Real-time Communication Systems For Automation Over Wireless: Enabling Future Interactive Tech

Vasuki Narasimha Swamy

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2018-114
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-114.html

August 10, 2018

Real-time Communication Systems For Automation Over Wireless: Enabling Future Interactive Tech

by

Vasuki Narasimha Swamy

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Prof. Anant Sahai, Chair
Prof. Steven Glaser
Prof. Borivoje Nikolić
Prof. Kristofer Pister
Prof. Gireeja Ranade

Summer 2018

The dissertation of Vasuki Narasimha Swamy, titled Real-time Communication Systems For Automation Over Wireless: Enabling Future Interactive Tech, is approved:

Chair _____ Date _____

_____ Date _____

_____ Date _____

_____ Date _____

_____ Date _____

University of California, Berkeley

Real-time Communication Systems For Automation Over Wireless: Enabling Future Interactive Tech

Abstract

Real-time Communication Systems For Automation Over Wireless: Enabling Future Interactive Tech

by

Vasuki Narasimha Swamy

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Prof. Anant Sahai, Chair

The density and frequency of interaction between society and technology is increasing and this presents us with opportunities to improve lives and livelihoods, and also to address the inequities in our society holistically. The Internet of Things (IoT) and 5G wireless communication technologies are the future technologies that envision to bring us closer to these opportunities. These technologies leverage the ubiquitous sensing, actuation, and computing capabilities to enable smart devices to perform interesting tasks and to gain knowledge about the environment. These may include futuristic healthcare systems, affordable precision agriculture systems, smart energy-efficient cities, and advanced flexible manufacturing. However, the present understanding of wireless communication is not enough to get us to this future.

In this dissertation, we look at designing wireless communication frameworks for interactive applications like drone swarms, and industrial automation that require fast and reliable communication. We focus on designing frameworks that make wireless communication highly reliable while also maintaining the latency requirements of the systems being targeted. This is the missing link needed to leverage the power of wireless communication for the next generation of interactive applications.

One of the key contributions of this thesis is the design of cooperative communication based protocol frameworks that leverage a combination of diversity techniques to achieve the target reliability and latency. The framework uses spatial diversity to combat multipath channel fading and repetitions in time and frequency to shield against unmodeled error events. We analyze these protocols using the communication-theoretic delay-limited-capacity framework and consider their sensitivity to different modeling assumptions.

Another key contribution of this thesis is an in-depth exploration of the dynamics of wireless channels in the context of ultra-reliable low-latency communication (URLLC). We revisit some standard concepts such as coherence time and question whether some of the modeling assumptions made in the context of cellular and WLAN communications make sense in the context of URLLC. We find that our cooperative communication frameworks

are robust to the nominal dynamic channel models, especially spatial dependence. However, events such as synchronization mismatch or sudden change in the channels due to shadow-causing objects need to be protected against and therefore we build in frequency and time margins.

The final contribution of this thesis is bringing together the temporal model of channel dynamics and machine learning to build intelligent relay selection strategies. This essentially provides the reliability needed by smartly selecting a small set of relays instead of relying on using *every single relay available* to combat fading. Finally, we present some preliminary experimental findings that pave the way to making these systems practical.

To all my teachers who have taught me valuable lessons and encouraged me to dream big.

Contents

List of Figures

List of Tables

Acknowledgments

There are several people from all different walks of my life that I would like to thank. First and foremost, my amazing advisor Prof. Anant Sahai. I have learned so much from him not only about research but also about teaching and social issues. Thank you so much for supporting me throughout this whole journey. Thanks to Prof. Gireeja Ranade who has been both a friend and a mentor throughout my time here who has taught me so much in research and guided me during difficult times. Thanks to Prof. Bora Nikolic, who has guided me to keep the big picture in mind while thinking about any problem. I also want to thank my committee members, Prof. Kris Pister and Prof. Steven Glaser for their valuable feedback and guiding me towards the right questions. Thanks to my co-authors – Paul Rigge, Sahaana Suri, Leah Dickstein and Matt Weiner. Working with them has been a pleasure.

Thanks to Prof. Kannan Ramchandran, my masters thesis advisor for his infectious enthusiasm and unending curiosity. I have been really fortunate to have gotten chances to explore new things during my time here which has shaped me quite a bit. I learned a lot from my internship at Qualcomm working with Venky Ekambaram and Jubin Jose. I had a wild time interning at Microsoft Research where I saw first hand what it was to work on something so out of the box. Not only did I learn a lot from my mentors Ranveer Chandra, Anirudh Badam, Sudipta Sinha and Gireeja Ranade but also a lot from my co-interns – Deepak Vasisht, Zerina Kapetanovic, Talal Ahmed and Amy Kumar.

EE16A has touched the lives of pretty much everyone in the EECS department at UC Berkeley. Being a TA for the pilot was a wild and fun journey and I learned so much from Prof. Elad Alon, Gireeja Ranade and my co-TAs – Paroma Varma, Nathan Mailoa and Vikram Iyer. I was also super fortunate to have gotten the chance to teach it last summer and I could not have done it without my colleagues Daniel Aranki and Filip Maksimovic.

