downlink Phase III (length T_{D_3}) follows the same structure as downlink Phase II, and transmits using the rate $R_{D_3} = \frac{2n+bn_1}{T_{D_3}}$ (if we choose flexible scheduling) or $R_{D_3} = \frac{(2+b)n}{T_{D_3}}$ (if we choose fixed scheduling). There exists the potential of three-hop relay paths from those who were successful in Phase II. For example, in Fig. 1, Column 5, S7 succeeds through S3. 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.

Note that the strong nodes that received the information from the controller in Phase I are the bottleneck for successful relay paths to other nodes during downlink.

D. Uplink Phase II and III

The calculated schedule from earlier phases allocates a slot in Phases II (length T_{U_2}) and III (length T_{U_3}) for each unsuccessful node from Uplink Phase I (if we choose adaptive scheduling) or for each node (if we choose fixed scheduling). Time slots can either be evenly divided among all n_1 unsuccessful slaves or among all slave nodes. For the rest of the section we will assume

that time slots have been evenly divided among the unsuccessful notes and the treatment for the other case is similar. In the slot for each failed slave node, the slave and everyone who heard that slave in an earlier uplink phase will simultaneously transmit the relevant message at the new rates $R_{U_2} = \frac{n_1 \cdot b}{T_{U_2}}$ and $R_{U_3} = \frac{n_1 \cdot b}{T_{U_3}}$.

This creates the potential for two-hop relaying if another slave heard the message in Uplink Phase I. For example, S2 and S0 transmit the message for S4 to the controller in Fig. 1, Column 6, since they already heard S4 in Phase I. Three-hop relaying is also possible in Uplink Phase III, for example the S6 → S4 → S2 → C chain in Figure 1, Column 7. Note that this relies on S4 hearing S6 in Phase I, and S2 hearing S4 in Phase II. It is also possible to have new two-hop relay paths emerge due to the creation of new links (e.g. S7 to S3 in Phase II and S3 to controller in Phase III).

The uplink phases are similar to their downlink counterparts, but are in a sense inverted. The bottleneck to the controller now occurs on the last-hop, i.e. in Phase III.

As a final note, the exact transmission rates for each of the uplink and downlink phases depend on the time allocated and number of nodes remaining. We will provide exact formulas for each scenario that might possibly arise in the appendix.

IV. Analysis of Occupy CoW

We explore Occupy CoW with parameters in the neighborhood of a practical application, the industrial printer case described in [1]. Recall that in one practical scenario, the SERCOS III protocol [34] supports the printer's required cycle time of 2 ms with reliability of 10^{-8}. So we target a 10^{-9} probability of error 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. If we assume 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.

A. Behavioral assumptions for analysis

We include the following behavioral assumptions in addition to the resource assumptions in Sec. III. We assume a fixed nominal SNR on all links with independent Rayleigh fading on each

link. We assume a single tap channel[4] (hence flat-fading). Because the cycle-time is so short, we use the delay-limited-capacity framework [35], [36]. We also assume channel reciprocity.

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

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

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 dispersion-style finite-block-length effects on decoding (justified in spirit by [38]). A related assumption is that no transmission or decoding errors are undetected [39] — a corrupted packet can be identified[6] and is then completely discarded.

Equations for error probabilities corresponding for different hops (both uplink and downlink) are derived in the appendix.

B. Results and comparison

Following [1] and the communication-theoretic convention, we use the minimum SNR required to achieve 10^{-9} reliability as our metric to compare Occupy CoW to two other baseline schemes.

Fig. 2 looks at performance with fixed payload size $m = 160$ bits as the number of nodes n, varies. Initially the minimum required SNR for Occupy CoW decreases with increasing n, even through the throughput increases as $b \cdot n$, but the curves then flatten out[7].

The topmost comparison scheme (blue solid curve) restricts uplink and downlink to the first hop of Occupy CoW. The required SNR shoots off the figure, because the throughput increases linearly with nodes and it still gets only one shot at succeeding. The second scheme (red dashed

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

[5]We are ignoring a subtle effect here due to space limitations. The cyclic-delay-diversity space-time-coding schemes we envision effectively make the channel response longer. This pushes the PHY into the "wideband regime" in wireless communication theory, and a full analysis must account for the required increase in channel sounding by pilots to learn this channel [37]. 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.

[6]Consider all messages to include a 40 bit hash that is checked. This can be added to the underlying message size.

[7]This impact of multi-user diversity eventually gives way and the required SNR would start to increase for very large n.

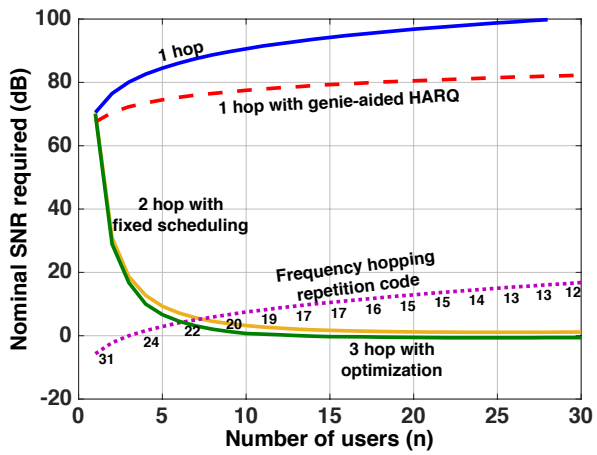


Fig. 2: The performance of Occupy CoW as compared with reference schemes for $m = 160$ bit messages and $n = 30$ nodes with 20MHz and a 2ms cycle time, aiming at 10^{-9}. The numbers next to the frequency-hopping scheme represent the amount of frequency diversity needed.

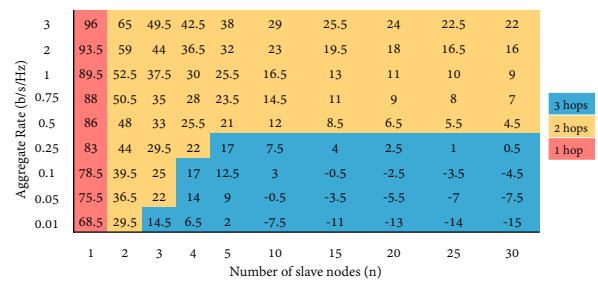


Fig. 3: The above figure tells us the number of hops and minimum SNR to be operating at to achieve a high-performance of 10^{-9} as aggregate rate and number of users are varied. Here, the time division within a cycle is unoptimized. Uplink and downlink have equal time, 2-hops has a 1:1 ratio across phases, and 3-hops has a 1:1:1 ratio for the 3 phases.

curve) is purely hypothetical. It allows each node to use the entire 2 ms 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.

The last reference curve (purple dotted line) represents a hypothetical (non-adaptive) frequency-hopping scheme that divides the bandwidth W into k sub-channels that are assumed to be independently faded. The curve is annotated with the optimal k. As k (and thus frequency 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 (requiring a spectral efficiency less than 1.5bits/s/Hz) that we are still in the emergy-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 independently faded sub-channels to repeat across.

Amongst Occupy CoW schemes, we compare fixed schedule 2-hop protocol with equal times

for each phase and 3-hop scheme optimized to minimize SNR. We see that the choice between 2-hop and 3-hop or doing a fixed or adaptible schedule is not very important and we will discuss this in detail in section V.

It turns out that the aggregate throughput 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. 3. 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, the one-hop scheme will be best even with more users. But when the target reliability is 10^{-9}, this is at absurdly high aggregate rates[8]. In the practical regime, diversity wins.

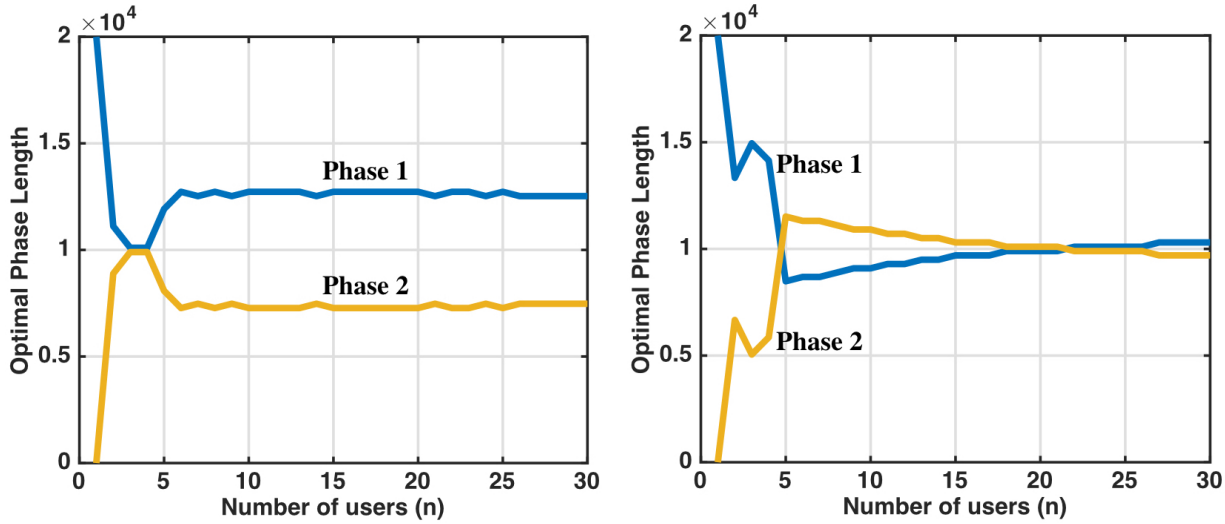
V. Phase-length optimization

We have described uplink and downlink protocols with multiple phases including fixed scheduling and adaptive scheduling — thus providing two protocol selection parameters. A third parameter is the time allocated for different phases. It may seem natural to allocate the same amount of time for each phase so that links in different phases fail with the same probability but we find that smarter allocation of time (resulting in unequal phase lengths) lower the SNR required to achieve the same specs. We consider downlink and uplink protocols separately and look at the optimal allocation of time for both 2-hop and 3-hop protocol which minimizes the SNR required to meet the performance specifications. The saving in SNR that we achieve by allocating optimum phase lengths for different phases is minimal. The 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 optimization.

A. *Phase length allocation in 2-hop protocol*

In the 2-hop protocol, the time available for downlink is 1ms and uplink is 1ms. We only look at the flexible scheduling protocol which allocates time equally only for the unsuccessful

[8]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.