Thanks to all the faculty of BLISS with whom I have had so many wonderful and insightful conversations. Thanks to all the students of BLISS and BWRC – Nihar Shah, Rashmi Vinayak, Gireeja Ranade, Kate Harrison, Giulia Fanti, Varun Jog, Po-Ling Jog Loh, Vijay Kamble, Stephan Adams, Venky Ekambaram, Sameer Pawar, Payam Delgosha, Orhan Ocal, Vidya Muthukumar, Ashwin Pananjady, Raaz Dwivedi, Soham Phade, Sang Min Han, Laura Brink, Yuting Wei, Fanny Yang, Ilan Shomorony, Eren Sasoglu, Regina Eckert, Emily Naviasky, Aaditya Ramdas for all the fun times and deep conversations. We have the most supportive and friendly department staff – thanks to Shirley Salanio, Audrey Sillers, Tiffany Reardon, Alex Burr, Eric Arvai, Mayra Rivera, Kim Kail, James Dunn, Fred Burghardt and Ajith Amerasekera at BWRC.

I have been very lucky to have met so many wonderful people through BiasBusters and to have played a part in some positive changes. Special thanks to Regina Eckert, Roel Dobbe, Emily Naviasky, Laura Brink, Sheila Humphreys, Audrey Sillers and Shirley Salanio for their continual work and support for this organization.

I have faced quite some setbacks during my time here. Starting with a near fatal accident in the first month of grad school which took a toll on my physical and mental well-being. But perhaps the hardest has been the diagnosis of auto-immune condition that made leading a

normal life very painful. Despite that, I have been very lucky to have had friends who cared for me and looked out for me when I was at my lowest – especially Vaishnavi Surendra, Debanjan Mukherjee, Sharanya Prasad, Avinash Bhardwaj, Madhavi Yenugula, Vikram Rao, Arun Chaganty, Pranesh Srinivasan and Samhita Kasula. No medication was helping and I had given up hope on living pain-free. But my therapist Dr. Daniel Goldstine had not yet given up. He went above and beyond to get me to see a good doctor and followed up to make sure I was getting good medical care and I am grateful to him for that. Along with medication, strength training with Heather Lockwood at the RSF has helped me get back to leading a normal life and she has been a great guide and friend in the past year. I am also grateful to have friends who have helped me through the stress of finishing up my thesis these last few months – Tejal Bhamre, Varun Pattabhiraman and Sudeep Srivastava. And facing all these hurdles would have been doubly difficult without Nishant Totla who continues to be my pillar of support. And last but not the least, my parents who have loved me and trusted me to make good decisions and let me explore and dream big while the society around them advised them not to.

The Thesis in a Nutshell

A new industrial revolution with an emphasis on the tight interaction between cyber-physical systems, Internet-of-Things, cloud computing and machine intelligence is here. On the one end of the spectrum of applications is precision agriculture that requires low-power, low-cost devices and long-range, sparse communication. On the other end of the spectrum are high-performance interactive and immersive applications that require low-latency, high-reliability and high-density of communication.

Engineering wireless communication systems to support the communication requirements of interactive and immersive applications will open up possibilities such as exoskeletons for health-care, inter-vehicle communication for self-driving cars and traffic management, robotics and factory automation, virtual and augmented reality, the smart grid and drone swarms. To meet the demands of such high performance IoT applications as well as to deliver an interactive cyber-sickness-free experience for human-in-the-loop applications, reliable communication with latencies of about 1ms is crucial. The operating region (in terms of latency and reliability) for high-performance immersive applications is very different from current technologies like WiFi and 4G where the main focus has been to increase spectral efficiency in the low-reliability, moderate-latency regime. Sensor network standards like WirelessHART prioritize energy efficiency and reliability. In this thesis, we study the largely unexplored space of low-latency and high-reliability suited for interactive applications.

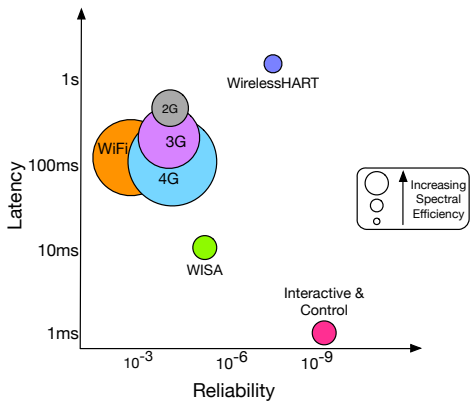


Figure 0.1: Comparison of some current wireless technologies on latency and reliability axis. Interactive and immersive applications that we target is depicted by the pink circle.

A popular motivating example for low-latency and high-reliability communication is industrial control. In industrial control, the communication requirements of high-reliability and low-latency are supported by wired fieldbus systems such as SERCOS III. The parameters associated with such applications include network sizes of 10 - 100 nodes, a steady stream of short control and sensor packets of about 10 - 50 bytes and latency of about 1ms.

They also require that the probability that a packet doesn't reach its destination before the deadline does not exceed 10^{-9}. We address the problem of engineering a drop-in wireless replacement for these wired communication systems. The four main goals of our design are the following: a) communication has to be *ultra-reliable*, b) the *low-latency* requirements have to be met, c) the design has to be *practically implementable*, and d) the protocol should *scale well* with increasing network size.
The main results of the thesis are the following:

- We design a wireless communication framework for URLLC that is intended as a drop-in replacement for wired protocols in place. The framework combines a variety of techniques to combat different kinds of challenges. We provide in-depth analysis of its performance using the delay-limited-capacity framework.
  - To fight fading, we primarily use spatial diversity techniques.
  - To fight local error-causing events such as mis-synchronized packets as well as global error-causing events like interference from a jammer, we use repetitions in time and frequency.
  - We find that further optimizing the protocol by incorporating network coding is advantageous but any other minute optimization such as rate adaptation does not provide benefits to offset the cost of information dissemination as well as designing advanced radios to perform variable rate decoding.
- We take an in-depth look at channel dynamics to study the fading events that may occur in timescales ranging from tens of microseconds (corresponding to the length of a single short packet) to a few milliseconds (corresponding to the cycle time).
  - We formulate the fading process as a Gaussian process using the Jakes's model. Theoretically, the covariance function is a Bessel function of the first kind and our simulations co-incide with the Bessel function, even for a small number of scatterers.
  - Although channels in Jakes's model are spatially correlated in principle, as long as antennas are multiple wavelengths apart, the correlation is slight, and can be offset by a fraction of a dB increase in transmit power.
  - We find that for short packets of duration under 100μs (motion under 0.01λ), channels that are good enough stay quite static. There is no large variation in channel energy within this time if the channels started out to be good. On the other hand, if channels were deeply faded to begin with, even minute changes in energy would manifest as large relative changes but this really has no impact on the performance as those were bad channels anyways.
  - We study the channel dynamics on the order of hundreds of microseconds to a few milliseconds which corresponds to the cycle time as well as relaying events.

We find that Rayleigh fading processes are not bandlimited. This has significant implications in channel quality prediction and relay selection techniques.

- As the focus is on ultra-reliable communication, we critically question how the framework would perform if the nominal channel models break down. We model the uncertainties with bounds on the following uncertainties:
  - the maximum probability of an unmodeled link error that is independent across transmitter/receiver pairs (i.e., from a different CDF for multipath fading).
  - the maximum probability of an unmodeled decoding error that is independent across time-slots (similarly we can think of errors to be correlated across time but independent across different frequencies).
  - the maximum probability of an unmodeled decoding error that compounds with the number of simultaneous transmitters, but is independent across time-slots.

  We use the above uncertainties combined with our nominal channel model to provide a robust model for wireless uncertainty.

- We leverage the channel dynamics knowledge to theoretically study relay selection performance under a Gaussian process model. We find that the most important parameters for prediction accuracy are: sampling frequency, future horizon, and number of potential relays to choose from. In a simulated setup we consider different relay quality prediction schemes including the state-of-the-art static predictor that does not consider channel dynamics, simple polynomial based predictor as well as linear predictor based on the theoretical Gaussian process model, and a neural-network-based predictor that leverages the non-linearity of channel quality progression. The predictors predict the channel quality and we employ one of the standard relay-selection techniques from literature to choose a relay from the set of relays available. The probability that the chosen relay ends up being bad 1ms from now (which is when they are scheduled to transmit) is the error event that we care about. Our neural network based predictor chooses a relay that ends up being bad robustly (in the presence of measurement noise) at the rate 10^{-4} whereas the static predictor would pick a bad relay (in the absence of measurement noise) at the rate of 10^{-2}. This suggests that for large enough networks, the total number of relays message message stream can be as low as 3 relays.

In the rest of the summary, we describe in more detail the modeling philosophy as well as results in the rest of the thesis.

0.1 Designing Wireless Communication Systems for Low-latency High-reliability Interactive Applications

Wireless channels are unreliable due to the physics of multipath propagation which causes fading. To address the unreliability, we consider diversity schemes where we send a message multiple times via different means where success occurs if at least one trial succeeds then we have overall success. The main diversity techniques are *time, frequency and space*. Time-diversity techniques essentially involve transmitting messages multiple times and if at least one of these messages encounter a good channel. To adopt these techniques for ultra-reliable low-latency communications, channels need to change from 'bad' to 'good' reliably in a short period of time. However, there is nothing in the physics of channels that forces this to occur. Hence, time-diversity-based techniques alone cannot support these applications but they turn out to be extremely important in guarding against errors (such as interference from a jammer) that are not caused due to the multipath fade being bad alone. Similarly frequency-diversity-based schemes are alone not sufficient and additionally they do not scale well with network sizes (though they are good for small networks if the multipath channel is very reliably frequency selective). However, we find that spatial-diversity-based techniques – *sending a message using multiple spatially separated antennas* such that the channels between each pair of transmit-receive antennas are uncorrelated are viable candidates for enabling such applications. As there are multiple nodes in the network, they provide several potential paths for messages to reach their destination and we use this to *harvest spatial diversity*. However, as mentioned earlier, there are other causes of transmissions failures like interference or mis-synchronized packets. These events cannot be fought by spatial diversity. They fundamentally require repetitions (or coded repetitions) in time or frequency (the effects are similar). The amount of repetition needed depends primarily on how often these error events could occur.