(a) Optimal fraction of time allocated for downlink phase I and II in the 2-hop protocol at the smallest SNR which meets the performance requirements.

(b) Optimal fraction of time allocated for uplink phase I and II in the 2-hop protocol at the smallest SNR which meets the performance requirements.

Fig. 4: Optimal phase allocation for 2-hop protocol. Parameters used were 160 bit messages, 30 users, 2×10^4 total bits.

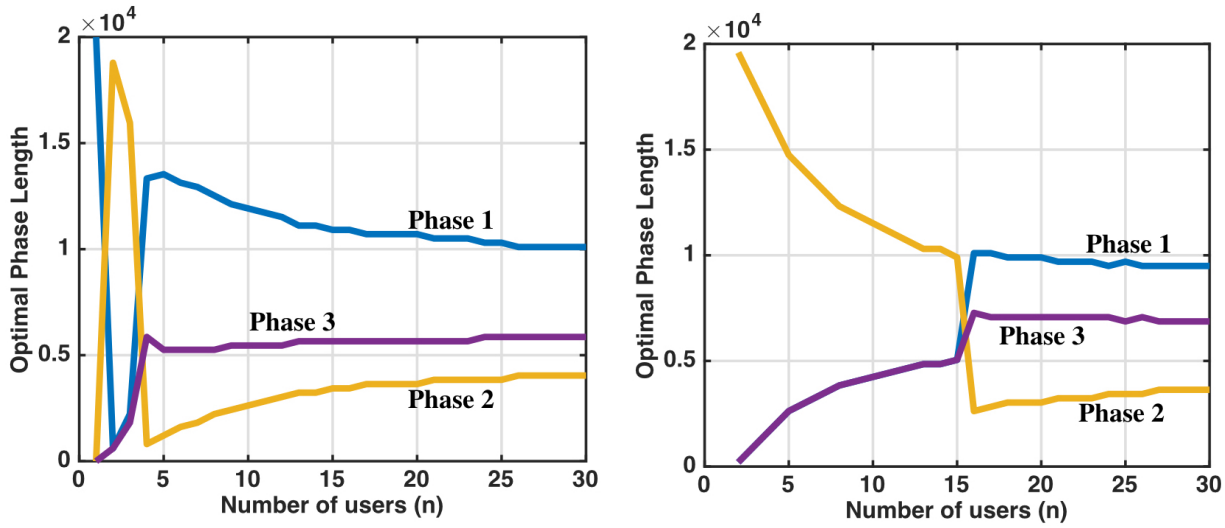
nodes. Let the time allocated for phase I of downlink and uplink be T_{D_1} and T_{U_1} respectively and the time allocated for phase II of downlink and uplink be T_{D_2} and T_{U_2} respectively such that $T_{D_1} + T_{D_2} = 1$ms and $T_{U_1} + T_{U_2}x = 1$ms. We search over all allocations of T_{D_1}, T_{D_2}, T_{U_1} and T_{U_2} such that the above conditions are met.

Downlink: Figure 4a shows the optimal allocation of time for phase I and II for downlink. For mid-large size networks (5 - 30), phase I is allocated a longer time than phase II. In the flexible scheduling protocol, we can anticipate that some nodes succeed in the first phase and we can remove their downlink information from phase II packet. As the phase II packet size is reduced, we can maintain a coding rate comparable with phase I with a smaller time.

Uplink: Figure 4b shows the optimal allocation of time for phase I and II for uplink. The optimum allocation is different for uplink and downlink. The key insight is in the difference between the paths taken to succeed in downlink and uplink. In downlink, nodes succeed in the second phase by connecting to successful relays in the second phase — thus depending on the presence of links *different* from the links being utilized in phase I. On the other hand, in uplink the links which were successful in phase I are *reused* in phase II. The coding rate should not go

up as the fades might be unable to supper higher rates. Additionally, there might be nodes which were initially unsuccessful in phase I whose fades can now support the lower rate in phase II. These two paths are the *critical* or *bottleneck* paths for succeeding in uplink phase II and thus allocating more time for phase II is beneficial.

B. Phase length allocation in 3-hop protocol



(a) Optimal fraction of time allocated for downlink phase I, II and III in the 3-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 3-hop protocol at the smallest SNR which meets the performance requirements.

Fig. 5: Optimal phase allocation for 3-hop protocol. Parameters used were 160 bit messages, 30 users, 2×10^4 total bits.

In the 3-hop protocol, the time available for downlink is 1ms and uplink is 1ms. Again, we only look at the flexible scheduling protocol which allocates time equally only for the unsuccessful nodes. Let the time allocated for phase I of downlink and uplink be T_{D_1} and T_{U_1} respectively, the time allocated for phase II of downlink and uplink be T_{D_2} and T_{U_2} respectively and the time allocated for phase III of downlink and uplink be T_{D_3} 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. We search over all allocations of T_{D_1}, T_{D_2}, T_{D_3}, T_{U_1}, T_{U_2} and T_{U_3} such that the above conditions are met.

Downlink: Figure 5a 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 size 1 and 2 where the optimal strategy is 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 open up during phase III. Phase I is longest to ensure that the messages are successfully decoded by enough number of nodes in the beginning to ensure maximal spread.

To further understand why it is better to allocate more time to phase I in downlink, consider the difference between a link that fails in phase I and a link that fails in a later phase. A link between node i and the controller that fails in phase I is equivalent to all of the other $n-1$ links at node i failing in phase II. A link connected to node i that fails in phase II does not prevent other nodes from using node i as a relay from the controller. Then we see that a link between node i and the controller is on many more paths from the controller than a link connected to node i in phase II. As a result, we view the qualities of the links between the controller and each node as the bottleneck of the system. Allocating more time to phase I during downlink improves these critical links at the expense of less important links in later phases. This explains why downlink protocols perform better with a longer phase III.

Uplink: Figure 4b shows the optimal allocation of time for phase I, II and III for uplink. Though the order of time allocated is similar to downlink, the absolute numbers are different and we see that phase III is allocated almost as much as phase I. The reasoning is similar to the case of 2-hop uplink where the critical paths are the ones connecting to the controller. Phase III of the 3-hop uplink protocol is effectively as important as phase II of the 2-hop uplink protocol.

C. How much SNR does optimization save?

Without loss of generality, let us consider the downlink protocol. Figure 6 considers three different phase length allocations: the optimal phase length allocation as shown in Fig. 4a, an approximation of the phase allocations for mid size network of $10:3:4$ applied to all network sizes and a simple $2:1:1$ ratio of phase length allocation. For a network size of 30 nodes, we see that while the lowest SNR meeting the performance is -1.3db (solid blue curve with markers), the SNR required at phase allocation $10:3:4$ is -1.08db (dotted purple curve). Moreover, the SNR required for the simple allocation of $2:1:1$ is only -1.06db (solid yellow curve). Though we have many knobs to turn which can optimize the performance of the protocol, we really only get a marginal benefits.

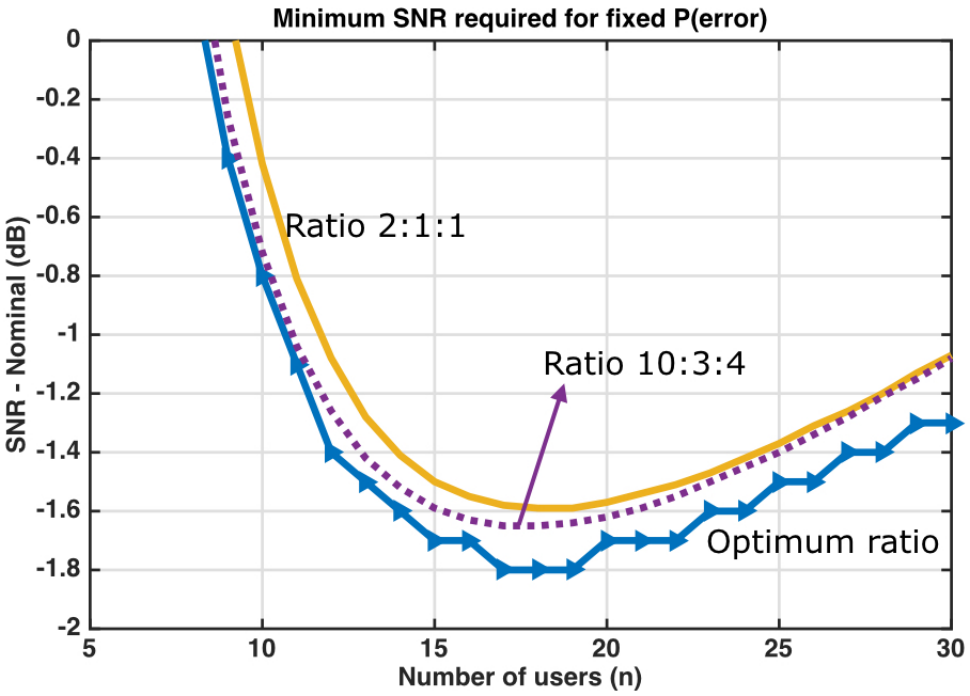


Fig. 6: Comparing the SNR required for optimum downlink phase length allocation, an approximate $10:3:4$ allocation and a simple $2:1:1$ allocation

VI. Conclusions & Future Work

In this work (first paper in the trilogy), we have designed a wireless communication protocol framework for high-performance control-like systems. We have shown why cooperative communication based protocols are the most viable options which meet the stringent system requirements. We have additionally shown that simple allocations of phase lengths are good enough and heavy optimizations only provide marginal benefits. In the second paper we integrate network coding into the cooperative communication protocol dubbed "XOR-CoW" and in the third paper we analyze the impact of channel models on the performance of both Occupy CoW and XOR-CoW.

Acknowledgements

Thanks to Venkat Anantharam and Matthew Weiner 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.

APPENDIX

In order to analyze the reliability of Occupy CoW, we consider the uplink and downlink stages of the protocol separately. We use the union bound to calculate an upper bound on the probability of cycle failure. This is a slightly conservative estimate, since in reality, each phase reuses channels from previous phases and iterations of the protocol.

In our analysis, a downlink failure occurs when at least one node fails to receive its message from the controller in the downlink stage. An uplink failure occurs when the controller fails to receive at least one node's message in the uplink stage. The method of calculating the probability of error for uplink and downlink depends on how many hops the protocol consists of. Finally, a union bound over the uplink and downlink phases is used to determine the overall probability of cycle failure, as noted earlier. We consider the adaptive schedule protocol in all our computations as it is more general. Moreover, the fixed schedule protocol only involves a single tweek in the computation of rates and rest of the computation for any version of the protocol remains the same.