When a node wants to transmit a message, it transmits the packet multiple times, denoted by k_1 (the number high enough to guard against interference-like events to meet the target reliability) and other nodes in the network are listening. They can decode the packet if the channel between the transmitter and themselves is good *and* an interference like event did not occur during at least one transmission. Nodes that hear the message can *simultaneously broadcast* the message using distributed space-time-codes (DSTC) such that signals sent by multiple antennas do not destructively interfere at the receiver. They again do so multiple times, denoted by k_2 – a) to guard against interference-like events and b) to protect against mis-synchronized-packet-transmission-like errors that compound with the number of simultaneous transmitters. We can guard against these unmodeled errors through a combination of time repetitions and by frequency hopping. Although we do not do a detailed treatement of frequency hopping in this thesis, we acknowledge that this could potentially be a crucial part of URLLC design. These ideas have been explored in the cooperative communication literature in the high-SNR (large power) and large message size regime. The

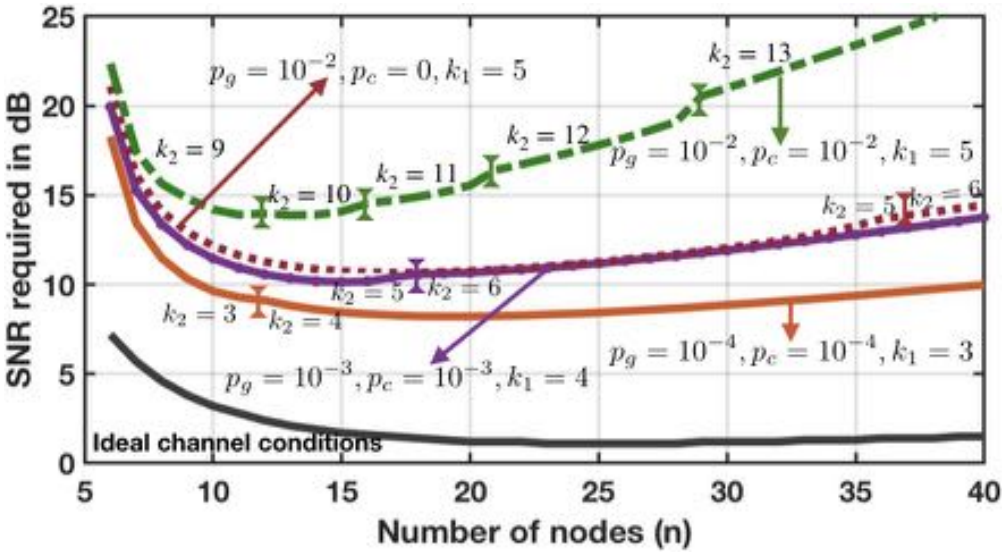


Figure 0.2: Performance of Occupy CoW – minimum SNR demanded to meet the target reliability of 10^{-9} and latency of 2ms when receiver-center, transmitter-centric and fading uncertainty errors occur. As the probability of unmodeled errors increases, the time and frequency margins needed to combat these errors increases causing the demanded spectral efficiency to much higher. It is this increase in spectral efficiency demand that pushes the SNR demanded higher.

high-performance applications that we want to enable are outside this regime. In this thesis, we explore the translation of these techniques in the medium-power, short-message regime for highly interactive applications.

To give a flavor of the key results in this thesis, consider Fig. 0.2. This figure captures the minimum SNR demanded by the system to meet the target reliability of 10^{-9} and latency of 2ms for varying number of nodes in the network. We use minimum demanded SNR as a metric to compare various curves as the smaller SNR demanded means easier implementation, lower energy consumption, longer range, less interference, etc. What we see in Fig. 0.2 is the minimum SNR demanded under ideal channel conditions (or nominal channel model where fading is the only cause of errors) and minimum SNR demanded when the likelihood of various *unmodeled* error causing events varies. In the case of ideal channel conditions, the likelihood of unmodeled error causing events is 0. Fig. 0.2 captured the effects of the following events: a) deep fade causing links to be bad captured by the nominal channel model, b) bounded uncertainty in fading model $p_{off} = 0.01$ – say due to incorrect modeling of the environment, c) global per-slot bounded badness such as interference, error-correcting-code failures, or receiver shadow transitions that do not cumulate with the number of transmitters p_g, and d) bounded per-transmitter badness due to say mis-synchronized packets, channel estimation errors, or transmitters transitioning into shadows p_c which cumulates with the number of transmitters. The exact formula used to make these curves can be found in

Appendix C.

We immediately notice: as p_g and p_c values increases, the number of retransmissions k_1 and k_2 required as well as the minimum SNR demanded increases. The increased retransmissions induce a need for higher raw spectral efficiency which drives up the SNR required. In fact, if we consider $p_g = p_c = 10^{-2}$ essentially uncontrollable unmodeled events occurring 1% of the time and budget an extra 3dB for finite-blocklength codes, we see that we need to roughly operate in the regime of 15dB to 20dB nominal SNR to be robust to most realistic error events, whereas under ideal channel conditions, we only needed to be around 3dB. Most of this is due to the increased bitrate (by at least 10x) needed to support the repeated transmission of the small packets. This kind of robustness analysis which teases apart different effects and helps engineers design the protocol appropriately is a big contribution of this thesis. More discussion about this can be found in Chapter 5.

We need to ask another question: can we protect ourselves against all error events? In other words, are there some events that we just cannot tolerate? Our goal is to make a wireless system ultrareliable to the impairments for which there is hope of being robust to. If a node were to turn into a persistent jammer, we cannot protect against that, the same as not being robust to placing one of the nodes in a Faraday cage, or the entire power grid shutting down due to a cyber attack. To put it more precisely, we have three avenues to build in robustness – time, frequency and space. If there are error events that are correlated in *time and frequency* then the hope is to move the system (or at least the data among the nodes) to a different location where hopefully the error events can be uncorrelated. On the other hands, if error events are correlated on all dimensions, then there is no hope. Those kinds of impairments are both unmodeled and irrelevant for wireless ultrareliability.

The overarching modeling philosophy of this thesis is the following. There is a large system design problem to be addressed; in this thesis it's the wireless communication protocol framework. We start by working with a nominal model by abstracting away many nuances. Here, we consider a simple model of wireless channels and focus on combating the main error event – fading. In order to arrive at the nominal model, we made assumptions. However, the system has to work ultra-reliably. So our next step is to push each assumption by constructing an 'uncertainty ball' around that assumption by capturing somehow its worst case scenario. If the protocol framework is not too sensitive to the worst case scenario, then we do not need to model this assumption more finely. For instance, we find that the performance of the protocol framework is not heavily dependent on the exact fading distributions. Therefore, we are confident of the robustness of the framework to different fading distribution (Sec. 5.1). However, if the framework's performance is adversely affected by the worst case scenario, then it calls for further examination and refinement of the model. For instance, if we assumed that channel correlation manifests as killing of diversity, then we saw the performance degraded very heavily (Sec. 5.6.1). To address that, we delved into channel dynamics and spatial correlation and refined our sensitivity (Chapter 4)

The first two chapters motivate the problem setting of this thesis (Chapter 1.6) and review the literature (Chapter 2). Chapter 3 focusses on the feasibility of using cooperative communication for high-performance applications by abstracting away most of the nuanced

effects and focussing only on fading. We also generalize our protocol for generic information topologies, which model real world systems like distributed control systems and information flows like multicast. We optimize the protocol by leveraging bi-directional traffic using network coding. We find opportunities to either reduce the power requirement by a couple of dB while keeping the end-to-end latency constant, or reduce the time needed to deliver packets by a quarter while keeping the power constant.

0.2 Wireless Channel Dynamics and Relay Selection

Since we are interested in both low-latency and high-reliability, it is imperative to have a better understanding of how channels behave on short timescales. Not just their average behavior but also the 'tails' of their behavior since 'rare' events that occur as often as 10^{-4} are too frequent when we need reliabilities of 10^{-9}. Therefore, in this thesis, we take an in-depth look at wireless channel dynamics to study the events that may occur in timescales ranging from tens of microseconds (corresponding to the length of a single short packet) to a few milliseconds (corresponding to the cycle time). A knowledge of these events helps us to focus on issues that might otherwise be overlooked if we only considered a traditional quasi-static-channel models which were sufficient in the context of cellular and WiFi like systems where events occurring as frequently as 10^{-4} are considered 'very rare'.

Chapters 4 and 5 take a critical look at the main characteristics of wireless channels that impact the design and performance of different communication schemes. We do this by looking at both the nominal model of fading processes as well as identifying key dimensions of uncertainty to capture the impact of unmodeled effects. We find that Rayleigh-fading processes are not bandlimited and this has ramifications on the predictability of channels. We specifically find that channels vary significantly even within the traditionally defined "coherence time" which poses a tough challenge for predictability. We give a brief flavor of the results to come in Fig. 0.3 which studies the variation of channel energy within a packet for a static transmitter transmitting at center frequency of 3Ghz and a mobile receiver moving at speed 10m/s. The traditional notion of coherence time for this setup is 2.5ms. However, Fig. 0.3 shows that even good channels (solid curves which correspond to those channels that originally started out as good) do not reliably remain static for 1ms. This suggests that in the context of ultra-reliable low-latency communication, having small packets (on the order of 10μs) is important for better stability. Although small packet sizes seem to be an artifact of short message sizes, they may actually be necessary for ultra-reliability. This also has significant implications for channel prediction and relay selection as described below.

Chapter 5 identifies the key parameters needed to bound unmodeled uncertainty and analyzes the impact of various unmodeled error events on the performance of cooperative communication based protocols. We find that many unmodeled errors such as not knowing the channel distribution accurately or having channels be spatially correlated can be protected against easily. However, unmodeled external errors like stray interference may be something we need to think about more carefully. This gives rise to the need for using a com-