The crux of this analysis relies on partitioning each stage of the protocol into a number of distinct states. As we saw when stepping through Fig. 1, our protocol facilitates successful transmission via various different pathways. Successes and failures occur in many different ways. We account for all means of success by first enumerating all possible paths of success in each phase. We then partition the set of all nodes, $\mathcal{S}$, into sets corresponding to those paths of success (if they succeed), and the set of nodes that fail, $\mathcal{E}$. We refer to any given instantiation of these sets as a state, and the probability of error is calculated by analyzing all possible instantiations of these sets. There are two main methods of analysis used to calculate the probability of error: by counting the number of failure states, or by calculating the probability of failing given a particular state.

We divide the analysis into three sections, corresponding to the one-hop, two-hop, and three-hop protocols. We then derive the probabilities of error for the downlink and uplink stages in each protocol.

Before continuing with the analysis itself, we first define the notation that will be used.

Notation:

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

$$B(n, m, p) = \binom{n}{m}(1-p)^m p^{n-m}. \tag{2}$$

The probability of at least one out of n independent experiments failing will be denoted as

$$F(n, p) = 1 - (1-p)^n. \tag{3}$$

A link with fading coefficient h and bandwidth W is considered "good" (thus decodable) if the rate of transmission R_i is less than or equal to the link's capacity, $C = W \log(1 + |h|^2 \text{SNR})$. We assume that the nominal operating SNR is held consistent across the entire system. Consequently, for a rate R_i, the assumption of Rayleigh fading tells us that the probability of an unsuccessful transmission is defined as

$$p_i = P(R_i > C) = 1 - \exp\left(-\frac{2^{R_i/W} - 1}{\text{SNR}}\right). \tag{4}$$

We assume that if R_i exceeds capacity, the transmission will surely fail (with probability 1). If R_i is less than capacity, the transmission will surely succeed and decode to the right codeword.

Recall that when calculating the probability of cycle error, we partition the set of all nodes into various other sets corresponding to their method of success. Through the course of the analysis, we will be using the sets denoted in Fig. 7 for both uplink and downlink. In addition, all figures used to depict the three protocols (one, two and three-hop) will follow the notation guide in Fig. 7.

Following general convention, for each depicted set, the set itself will be represented in script font. The random variable representing the number of nodes in that set will be presented in uppercase letters. Finally, the instantiation of that random variable (the cardinality of the set), will be in lowercase letters.

A. One-Hop Protocol:

Recall that in this framework the entire protocol consists of stages 1 and 2 of Fig. 1. The controller broadcasts messages, each of length m bits for each node, to the n nodes, and the

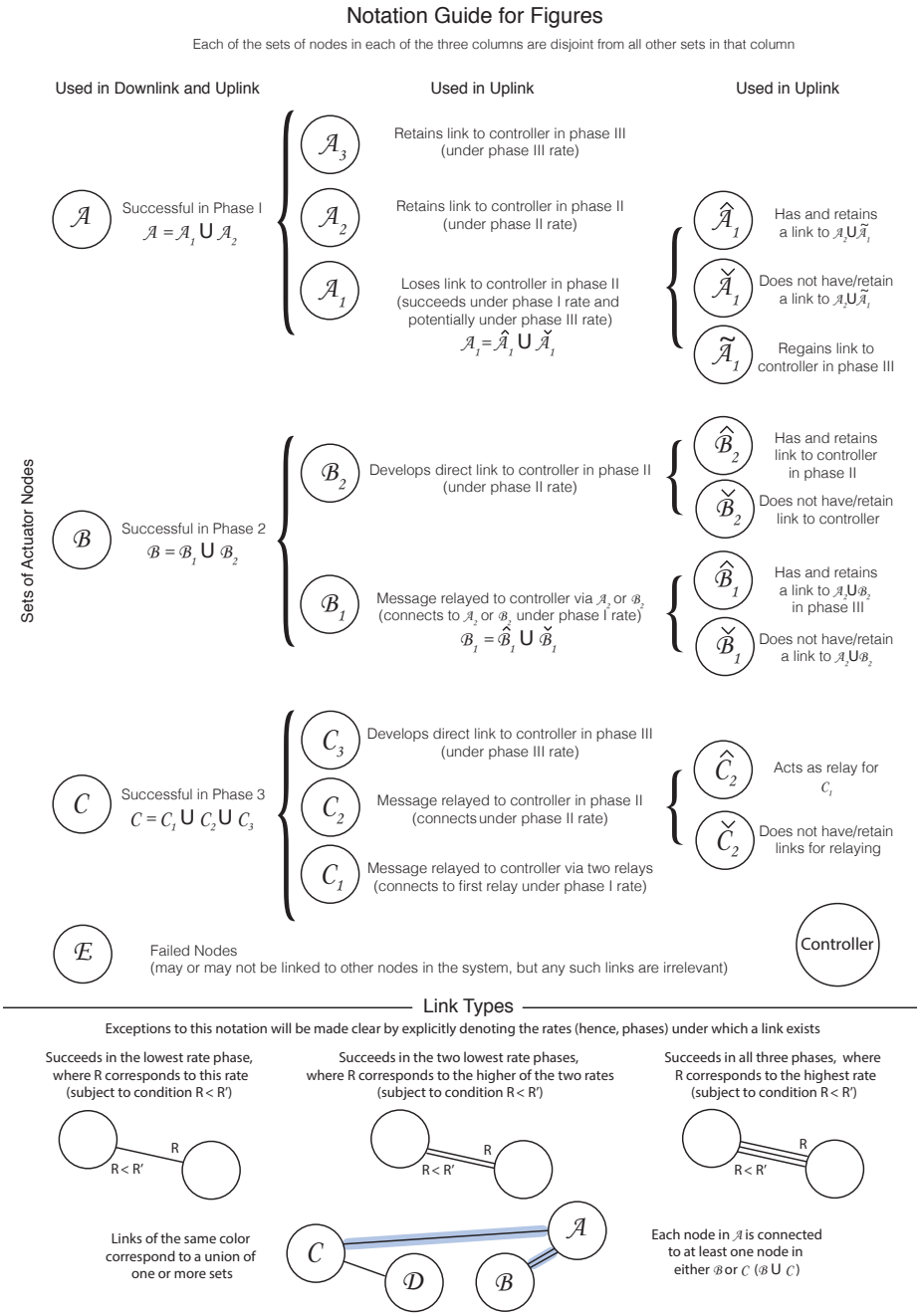


Fig. 7: This figure enumerates the various sets that we will be using throughout the analysis. In addition, how we represent various links in each of the protocol figures is also found here.

nodes respond by transmitting their information as in Fig. 1. In this case, no relaying occurs at all. Downlink receives time T_D and uplink receives time T_U, where $T_U + T_D = T$, the total cycle time.

1) One-Hop Downlink:

Theorem 1: Let the downlink time be T_D, the number of non-controller nodes be n, and the message size be m. The transmission rate is given by $R_D = \frac{m \cdot n}{T_D}$, and the corresponding probability of failure of a single link, denoted by p_D, is given by eq(4). The probability of cycle failure is then

$$P(\text{fail, 1D}) = F(n, p_D) \tag{5}$$

Fig. 8: In this figure, we let $\mathcal{A}$ denote the set of nodes that have a direct link to the controller. A node fails in one hop if it is not in Set $\mathcal{A}$, whether it is completely isolated from the system or not. This case is the same for both downlink and uplink, but the rates of trasmission are R_D and R_U, respectively. Just the downlink is depicted in this figure. Referring back to the original example used in the protocol section, nodes $S0$, $S1$, and $S2$ belong in Set $\mathcal{A}$, while the rest would fall under Set $\mathcal{E}$.

Proof: The rate of transmission is $R_D = \frac{m \cdot n}{T_D}$. Hence, following Eq. (4), we can define probability p_D of failure of a single link. The protocol succeeds only if all nodes receive their messages from the controller in a single transmission. Therefore their point-to-point links to the controller must all succeed (see Fig. 8). Thus we get that the probability of failure for a one-hop downlink protocol is $P(\text{fail, 1D}) = F(n, p_D)$. ■

2) One-Hop Uplink:

Theorem 2: Let the uplink time be T_U, the number of non-controller nodes be n, and the message size be m. The transmission rate is given by $R_U = \frac{m \cdot n}{T_U}$ and the corresponding probability of failure of a single link, denoted by p_U, is given by eq(4). The probability of cycle failure is then

$$P(\text{fail, 1U}) = F(n, p_U). \tag{6}$$

Proof: For the uplink transmission rate of $R_U = \frac{m \cdot n}{T_U}$, the probability of failure of a single link is denoted as p_U. Analogous to downlink, a one-hop uplink protocol succeeds if and only if all nodes get their information to the controller in a single transmission (see Fig. 8). Thus we get $P(\text{fail, 1U}) = F(n, p_U)$. ■

B. Two-Hop Protocol

In a two-hop protocol, both the controller and the nodes get two chances to get their messages across. Phases 5 and 7 in Fig. 1 would not occur. Again we use the union bound to upper bound the total probability of cycle error by adding the probability of downlink failure and the probability uplink failure. If downlink wasn't successful, the nodes would not have the scheduling information thus leading to uplink failure as well. Thus, we see that the union bound is a slightly conservative estimate of the total probability of cycle failure.

1) Two-Hop Downlink:

Theorem 3: Let the Phase I downlink time be T_{D_1}, the Phase II downlink time be T_{D_2}, the number of non-controller nodes be n, and the message size be m. The Phase I transmission rate is given by $R_{D_1} = \frac{m \cdot n}{T_{D_1}}$ and the corresponding probability of a single link failure, p_{D_1}, is given by eq(4). The Phase II transmission rate is given by $R_{D_2}^{(a)} = \frac{m \cdot (n-a)}{T_{D_2}} + \frac{2n}{T_{D_2}}$, where a is the number of "successful nodes" in Phase I and the corresponding probability of a single failure, $p_{D_2}^{(a)}$, is given by eq(4) (the superscript is to indicate the dependence on a). The probability of downlink failure is then

$$P(\text{fail, 2D}) = \sum_{a=0}^{n-1} F\left(n-a, \left(p_{D_2}^{(a)}\right)^a \cdot p_{con}^{(a)}\right) B(n, a, p_{D_1}) \tag{7}$$

where, $p_{con}^{(a)} = \min\left(\frac{p_{D_2}^{(a)}}{p_{D_1}}, 1\right)$.