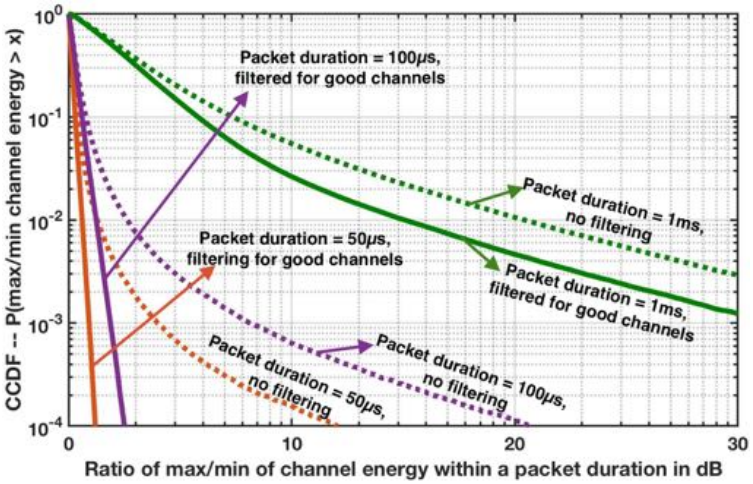


Figure 0.3: CCDF of the ratio of max channel energy by min channel energy in dB within a packet duration for various packet durations. The receiver is traveling at a speed of 10m/s and the center frequency is 3GHz. The dotted curves correspond to all channels and the solid curves correspond to those channels that are good at the beginning of the packet. For short packet sizes of 50μs, there is no discernible change in the channel energy when conditioned on the initial channel being good but as the packet duration increases, we see bigger variations become possible. The traditional "coherence time" in this setup is 2.5 ms.

bination of time/frequency repetition techniques in addition to spatial diversity techniques to build robustness and we modify our protocol framework to have repetitions in time and frequency.

One of the key insights from Chapter. 5 is that errors such as synchronization-mismatch-induced-decoding errors compound with the number of simultaneous transmitters. Moreover, having multiple nodes transmitting the same message not only wastes energy but also increases interference levels to neighboring networks. These reasons motivate us to intelligently select a small set of relays to act as helpers. However, this selection must be ultra-reliable – the probability of not having at least one of these helpers be good in the future should be lower than 10^{-9}. To this end, in Chapter 6 we develop intelligent relay selection schemes that can predict channel quality and pick a small number of good relays. We leverage the channel dynamics knowledge to theoretically study relay selection performance under a Gaussian process model. We find that the most important parameters for prediction accuracy are: sampling frequency, future horizon, and the number of potential relays to choose from. In a simulated setup we consider different relay quality prediction schemes including the state-of-the-art static predictor that does not consider channel dynamics, simple polynomial based predictors as well as linear predictors based on the Gaussian process model, and neural-network-based predictors that leverage the non-linearity of channel quality progression. The predictors predict the channel quality and we employ one of the standard relay-selection techniques from the literature to choose a relay from the set of relays available. The probability

that the chosen relay ends up being bad 1ms from now (which is when they are scheduled to transmit) is the error event that we care about. Our neural network based predictor chooses a relay that ends up being bad robustly (in the presence of measurement noise) at the rate 10^{-4} whereas the static predictor would pick a bad relay (in the absence of measurement noise) at the rate of 10^{-2}. This essentially brings the total number of relays needed for the cooperative communication framework to 3.

Chapter 7 presents preliminary experimental results that build a better understanding of real-world events and potential future work that can be built on this thesis. Through this thesis, we fundamentally address the question of designing low-latency high-reliability wireless communication system by using ideas from communication theory and machine learning while always having an eye on the practicality of the systems and unmodeled error events.

Chapter 1

Introduction

1.1 Ultra-reliable Low-latency Communication

Imagine a world where a large number of globally distributed embedded computing devices communicate with each other and interact with the physical world [1]. What does that look like? It can be a fully connected smart home that knows when you've woken up and delivers your breakfast whenever and wherever you want. It can also be a congestion-free driving experience with no accidents and cars platooning seamlessly [2]. It can be a world where wearable technology is ubiquitous and in addition to distributing content, control signals are also transferred wirelessly. This is the vision of the Internet of Things (IoT) that technologies like 5G are expected to enable (Fig. 1.1). On the one end of the spectrum of applications is precision agriculture that requires low-power, low-cost devices and long-range, sparse communication [3]. On the other end of the spectrum are high-performance interactive and immersive applications that require low-latency, high-reliability and high-frequency communication [4, 5]. This thesis focuses on the latter domain – applications requiring ultra-reliable low-latency communication (URLLC).

High-speed ultra-reliable wireless communication networks are critical for developing near-real-time machine-to-machine networks and applications such as industrial automation, immersive virtual reality (VR) and the "tactile internet." This interaction includes not just sensing but also simultaneous actuation of numerous connected devices. For truly immersive applications, the latency requirements on the whole control loop are in the teens 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 [6], with a round-trip delay of approximately 1 ms [2] and reliability of 10^{-8} [7], as achieved with wired connections. These correspond roughly to a third of a second of outage per year. A current domain that demands ultra-reliable low-latency application is industrial automation where the requirements are satisfied by wired communication systems like fieldbuses. However, the future of industry, especially manufacturing envisions a flexible and agile factory floor which is easily reconfigurable. This would demand

Figure 1.1: A smattering of applications and domains that the Internet of Things will enable including smart homes, connected wearable devices, precision agriculture, smart grid, vehicle platooning, drone swarms and precision medicine.

communication requirements to be met wirelessly. Therefore, in this thesis, we consider industrial automation to be a representative example for the applications enabled by URLLC and therefore we target the specifications demanded by industrial automation.

1.2 A Shift In Wireless System Design

The main focus of wireless systems such as cellular and WiFi has been on increasing data-rate. Specifically, the growth of data-rate from 3G to 4G is quite dramatic – from 200kbps [8] to at least 100Mbps [9]. Although the techniques used have undergone a tremendous transformation, the goal has remained more or less the same – increase data-rate and increase coverage area. This brought about a list of problems that had to be addressed including channel models, mobility and hand-off techniques, signaling strategies, frequency reuse, MIMO techniques, etc [10, 11]. Similarly, usage of WiFi has exploded. But unlike cellular systems, WiFi systems have slightly different constraints. Their goal still is to increase data-rate but the coverage area of WiFi is very much the size of a house rather than multiple city blocks like in the case of cellular systems. Additionally, as WiFi operates on ISM bands, co-existence with other WiFi systems as well as other technologies operating on the ISM band is a key

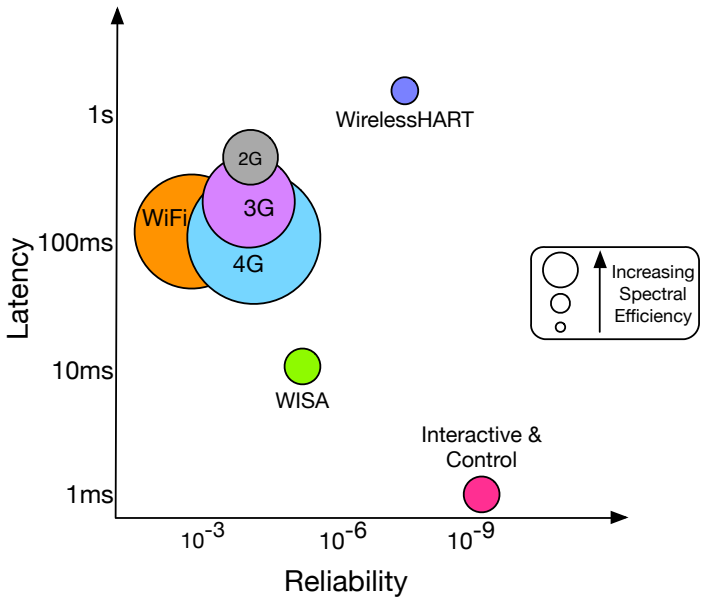


Figure 1.2: Comparison of some current wireless technologies on latency and reliability axis. Interactive and immersive applications that we target is depicted by the pink circle.

issue. Therefore, higher layer protocols in the OSI stack such as carrier-sense multiple access with collision avoidance [12] which is a data-link layer protocol are implemented to increase the overall usage of the available wireless medium.

As the focus primarily has been on increasing data-rate, increasing coverage area and overall power and spectral efficiency [13, 14], latency and reliability were not prioritized. Fig. 1.2 compares some of the existing wireless communication technologies on the reliability and latency axis [1]. Although overall reliability has been a requirement of networking in general, the techniques used to achieve high reliability has primarily been in the upper layers of the OSI stack such as TCP/IP. Due to this difference in focus so far, techniques used by existing standards are fundamentally ill-suited for low-latency and high-reliability [6]. We study the largely unexplored space of low-latency and high-reliability suited for interactive and critical control applications. This thesis focuses on the lower-layer protocols (PHY and MAC) without going into details of exact signalling strategies and such. We take a broad look at system-design and address some of the pain-points one may run into while designing a system demanding low-latency and high-reliability wireless communications.

1.3 Thesis Contributions

This thesis addresses the following question: what are the key elements for designing ultra-reliable low-latency wireless communication systems for highly interactive applications? In

this thesis we take a mixed approach – high level view and then zoom-in on the main pain points.

We first take a high level view on this problem while abstracting away several details to zero-in on the key features. We find the main ingredients that a communication scheme should have to enable URLLC: using cooperative communication techniques for increasing reliability along with simultaneous relaying and network coding for reducing latency. We design communication protocol schemes that incorporate these elements and analyze the performance of these schemes under some assumptions (in the same theme as the abstractions) and compare it with other techniques that do not incorporate these elements using the communication theoretic framework.

Through these abstractions, we arrive at a framework that would appropriately address the key pain point – *building in diversity in the system quickly.* As we target ultra-reliability, it is essential to build an in-depth knowledge of the events that could potentially lead to our system failing. To this end, this thesis takes delves into unraveling the abstractions that could cause these potential failure events. We look at the effects of these events as well as propose ways to counter them. One of the main results is to smartly select a small set of relays by leveraging the knowledge of channel dynamics to ultimately build a practical system.