Proof: A node can succeed by having a direct link to the controller in the first hop ($\mathcal{A}$), or by having a direct link to either the controller or the initially successful nodes in the second hop ($\mathcal{B}$). Note that it is possible for a node to not have a direct link to the controller under the initial rate, but have a direct link under the Phase II rate. In Fig. 9, we see that this list is exhaustive. We will now derive the probability that there exists at least one node that does not fall in Set $\mathcal{A}$ or $\mathcal{B}$.

The rate of transmission in Phase I, R_{D_1}, is dictated by the time allocated for this phase, T_{D_1}, given by $\frac{m \cdot n}{T_{D_1}}$. Let $\mathcal{A}$ (cardinality a), be the set of successful nodes in Phase I. The rate in Phase II, $R_{D_2}^{(a)}$, depends on the realized a and the time allocated for this phase, T_{D_2}. The result is

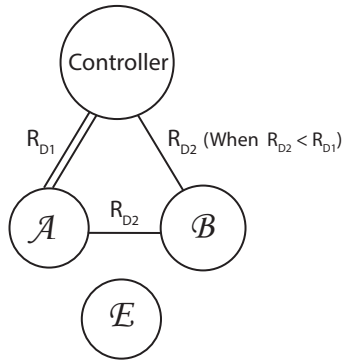


Fig. 9: The only ways to succeed in a two-hop protocol is by having a direct link to the controller to begin with (double line), or having a direct link under the new rate (single line) to either the controller or one of the nodes who heard the controller to begin with.

$R_{D_2}^{(a)} = \frac{m\cdot(n-a)}{T_{D_2}} + \frac{2n}{T_{D_2}}$, where $\frac{2n}{T_{D_2}}$ is the rate of the scheduling message sent (1 bit for downlink acknowledgement and 1 bit for uplink acknowledgement). For ease of analysis, we make use of the fact that the scheduling phase effectively behaves as an extension of the downlink portion of the protocol. Let the probability of link failure corresponding to R_{D_1} and $R_{D_2}^{(a)}$ be defined as p_{D_1} and $p_{D_2}^{(a)}$, respectively, by following Eq. (4)). As mentioned before, a link to the controller may improve in Phase II. The probability that a controller-to-node link fails in phase II, given it failed in phase I, is given by[9] $p_{con}^{(a)} = P\left(R_{D_2}^{(a)} > C | R_{D_1} > C\right) = \min\left(\frac{p_{D_2}^{(a)}}{p_{D_1}}, 1\right)$.

We decouple the two phases of the protocol. An error event can only occur if fewer than n nodes succeed in Phase I — $\mathcal{A} < n$. The probability of a certain number of nodes succeeding in the first round, $P(A = a)$ can be modeled as a binomial distribution with probability of failure p_{D_1}, as a node must rely on just its link to the controller. Thus, $P(A = a) = B(n, a, p_{D_1})$.

Conditioned on the number of nodes that succeeded in Phase I, the probability of a node in $\mathcal{S}\backslash\mathcal{A}$ failing in Phase II reduces to the probability of the node failing to reach any of the nodes in $\mathcal{A}$ and the controller under the new rate, $R_{D_2}^{(a)}$. Each node in $\mathcal{S}\backslash\mathcal{A}$ has a probability $\left(p_{D_2}^{(a)}\right)^a \cdot p_{con}^{(a)}$ of failing in this way, where $p_{con}^{(a)}$ is the probability of failing to the controller under the new rate and $\left(p_{D_2}^{(a)}\right)^a$ is the probability of failing to reach any of the previously successful nodes. Hence

[9]Recall that the fading distributions are assumed to be Rayleigh. Hence $p_{con}^{(a)} = P(R_{D_2}^{(a)} > C | R_{D_1} > C) = \frac{P(R_{D_2}^{(a)} > C \& R_{D_1} > C)}{P(R_{D_1} > C)} = \frac{P(C < \min\{R_{D_1}, R_{D_2}^{(a)}\})}{P(C < R_{D_1})}$. Then we use Eq. (4) to get the final expression.

the probability that at least one of the remaining $n-a$ is unable to connect to the controller can be expressed with Eq. (3) as, $P(\text{fail}|A=a) = F\left(n-a, \left(p_{D_2}^{(a)}\right)^a \cdot p_{con}^{(a)}\right)$.

We then sum over all possible values of a less than or equal to $n-1$, as a cycle failure only occurs when at least one node fails. The probability of failure of the 2-hop downlink protocol is then given by:

$$\begin{aligned} P(\text{fail, 2D}) &= \sum_{a=0}^{n-1} P(\text{fail}|A=a) \cdot P(A=a) \\ &= \sum_{a=0}^{n-1} F\left(n-a, \left(p_{D_2}^{(a)}\right)^a \cdot p_{con}^{(a)}\right) B(n, a, p_{D_1}) \end{aligned} \tag{8}$$

■

2) Two-Hop Uplink:

Theorem 4: Let the Phase I uplink time be T_{U_1}, the Phase II uplink time be T_{U_2}, the number of non-controller nodes be n and the message size be m. The Phase I transmission rate is given by $R_{U_1} = \frac{m \cdot n}{T_{U_1}}$, and the corresponding probability of a single link failure, p_{U_1}, is given by eq(4). The Phase II transmission rate is given by $R_{U_2}^{(a)} = \frac{m \cdot (n-a)}{T_{U_2}}$, where a is the number of "successful nodes" in Phase I and the corresponding probability of a single failure, $p_{U_2}^{(a)}$, is given by eq(4). The probability of cycle failure is then

$$\begin{aligned} P(\text{fail, 2U}) = &\sum_{a=0}^{a_0-1} \sum_{a_2=0}^{a} F\left(M_U, p_{U_1}^{a_2}\right) B\left(a, a_2, q^{(a)}\right) \cdot B(n, a, p_{U_1}) \\ &+ \sum_{a=a_0}^{n-1} \sum_{b_2=0}^{M_U-1} F\left(M_U - b_2, p_{U_1}^{a+b_2}\right) B\left(M_U, b_2, 1-\tilde{q}^{(a)}\right) B(n, a, p_{U_1}) \end{aligned} \tag{9}$$

where,

- $a_0 = \min\left(n \cdot \frac{T_{U_1} - T_{U_2}}{T_{U_1}}, 0\right)$
- $q^{(a)} = P\left(C < R_{U_2}^{(a)} | C > R_{U_1}\right) = \frac{p_{U_2}^{(a)} - p_{U_1}}{1 - p_{U_1}}$
- $\tilde{q}^{(a)} = P\left(R_{U_2}^{(a)} < C | R_{U_1} > C\right) = 1 - \frac{p_{U_2}^{(a)}}{p_{U_1}}$
- $M_U = n - a$

Proof: The derivation of the two-hop uplink error is a little more involved. For the two-hop uplink, the rate of transmission in Phase I, R_{U_1}, is dictated by the time allocated for this phase, T_{U_1} and is equal to $\frac{m \cdot n}{T_{U_1}}$. Let the nodes that were successful in Phase I be in Set $\mathcal{A}$ (cardinality a). The rate in Phase II, $R_{U_2}^{(a)}$, depends on the realization of a, and the time allocated for this phase, T_{U_2}. The result is $R_{U_2}^{(a)} = \frac{m \cdot (n-a)}{T_{U_2}}$. This means there are two distinct cases to consider, one where the new rate has increased, and one where it has decreased.

Case 1: $R_{U_2}^{(a)} \geqslant R_{U_1}$

If the second phase rate is higher, the means of success can be depicted as in Fig. 10. We will now derive the probability of error for this case.

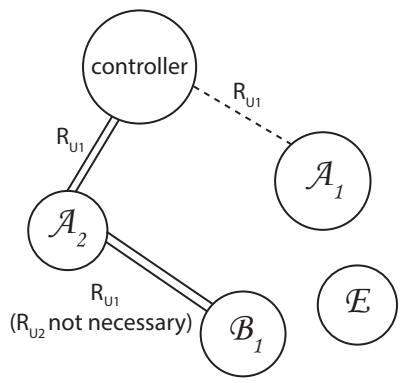


Fig. 10: This figure depicts the possible means of success in a two-hop uplink protocol when the rate increases. The paths are: only having a direct link to the controller under the first rate (dashed line), having a direct link under the new and old rates (double lines) to either the controller or one of the nodes who retained their link to the controller under the new rate. Please refer to Fig. 7 to recall the exact meaning of each set name.

When $R_{U_2}^{(a)} \geqslant R_{U_1}$, some initially successful links will no longer exist as the link between nodes may not be capable of tolerating a higher rate (the rate of transmission may become larger than capacity). In order to enter this case, there exists a threshold, a_0, of how many users must fail in Phase I. The threshold is derived from the condition for having $R_{U_2}^{(a)} \geqslant R_{U_1}$, as $a_0 = \min\left(n \cdot \frac{T_{U_1} - T_{U_2}}{T_{U_1}}, 0\right)$.

There exist three methods of success in a two-hop uplink protocol with potentially increased rate.

- A node can have a direct link to the controller in the first phase, and in the second phase as well, under the higher rate. Let $\mathcal{A}_2$ (cardinality = a_2) be the nodes in $\mathcal{A}$ that retain their connection to the controller in both phases.
- A node can simply have a link to the controller in the first phase, and lose its connection in the second phase. Let the probability of a successful link (in Phase I) failing in Phase

II be denoted as[10] $q^{(a)} = P(C < R_{U_2}^{(a)} | C > R_{U_1}) = \frac{p_{U_2}^{(a)} - p_{U_1}}{1 - p_{U_1}}$. The nodes that lose their links are in Set $\mathcal{A} \backslash \mathcal{A}_2 = \mathcal{A}_1$.

- A node can succeed in two-hops if, in the *first* phase, it connected to a node in $\mathcal{A}_2$, so its message can be relayed in the second phase. These nodes are denoted by $\mathcal{B}_1$ in Fig. 10. The third method is the only means of succeeding in the second phase, as we are in the case where the rate can only increase, so no new links will be formed.

We now derive the probability that a node is not in any of the above sets. We first expand the quantity we wish to compute into a form that is simpler to work with.

$$\begin{aligned} P(\text{fail, 2U case 1}) &= P(\text{fail 2U } | \text{case 1}) \cdot P(\text{case 1}) \\ &= \sum_{a=0}^{a_0-1} P(\text{fail 2U} | A = a) \cdot P(A = a) \\ &= \sum_{a=0}^{a_0-1} \sum_{a_2=0}^{a} P(\text{fail to reach } \mathcal{A}_2 | A = a, A_2 = a_2) \cdot P(A_2 = a_2 | A = a) \cdot P(A = a) \end{aligned}$$

Conditioned on the events that occurred in Phase I, i.e., given some realization of A and A_2, a failure occurs when a node in $S \backslash A$ fails to reach any of the nodes in A_2 under R_{U_1}. This can be expressed with Eq. (3), as $P(\text{fail to reach } \mathcal{A}_2 | A = a, A_2 = a_2) = F(M_U, p_{U_1}^{a_2})$ where $M_U = n - a$.

Given that $A = a$ nodes succeeded in the first phase, we can calculate the probability of $A_2 = a_2$ by treating the probability of a given link failing as being distributed Bernoulli$(1 - q)$. Using Eq. (4), we get $P(A_2 = a_2 | A = a) = B\left(a, a_2, q^{(a)}\right)$.

The probability that $A = a$ is then distributed as a binomial distribution, just as $A = a$ in the downlink case, meaning $P(A = a) = B(n, a, p_{U_1})$.

This gives us the first portion of Theorem 4, the probability of failure in a two-hop uplink scheme:

$$P(\text{fail 2U, case 1}) = \sum_{a=0}^{a_0-1} \sum_{a_2=0}^{a} \left\{ F\left(M_U, p_{U_1}^{a_2}\right) B\left(a, a_2, q^{(a)}\right) \cdot B(n, a, p_{U_1}) \right\}$$

where $M_U = n - a$.

[10]Recall that the fading distributions are assumed to be Rayleigh. Hence $q = P(C < R_{U_2}^{(a)} | C > R_{U_1}) = \frac{P(R_{U_1} < C < R_{U_2}^{(a)})}{P(C < R_{U_1})}$. Then we use Eq. (4) to get the final expression.

Case 2: $R_{U_2}^{(a)} < R_{U_1}$

We are interested in the event that $R_{U_2}^{(a)} < R_{U_1}$. This case arises when $A = a > a_0$. Here, some new links may have been added to the system with probability[11] $\tilde{q}^{(a)} = P\left(R_{U_2}^{(a)} < C | R_{U_1} > C\right) = 1 - \frac{p_{U_2}^{(a)}}{p_{U_1}}$. Let $\mathcal{B}_2$ (cardinality b_2) be the nodes in $S \backslash \mathcal{A}$ that can directly reach the controller in Phase II.

Fig. 11 portrays all possible paths of success. In order to succeed, a node must fall under one of three categories.

- A node may succeed directly in the first hop (is in $\mathcal{A}$). In this case, links cannot go bad, so we will never have a set of nodes which we have denoted by Set $\mathcal{A}_1$ which lose connection to the controller .
- A node may also succeed in the second phase by being able to connect to the controller under the new, lower rate (is in $\mathcal{B}_2$), even if it did not connect to the controller under the first rate.
- A node can succeed in two-hops by reaching any other node in $\mathcal{A}_2$ or $\mathcal{B}_2$ in the first hop, and having its message relayed to the controller in the second hop (is in $\mathcal{B}_1$ in Fig. 11).

We derive the probability that a node does not connect to the controller in any of the above ways. We first expand the quantity we wish to compute into a form that is simpler to work with.

$$\begin{aligned} P(\text{fail 2U, case 2}) &= P(\text{fail 2U } | \text{case 2}) \cdot P(\text{case 2}) \\ &= \sum_{a=a_0}^{n-1} P(\text{fail 2U} | A_2 = a) \cdot P(A_2 = a) \\ &= \sum_{a=a_0}^{n-1} \sum_{b_2=0}^{M_U-1} P(\text{fail to reach } \{\mathcal{A}_2, \mathcal{B}_2\} | A_2 = a, B_2 = b_2) \cdot P(B_2 = b_2, A_2 = a) \end{aligned}$$

where $M_U = n - a$.

The first term in the final expression corresponds to failing to reach a previously successful node in Phase I. Given some instantiation of A_2 and B_2, the probability that a node fails to reach the controller is the probability that it failed to reach any of the nodes in set $\mathcal{A}_2$ and $\mathcal{B}_2$ under the first rate. This is distributed Bernoulli with parameter $p_{U_1}^{a+b_2}$, so the probability that

[11] Recall that the fading distributions are assumed to be Rayleigh. Hence $\tilde{q}^{(a)} = P\left(R_{U_2}^{(a)} < C | R_{U_1} > C\right) = \frac{P\left(R_{U_2}^{(a)} < C < R_{U_1}\right)}{P(C < R_{U_1})}$. Then we use Eq. (4) to get the final expression.

at least one node failed to reach the controller after two-hops can be expressed with Eq. (3) as $P(\text{fail to reach } \{\mathcal{A}_2, \mathcal{B}_2\} | A_2 = a, B_2 = b_2) = F(M_U - b_2, p_{U_1}^{a+b_2})$.

The probability of a node succeeding directly to the controller under $R_{U_2}^{(a)}$ given it was not in $\mathcal{A}_2$ is $\tilde{q}^{(a)}$, so the probability that $B_2 = b_2$ given $A_2 = a$ can be written with Eq. (4) as $P(B_2 = b_2 | A_2 = a) = B\left(M_U, b_2, 1 - \tilde{q}^{(a)}\right)$.

The probability that $A_2 = a$ is exactly as in the first case, as Set $\mathcal{A}_2$ is the set of nodes that were able to successfully transmit their message to the controller in Phase I. This gives us $B(n, a, p_{U_1})$, completing the second portion of Theorem 4 as follows.

$$P(\text{fail 2U, case 2}) = \sum_{a=a_0}^{n-1} \sum_{b_2=0}^{M_U-1} F(M_U - b_2, p_{U_1}^{a+b_2}) B\left(M_U, b_2, 1 - \tilde{q}^{(a)}\right) B(n, a, p_{U_1})$$

where $M_U = n - a$.

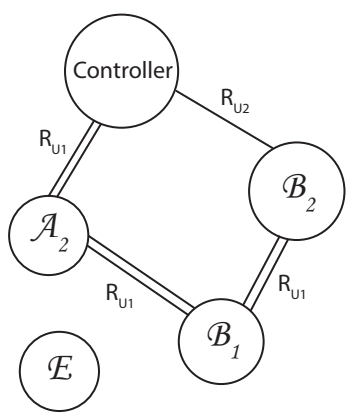


Fig. 11: This figure depicts the only ways to succeed in two-hop uplink, given that the second phase rate is lower. They are: to have a direct connection to the controller under any of the two rates, or to have connected, in the first phase (double lines), to a node that can succeed via a direct link to the controller. Please refer to Fig. 7 to recall the exact meaning of each set name.

The probability of failure of the two-hop uplink protocol is then given by the following expression, where the first term comes from case 1, and the second is from case 2.

$$P(\text{fail 2U}) = \sum_{a=0}^{a_0-1} \sum_{a_2=0}^{a} F(M_U, p_{U_1}^{a_2}) B\left(a, a_2, q^{(a)}\right) \cdot B(n, a, p_{U_1}) + \sum_{a=a_0}^{n-1} \sum_{b_2=0}^{M_U-1} F(M_U - b_2, p_{U_1}^{a+b_2}) B\left(M_U, b_2, 1-\tilde{q}^{(a)}\right) B(n, a, p_{U_1}) \tag{10}$$

where $M_U = n - a$. ■

C. Three-Hop Protocol

The completed protocol depicted in Fig. 1 is a three-hop protocol, where both the controller and nodes get three chances to get their message across. The total time for downlink and uplink are optimally divided between the three phases to minimize the SNR required to attain a target probability of error.

1) Three-Hop Downlink:

Theorem 5: Let the Phase I, Phase II and Phase III downlink time be T_{D_1}, T_{D_2} and T_{D_3} respectively, number of non-controller nodes be n, and message size be m. The Phase I transmission rate is given by $R_{D_1} = \frac{m \cdot n}{T_{D_1}}$, and the corresponding probability of a single link failure, p_{D_1}, is given by eq(4). The Phase II and Phase III transmission rate is given by $R_{D_2}^{(a)} = \frac{m \cdot (n-a)}{T_{D_2}} + \frac{2n}{T_{D_2}}$, and $R_{D_3}^{(a)} = \frac{m \cdot (n-a)}{T_{D_3}} + \frac{2n}{T_{D_3}}$ where a is the number of "successful nodes" in Phase I, and the corresponding probability of a single failure, p_{D_2} and p_{D_3}, is given by eq(4). The probability 3-hop downlink failure is then

$$P(\text{fail, 3D}) = \sum_{a=0}^{n-1} \sum_{b=0}^{M_D-1} B(n, a, p_{D_1}) B\left(M_D, b, \left(p_{D_2}^{(a)}\right)^a q_{21}^{(a)}\right) F\left(M_D - b, \left(p_{D_3}^{(a)}\right)^b \left(q_{32}^{(a)}\right)^a q_{321}^{(a)}\right) \tag{11}$$

where,

- $M_D = n - a$
- $q_{21}^{(a)} = P\left(C < R_{D_2}^{(a)} | C < R_{D_1}\right) = \min\left(\frac{p_{D_2}^{(a)}}{p_{D_1}}, 1\right)$
- $q_{32}^{(a)} = P\left(C < R_{D_3}^{(a)} | C < R_{D_2}^{(a)}\right) = \min\left(\frac{p_{D_3}^{(a)}}{p_{D_2}^{(a)}}, 1\right)$
- $q_{321}^{(a)} = P\left(C < R_{D_3}^{(a)} | C < \min(R_{D_1}, R_{D_2}^{(a)})\right) = \min\left(\max\left(\frac{p_{D_3}^{(a)}}{p_{D_1}}, \frac{p_{D_3}^{(a)}}{p_{D_2}^{(a)}}\right), 1\right)$

Proof: The rate of transmission in Phase I, R_{D_1}, is determined by the time allocated for this phase, T_{D_1}. Let the nodes who were successful in Phase I be in Set $\mathcal{A}$ (cardinality a). The rate in Phase II, $R_{D_2}^{(a)}$ and Phase III, $R_{D_3}^{(a)}$ depends on the realization of a, and the time allocated for the

phase, T_{D_2} and T_{D_3}. As before, $R_{D_2}^{(a)} = \frac{m\cdot(n-a)}{T_{D_2}} + \frac{2n}{T_{D_2}}$, $R_{D_3}^{(a)} = \frac{m\cdot(n-a)}{T_{D_3}} + \frac{2n}{T_{D_3}}$. The probabilities of link error corresponding to each rate R_{D_1}, $R_{D_2}^{(a)}$ and $R_{D_3}^{(a)}$ are p_{D_1}, $p_{D_2}^{(a)}$ and $p_{D_3}^{(a)}$ respectively. Fig. 12 displays an exhaustive list of ways to succeed in a three-hop downlink protocol.

- A node can succeed directly from the controller in the first hop under rate R_{D_1} (is in Set $\mathcal{A}$).
- A node can succeed in the second phase of the protocol by either hearing directly from the controller under the new rate, $R_{D_2}^{(a)}$, or by hearing the message from one of the nodes in Set $\mathcal{A}$ (is in Set $\mathcal{B}$).
- A node can succeed in the third phase from any of the nodes in Set $\mathcal{B}$ or Set $\mathcal{A}$ (if $R_{D_3}^{(a)} < R_{D_2}^{(a)}$) or directly from the controller (if $R_{D_3}^{(a)} < \min(R_{D_2}^{(a)}, R_{D_1})$).

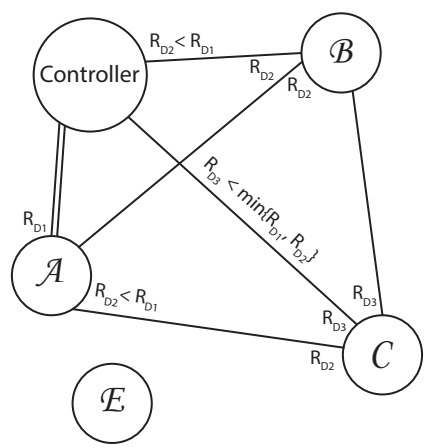


Fig. 12: In this figure, the only ways to succeed in a three-hop downlink protocol are displayed. A node can succeed in the first phase directly from the controller, in Phase II from either the controller or someone who succeeded in Phase I, and in Phase III from someone who succeeded in Phase II. Please refer to Fig. 7 to recall the exact meaning of each set name.

In order to calculate the probability of error of a three-hop downlink protocol, we will unroll the state space in a manner similar to the two-hop derivations. To calculate the overall probability of failure in 2-hop downlink, we sum over all possible instantiations of the sets of interest that result in failure. In this case, we are interested in the event that at least one node, which does not fall in Sets $\mathcal{A}$ and $\mathcal{B}$, is also not in $\mathcal{C}$ (fails given the instantiations of set $\mathcal{A}$ and $\mathcal{B}$).

$$P(\text{fail, 3D}) = \sum_{a=0}^{n-1} \sum_{b=0}^{M_a-1} P(\text{fail}|A=a, B=b)P(B=b|A=a)P(A=a)$$

where $M_D = n - a$.

Given $B = b$ and $A = a$, the probability of a node (not in $\mathcal{A}$ or $\mathcal{B}$) failing after three-hops is the probability that it cannot receive its message from either a node in Set $\mathcal{B}$ or Set $\mathcal{A}$ (if $R_{D_3}^{(a)} < R_{D_2}^{(a)}$) or directly from the controller (if $R_{D_3}^{(a)} < \min(R_{D_2}^{(a)}, R_{D_1})$). This is distributed Bernoulli $\left(p_{D_3}^{(a)}\right)^b \cdot \left(q_{32}^{(a)}\right)^a \cdot q_{321}^{(a)}$, and can be written with Eq. (3) as $F\left(n-(a+b), \left(p_{D_3}^{(a)}\right)^b \cdot \left(q_{32}^{(a)}\right)^a \cdot q_{321}^{(a)}\right) = F\left(M_D - b, \left(p_{D_3}^{(a)}\right)^b \cdot \left(q_{32}^{(a)}\right)^a \cdot q_{321}^{(a)}\right)$.

Given $A = a$, we can calculate the probability of a node not succeeding in Phase II as $\left(p_{D_2}^{(a)}\right)^a q_{21}^{(a)}$, as it must fail to receive its message from all of the nodes in Set $\mathcal{A}$, and from the controller under the phase II rate. Hence we calculate the probability that $B = b$ using a binomial distribution with parameter $\left(p_{D_2}^{(a)}\right)^a \cdot q_{21}^{(a)}$ as $B\left(M_D, b, \left(p_{D_2}^{(a)}\right)^a \cdot q_{21}^{(a)}\right)$

The probability of $A = a$ is exactly the same as we have seen before, at it relies on just point to point links to the controller, each of which fails with probability p_{D_1} (we use Eq. (4)). This gives us $B(n, a, p_{D_1})$.

Therefore, the probability of failure of the 3-phase downlink protocol is given by

$$\begin{aligned} P(\text{fail, 3D}) &= \sum_{a=0}^{n-1} \sum_{b=0}^{M_D-1} P(A=a)P(B=b|A=a)P(\text{fail}|A=a, B=b) \\ &= \sum_{a=0}^{n-1} \sum_{b=0}^{M_D-1} B(n, a, p_{D_1}) B\left(M_D, b, \left(p_{D_2}^{(a)}\right)^a q_{21}^{(a)}\right) F\left(M_D - b, \left(p_{D_3}^{(a)}\right)^b \left(q_{32}^{(a)}\right)^a q_{321}^{(a)}\right) \end{aligned}$$

where $M_D = n - a$.

■

2) Three-Hop Uplink:

Theorem 6: Let the Phase I, Phase II and Phase III uplink time be T_{U_1}, T_{U_2} and T_{U_3} respectively, number of non-controller nodes be n, and message size be m. The Phase I transmission rate is given by $R_{U_1} = \frac{m \cdot n}{T_{U_1}}$. The Phase II and Phase III transmission rate is given by $R_{U_2}^{(a)} =$

$\frac{m\cdot(n-a)}{T_{U_2}} + \frac{2n}{T_{U_2}}$, and $R_{U_3}^{(a)} = \frac{m\cdot(n-a)}{T_{U_3}} + \frac{2n}{T_{U_3}}$ where a is the number of "successful nodes" in Phase I. The probability of cycle failure is then

$$
\begin{aligned}
P(\text{fail, 3U}) = \sum_{a=0}^{n-1} \Bigg[& \left(\sum_{b_2=0}^{n-a-1} \sum_{b_1=0}^{n-a-b_2-1} \sum_{c_3=0}^{n-a-b-1} \sum_{c_2=0}^{n-a-b-c_3-1} P(\text{fail}_1) \right) \mathbb{1}\left(R_{U_1} \geqslant R_{U_2} > R_{U_3}\right) \\
& + \left(\sum_{b_2=0}^{n-a-1} \sum_{b_1=0}^{n-a-b_2-1} \sum_{\widehat{b}_2=0}^{b_2} \sum_{\widehat{b}_1=0}^{b_1} \sum_{c_2=0}^{n-a-b-1} P(\text{fail}_2) \right) \mathbb{1}\left(R_{U_1} > R_{U_3} \geqslant R_{U_2}\right) \\
& + \left(\sum_{a_3=0}^{a} \sum_{b_2=0}^{n-a-1} \sum_{b_1=0}^{n-a-b_2-1} \sum_{\widehat{b}_1=0}^{b_1} \sum_{c_2=0}^{n-a-b-1} P(\text{fail}_3) \right) \mathbb{1}\left(R_{U_3} \geqslant R_{U_1} > R_{U_2}\right) \\
& + \left(\sum_{a_2=0}^{a} \sum_{a_3=0}^{a_2} \sum_{\widehat{a}_1=0}^{a-a_2} \sum_{b_1=0}^{n-a-1} \sum_{\widehat{b}_1=0}^{b_1} P(\text{fail}_4) \right) \mathbb{1}\left(R_{U_3} > R_{U_2} \geqslant R_{U_1}\right) \\
& + \left(\sum_{a_2=0}^{a} \sum_{\tilde{a}=0}^{a-a_2} \sum_{\widehat{a}_1=0}^{a-a_2-\tilde{a}_1} \sum_{b_1=0}^{n-a-1} \sum_{\widehat{b}_1=0}^{b_1} P(\text{fail}_5) \right) \mathbb{1}\left(R_{U_2} \geqslant R_{U_3} > R_{U_1}\right) \\
& + \left(\sum_{a_2=0}^{a} \sum_{b_1=0}^{n-a-1} \sum_{\widehat{b}_1=0}^{b_1} \sum_{c_3=0}^{n-a-b_1-1} \sum_{c_2=0}^{n-a-b_1-c_3-1} \sum_{\widehat{c}_2=0}^{c_2} P(\text{fail}_6) \right) \mathbb{1}\left(R_{U_2} > R_{U_1} \geqslant R_{U_3}\right) \Bigg]
\end{aligned}
\tag{12}
$$

where

$$
\begin{aligned}
P(\text{fail}_1) = {} & F\left(n-a-b-c_2-c_3, p_1^{b_1+c_2}\right) \times B\left(n-a-b-c_3, c_2, q_{21}^{a+b_2+c_3}\right) \times \\
& \times B\left(n-a-b, c_3, q_{32}\right) \times B\left(n-a-b_2, b_1, p_1^{a+b_2}\right) \times B\left(n-a, b_2, q_{21}\right) \times B(n, a, p_1)
\end{aligned}
$$

is the probability of failure of the 3-hop uplink protocol if the relationship between the rates is $R_{U_1} \geqslant R_{U_2} > R_{U_3}$,

$$
\begin{aligned}
P(\text{fail}_2) = {} & F\left(n-a-b-c_2, p_1^{\widehat{b}_1+c_2}\right) \times B\left(n-a-b, c_2, q_{21}^{a+\widehat{b}_2}\right) \times \\
& \times B\left(b_1, \widehat{b}_1, s_{22}[a+\hat{b}_2, a+b_2]\right) \times B\left(b_2, \hat{b}_2, r_{32}\right) \times B\left(n-a, b_2, q_{21}\right) \times B(n, a, p_1)
\end{aligned}
$$

is the probability of failure of the 3-hop uplink protocol if the relationship between the rates is $R_{U_1} > R_{U_3} \geqslant R_{U_2}$,

$$
\begin{aligned}
P(\text{fail}_3) = {} & F\left(n-a-b-c_2, p_1^{\widehat{b}_1+c_2}\right) \times B\left(n-a-b, c_2, q_{21}^{a_3}\right) \times B\left(b_1, \widehat{b}_1, s_{22}[a_3, a+b_2]\right) \times \\
& \times B\left(n-a-b_2, b_1, p_1^{a+b_2}\right) \times B\left(a, a_3, r_{31}\right) \times B\left(n-a, b_2, q_{21}\right) \times B(n, a, p_1)
\end{aligned}
$$

is the probability of failure of the 3-hop uplink protocol if the relationship between the rates is $R_{U_3} \geqslant R_{U_1} > R_{U_2}$,

$$P(\text{fail}_4) = F\left(n-a-b_1, p_1^{\widehat{b}_1}\right) \times B\left(b_1, \widehat{b}_1, s_{21}[a_3, a_2]\right) \times B\left(n-a, b_1, p_1\right) \times B\left(a_2, a_3, r_{32}\right) \times \\ \times B\left(a, a_2, r_{21}\right) \times B(N, a, p_1)$$

is the probability of failure of the 3-hop uplink protocol if the relationship between the rates is $R_{U_3} > R_{U_2} \geqslant R_{U_1}$,

$$P(\text{fail}_5) = F\left(n-a-b_1, p_1^{\widehat{a}_1+\widehat{b}_1}\right) \times B\left(a-\widetilde{a}_1-a_2, \widehat{a}_1, p_2^{\widetilde{a}_1+a_2}\right) \times B\left(b_1, \widehat{b}_1, s_{21}[a_2, a_2]\right) \times \\ \times B\left(n-a, b_1, p_1^{a_2}\right) \times B\left(a-a_2, \widetilde{a}_1, m_{312}\right) \times B\left(a, a_2, r_{21}\right) \times B(n, a, p_1)$$

is the probability of failure of the 3-hop uplink protocol if the relationship between the rates is $R_{U_2} \geqslant R_{U_3} > R_{U_1}$,

$$P(\text{fail}_6) = F\left(n-a-b-c_2-c_3, p_1^{\widehat{b}_1+\widehat{c}_2}\right) \times B\left(c_2, \widehat{c}_2, s_{21}[a+c_3, a+c_3]\right) \times B\left(b_1, \widehat{b}_1, s_{21}[a_2, a_2]\right) \times \\ \times B(n-a-b-c_3, c_2, p_1^{a_1+c_3}) \times B\left(n-a, b_1, p_1^{a_2}\right) \times B\left(a, a_2, r_{21}\right) \times B(n, a, p_1)$$

is the probability of failure of the 3-hop uplink protocol if the relationship between the rates is $R_{U_2} > R_{U_1} \geqslant R_{U_3}$, and,

- $p_1 = p_{U_1} = P(C < R_{U_1})$
- $p_2 = p_{U_2}^{(a)} = P(C < R_{U_2}^{(a)})$
- $p_3 = p_{U_3}^{(a)} = P(C < R_{U_3}^{(a)})$
- $q_{21} = P(C < R_{U_2}^{(a)} | C < R_{U_1})$
- $q_{31} = P(C < R_{U_3}^{(a)} | C < R_{U_1})$
- $q_{32} = P(C < R_{U_3}^{(a)} | C < R_{U_2}^{(a)})$
- $r_{21} = P(C < R_{U_2}^{(a)} | C > R_{U_1})$
- $r_{31} = P(C < R_{U_3}^{(a)} | C > R_{U_1})$
- $r_{32} = P(C < R_{U_3}^{(a)} | C > R_{U_2}^{(a)})$
- $m_{312} = P(C < R_{U_3}^{(a)} | R_{U_1} < C < R_{U_2}^{(a)})$
- $s_{ij}[f, g] = (1-p_i^f)/(1-p_j^g)$ where f and g are cardinalities of sets F and G.
- $b = b_1 + b_2$

Proof: The proof of the theorem is slightly involved and lengthy. Here we will describe Case 2: $(R_{U_1} > R_{U_3} \geqslant R_{U_2})$ to illustrate some of the nuanced effects that happen in uplink. The descriptions of other cases can be found in [40].

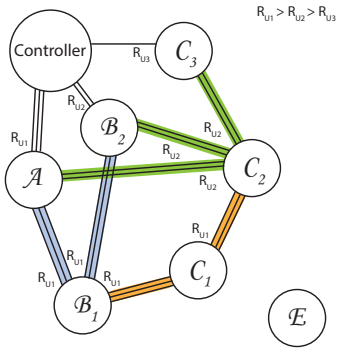
The rate of transmission in Phase I, R_{U_1}, is determined by the time allocated for this phase, T_{U_1}. Let the nodes who were successful in Phase I be in Set $\mathcal{A}$ (cardinality a). The rate in Phase

II, $R_{U_2}^{(a)}$ and Phase III, $R_{U_3}^{(a)}$ depends on the realization of a, and the time allocated for the phase, T_{U_2} and T_{U_3}. As before, $R_{U_2}^{(a)} = \frac{m\cdot(n-a)}{T_{U_2}} + \frac{2n}{T_{U_2}}$, $R_{U_3}^{(a)} = \frac{m\cdot(n-a)}{T_{U_3}} + \frac{2n}{T_{U_3}}$. The probabilities of link error corresponding to each rate R_{U_1}, $R_{U_2}^{(a)}$ and $R_{U_3}^{(a)}$ are p_{U_1}, $p_{U_2}^{(a)}$ and $p_{U_3}^{(a)}$ (abbreviated to p_1, p_2 and p_3) respectively.

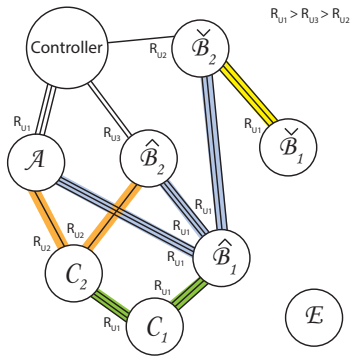
Fig. 13b displays an exhaustive list of ways to succeed in case 2 of three-hop uplink protocol.

- A node can succeed directly from the controller in the first hop under rate R_{U_1} (is in set $\mathcal{A}$).
- A node can succeed in the second phase of the protocol by connecting directly to the controller under the new rate, $R_{U_2}^{(a)}$ (is in set $\mathcal{B}_2$). This set is then segregated into two disjoint sets: $\widehat{\mathcal{B}}_2$ which retain links to the controller in the third phase and $\check{\mathcal{B}}_2$ which lose links to the controller in the third phase.
- A node can succeed in the second phase of the protocol by connecting in the first phase (these nodes are in set $\mathcal{B}_1$) to one of the nodes in the set $\mathcal{A}\bigcup\mathcal{B}_2$ (the set of nodes which can communicate to the controller in phase II). This ensures that the nodes which can connect to the controller in the second phase already have the message. This set is then segregated into two disjoint sets: $\widehat{\mathcal{B}}_1$ which has good links to the set which has link to controller in the third phase (set $\mathcal{A}\bigcup\widehat{\mathcal{B}}_2$) and $\check{\mathcal{B}}_1$ which does not have link to the set which has link to controller in the third phase (set $\mathcal{A}\bigcup\widehat{\mathcal{B}}_2$). Thus set $\check{\mathcal{B}}_1$ cannot act as relay for three-hop successes.
- A node can succeed in the third phase in a two-hop fashion by connecting to the set $\mathcal{A}\bigcup\widehat{\mathcal{B}}_2$ under the lower phase two rate $R_{U_2}^{(a)}$ (is in set $\mathcal{C}_2$). The set $\mathcal{A}\bigcup\widehat{\mathcal{B}}_2$ is the set of nodes which can connect to the controller in the third phase. Connecting to this set in the second phase ensures that the message to be conveyed in the third phase has been conveyed to the relays by the second phase.
- A node can succeed in the third phase in a three-hop fashion by connecting to the set $\mathcal{C}_2\bigcup\widehat{\mathcal{B}}_1$ in the first phase under rate R_{U_1} (is in set $\mathcal{C}_1$). The set $\mathcal{C}_2\bigcup\widehat{\mathcal{B}}_1$ is the set of nodes which can connect to the set $\mathcal{A}\bigcup\widehat{\mathcal{B}}_2$ (the set which can connect to the controller in the third phase) in the second phase. Connecting to this set in the first phase ensures that the message to be conveyed in the third phase has been conveyed to the right relays by the second phase.

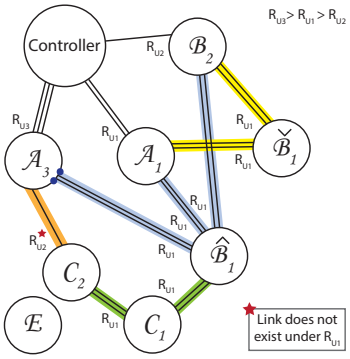
To calculate the probability of error of a three-hop uplink protocol, we will unroll the state



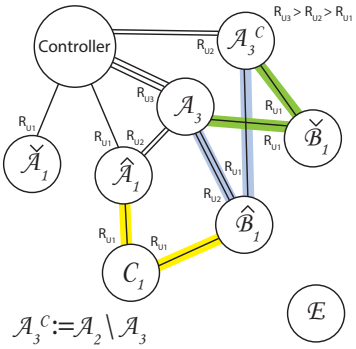
(a) Case 1: $R_{U_1} \geqslant R_{U_2} > R_{U_3}$. The only ways to succeed in the 1st case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to the controller or a node which can succeed in Phase II, and in Phase III by directly connecting to the controller or connecting to the nodes which have connections to the controller in Phase II (thus succeeding in 2 hops) or connecting via 2 hops to the nodes which have connections to the controller (thus succeeding in 3 hops).



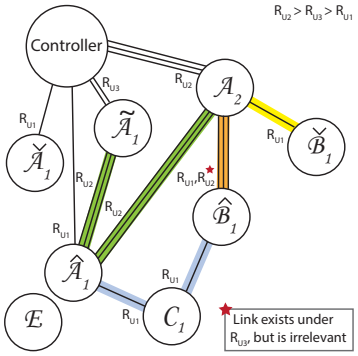
(b) Case 2: $R_{U_1} > R_{U_3} \geqslant R_{U_3}$. The only ways to succeed in the 2nd case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to the controller or a node which can succeed in Phase II, and in Phase III by connecting directly to the nodes which have connections to the controller in Phase II (thus succeeding in 2 hops) or connecting via 2 hops to the nodes which have connections to the controller (thus succeeding in 3 hops).



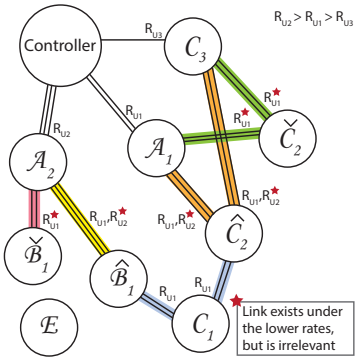
(c) Case 3: $R_{U_3} \geqslant R_{U_1} > R_{U_2}$. The only ways to succeed in the 3rd case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to the controller or a node which can succeed in Phase II, and in Phase III by connecting directly to the nodes which have connections to the controller in Phase II (thus succeeding in 2 hops) or connecting via 2 hops to the nodes which have connections to the controller (thus succeeding in 3 hops).



(d) Case 4: $R_{U_3} > R_{U_2} \geqslant R_{U_1}$: The only ways to succeed in the 4th case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to a node which can succeed in Phase II, and in Phase III by connecting via 2 hops to the nodes which have connections to the controller (thus succeeding in 3 hops).



(e) Case 5: $R_{U_2} \geqslant R_{U_3} > R_{U_1}$: The only ways to succeed in the 5th case of three-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to a node which can succeed in Phase II, and in Phase III by connecting via 2 hops to the nodes which have connections to the controller (thus succeeding in 3 hops).



(f) Case 6: $R_{U_2} > R_{U_1} \geqslant R_{U_3}$: The only ways to succeed in the 6th case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to a node which can succeed in Phase II, and in Phase III by directly connecting to the controller or connecting to the nodes which have connections to the controller in Phase II (thus succeeding in 2 hops) or connecting via 2 hops to the nodes which have connections to the controller (thus succeeding in 3 hops).

Fig. 13: The different ways to succeed in the three-hop uplink protocol.

space in a manner similar to the three-hop downlink derivations. We sum over all possible instantiations of the sets of interest that result in failure to calculate the overall probability of failure. In this case, we are interested in the event that at least one node which does not fall in sets $\mathcal{A}$, $\mathcal{B} = \mathcal{B}_1 \bigcup \mathcal{B}_2$, $\mathcal{C}_2$ and is also not in $\mathcal{C}_1$ (fails given the instantiations of set $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}_1$).

The probability of $A = a$ is exactly the same as we have seen before, at it relies on just point to point links to the controller, each of which fails with probability $p_1 = p_{U_1}$ (we use Eq. (4)). This gives us $B(n, a, p_1)$.

Given $A = a$, we can calculate the probability of a node not being able to gain a connection to the controller in the second phase given there was no connection in the first phase as $q_{21} = P(C < R_{U_2}^{(a)} | C < R_{U_1}^{(a)}) = (p_2)/(p_1)$. The set which can connect to the controller in the second phase is $\mathcal{B}_2$. Hence we calculate the probability that $B_2 = b_2$ using a binomial distribution with parameter q_{21} as $B(N - a, b_2, q_{21})$.

Given $A = a$, $B_2 = b_2$, we can calculate the probability of a node in $\mathcal{B}_2$ losing connection to the controller in the third phase as $r_{32} = P(C < R_{U_3}^{(a)} | C > R_{U_2}^{(a)}) = (p_3 - p_2)/(1 - p_2)$. This set is denoted as $\breve{\mathcal{B}}_2$ and the set that retains the link is denoted as $\widehat{\mathcal{B}}_2$. Hence we calculate the probability that $\widehat{B}_2 = \widehat{b}_2$ using a binomial distribution with parameter r_{32} as $B(b_2, \hat{b}_2, r_{32})$.

Given $A = a$, $B_2 = b_2$, $\widehat{B}_2 = \widehat{b}_2$, and $B_1 = b_1$ we can calculate the probability of a node in $\mathcal{B}_1$ being only connected to $\breve{\mathcal{B}}_2$ in the second phase given it connected to the set $\breve{\mathcal{B}}_2 \bigcup \widehat{\mathcal{B}}_2 \bigcup \mathcal{A}$ as $s_{22}[a + \hat{b}_2, a + b_2] = (1 - p_2^{a+\hat{b}_2})/(1 - p_2^{a+b_2})$. Hence we calculate the probability that $\breve{B}_1 = \breve{b}_1$ using a binomial distribution with parameter $s_{22}[a + \hat{b}_2, a + b_2]$ as $B(b_1, \widehat{b}_1, s_{22}[a + \hat{b}_2, a + b_2])$.

Given $A = a$, $B_2 = b_2$, $\widehat{B}_2 = \widehat{b}_2$, we can calculate the probability of a node not succeeding in Phase II in two hops as $p_1^{a+b_2}$, as it must fail to connect to $\mathcal{A} \bigcup \mathcal{B}_2$ in the first phase. Hence we calculate the probability that $B_1 = b_1$ using a binomial distribution with parameter $p_1^{a+b_2}$ as $B(N - a - b_2, b_1, p_1^{a+b_2})$.

Given $A = a$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$, $B_2 = b_2$, $\widehat{B}_2 = \widehat{b}_2$, we can calculate the probability of a node not succeeding in Phase III in two hops as $q_{21}^{a+\hat{b}_2}$, as it must fail to connect to $\mathcal{A} \bigcup \widehat{\mathcal{B}}_2$ in the second phase having failed to connect in the first phase already. Hence we calculate the probability that $C_2 = c_2$ using a binomial distribution with parameter $q_{21}^{a+\hat{b}_2}$ as $B(N - a - b, c_2, q_{21}^{a+\hat{b}_2})$.

Given $C_2 = c_2$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$, $B_2 = b_2$, $\widehat{B}_2 = \widehat{b}_2$ and $A = a$, the probability of a node (not in $\mathcal{A} \bigcup \mathcal{B}_1 \bigcup \mathcal{B}_2 \bigcup \mathcal{C}_2$) failing after three-hops is the probability that it cannot connect to $\mathcal{C}_2 \bigcup \widehat{\mathcal{B}}_1$ in the first phase. This is distributed Bernoulli $p_1^{\hat{b}_1 + c_2}$, and can be written with Eq. (3) as $F(n - a - b - c_2, p_1^{\hat{b}_1 + c_2})$.

Thus we have that given the realization $A = a$ and that the protocol falls under case 2: $R_{U_1} > R_{U_3} > R_{U_2}$ is given by

$$P(\text{fail}|\text{Case 2}, A = a) = \left(\sum_{b_2=0}^{n-a-1} \sum_{b_1=0}^{n-a-b_2-1} \sum_{\widehat{b}_2=0}^{b_2} \sum_{\widehat{b}_1=0}^{b_1} \sum_{c_2=0}^{n-a-b-1} P(\text{fail}_2) \right)$$

where

$$P(\text{fail}_2) = F(n-a-b-c_2, p_1^{\widehat{b}_1+c_2}) \times B(N-a-b, c_2, q_{21}^{a+\widehat{b}_2}) \ldots$$
$$\times B(b_1, \widehat{b}_1, s_{22}[a+\hat{b}_2, a+b_2]) \times B(b_2, \hat{b}_2, r_{32}) \times B(N-a, b_2, q_{21}) \times B(N, a, p_1)$$

The realizations of the states in other cases is given in Fig. 13a, 13c, 13d, 13e and 13f. ■

REFERENCES

[1] 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.

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

[3] "SERCOS news, the automation bus magazine." [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 *Communications (ICC), 2015 IEEE International Conference on*, June 2015, pp. 4380–4386.

[5] E. V. Buskirk, "'Inhuman Microphone' app circumvents occupy wall street megaphone ban," 2011. [Online]. Available: http://www.wired.com/2011/12/inhuman-microphone/

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

[7] S. K. Sen, *Fieldbus and Networking in Process Automation*. CRC Press, 2014.

[8] A. Willig *et al.*, "Wireless Technology in Industrial Networks," in *Proceedings of the IEEE*, vol. 93, no. 6, June 2005, pp. 1130–1151.

[9] 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.

[10] 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.

[11] 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.

[12] G. Cena *et al.*, "Hybrid wired/wireless networks for real-time communications," *Industrial Electronics Magazine, IEEE*, vol. 2, no. 1, pp. 8–20, Mar 2008.

[13] I. F. Akyildiz *et al.*, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar 2002.

[14] A. Bonivento *et al.*, "System Level Design for Clustered Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 3, pp. 202–214, Aug 2007.

[15] M. A. Yigitel *et al.*, "QoS-aware MAC Protocols for Wireless Sensor Networks: A Survey," *Comput. Netw.*, vol. 55, no. 8, pp. 1982–2004, June 2011.

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

[17] G. Scheible *et al.*, "Unplugged but connected [Design and implementation of a truly wireless real-time sensor/actuator interface]," *Industrial Electronics Magazine, IEEE*, vol. 1, no. 2, pp. 25–34, Summer 2007.

[18] V. Gungor *et al.*, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, Oct 2009.

[19] Z. A. Standard, "ZigBee PRO Specfication," October 2007.

[20] A. Kim *et al.*, "When HART goes wireless: Understanding and implementing the WirelessHART standard," in *IEEE International Conference on Emerging Technologies and Factory Automation*, 2008, pp. 899–907.

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

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

[23] 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.

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

[25] 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.

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

[27] 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.

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

[29] 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.

[30] 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.

[31] H. Rahul *et al.*, "SourceSync: A Distributed Wireless Architecture for Exploiting Sender Diversity," in *Proceedings of the ACM SIGCOMM 2010 Conference*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010, pp. 171–182.

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

[33] 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.

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

[35] 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.

[36] 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.

[37] A. Lozano *et al.*, "Non-peaky signals in wideband fading channels: Achievable bit rates and optimal bandwidth," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 1, pp. 246–257, January 2012.

[38] 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.

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

[40] V. N. Swamy *et al.*, “Wireless Communication for High-reliability Low-latency Control.” [Online]. Available: ARXIVLINK