The overarching modeling philosophy of this thesis is the following. There is a large system design problem to be addressed; in this thesis its the wireless communication protocol framework. We start by working with a nominal model by abstracting away many nuances. Here, we consider a simple model of wireless channels and focus on combating the main error event – fading. In order to arrive at the nominal model, we made assumptions. However, the system has to work ultra-reliably. So our next step is to push each assumption by constructing an 'uncertainty ball' around that assumption by capturing somehow its worst case scenario. If the protocol framework is not too sensitive to the worst case scenario, then we do not need to model this assumption more finely. For instance, we find that the performance of the protocol framework is not heavily dependent on the exact fading distribution. Therefore, we are confident of the robustness of the framework to different fading distribution (Sec. 5.1). However, if the framework's performance is adversely affected by the worst case scenario, then it calls for further examination and refinement of the model. For instance, if we assumed that channel correlation manifests as killing of diversity, then we saw the performance degraded very heavily (Sec. 5.6.1). To address that, we delved into channel dynamics and spatial correlation and refined our sensitivity (Chapter 4) .
The main results of the thesis are the following:

- We design a wireless communication framework for URLLC that is intended as a drop-in replacement for wired protocols in place. The framework combines a variety of techniques to combat different kinds of challenges. We provide in-depth analysis of its performance using the delay-limited-capacity framework.
  - To fight fading, we primarily use spatial diversity techniques.

  - To fight local error-causing events such as mis-synchronized packets as well as global error-causing events like interference from a jammer, we use repetitions in time and frequency.
  - We find that further optimizing the protocol by incorporating network coding is advantageous but any other minute optimization such as rate adaptation does not provide benefits to offset the cost of information dissemination as well as designing advanced radios to perform variable rate decoding.

- We take an in-depth look at channel dynamics to study the fading events that may occur in timescales ranging from tens of microseconds (corresponding to the length of a single short packet) to a few milliseconds (corresponding to the cycle time).
  - We formulate the fading process as a Gaussian process using the Jakes's model. Theoretically, the covariance function is a Bessel function of the first kind and our simulations co-incide with the Bessel function, even for a small number of scatterers.
  - Although channels in Jakes's model are spatially correlated in principle, as long as antennas are multiple wavelengths apart, the correlation is slight, and can be offset by a fraction of a dB increase in transmit power.
  - We find that for short packets of duration under 100μs (motion under 0.01λ), channels that are good enough stay quite static. There is no large variation in channel energy within this time if the channels started out to be good. On the other hand, if channels were deeply faded to begin with, even minute changes in energy would manifest as large relative changes but this really has no impact on the performance as those were bad channels anyways.
  - We study the channel dynamics on the order of hundreds of microseconds to a few milliseconds which corresponds to the cycle time as well as relaying events. We find that Rayleigh fading processes are not bandlimited. This has significant implications in channel quality prediction and relay selection techniques.

- As the focus is on ultra-reliable communication, we critically question how the framework would perform if the nominal channel models break down. We model the uncertainties with bounds on the following uncertainties:
  - the maximum probability of an unmodeled link error that is independent across transmitter/receiver pairs (i.e., from a different CDF for multipath fading).
  - the maximum probability of an unmodeled decoding error that is independent across time-slots (similarly we can think of errors to be correlated across time but independent across different frequencies).
  - the maximum probability of an unmodeled decoding error that compounds with the number of simultaneous transmitters, but is independent across time-slots.

We use the above uncertainties combined with our nominal channel model to provide a robust model for wireless uncertainty.

- We leverage the channel dynamics knowledge to theoretically study relay selection performance under a Gaussian process model. We find that the most important parameters for prediction accuracy are: sampling frequency, future horizon, and number of potential relays to choose from. In a simulated setup we consider different relay quality prediction schemes including the state-of-the-art static predictor that does not consider channel dynamics, simple polynomial based predictor as well as linear predictor based on the theoretical Gaussian process model, and a neural-network-based predictor that leverages the non-linearity of channel quality progression. The predictors predict the channel quality and we employ one of the standard relay-selection techniques from literature to choose a relay from the set of relays available. The probability that the chosen relay ends up being bad 1ms from now (which is when they are scheduled to transmit) is the error event that we care about. Our neural network based predictor chooses a relay that ends up being bad robustly (in the presence of measurement noise) at the rate 10^{-4} whereas the static predictor would pick a bad relay (in the absence of measurement noise) at the rate of 10^{-2}. This suggests that for large enough networks, the total number of relays message message stream can be as low as 3 relays.

1.4 Important questions in URLLC

A distinctive feature of URLLC is the size of the packets – they are much smaller than in WiFi or LTE. In fact, in 3GPP discussions [15], the payloads generated by nodes are assumed to be less than 256bytes as compared to 2000+bytes in WiFi. Therefore, a short packet communication paradigm kicks in. Although 3GPP has settled on using Orthogonal Frequency Division Multiplexing (OFDM) variants signaling for mobile broadband, signaling for short packets is still an open question.

1.4.1 PHY layer strategies for URLLC

Several works have studied the suitability of various signaling strategies for low-latency applications. Specifically alternatives for OFDM have been considered in order to relax synchronization requirements and reduce out-of-band (OOB) transmissions. Some of the alternatives being considered are Filter Bank Multi-carrier (FBMC) [16], Universal Filtered Multi-carrier (UFMC) [17] and Generalized Frequency Division Multiplexing (GFDM) [18]. A comparison of these techniques in the context of 5G and URLLC found that generally FBMC allows for relaxed guard-band spacing and is more suitable for achieving high spectral efficiency but not suitable for URLLC due to extended time-domain signal whereas UFMC offered better reliability [19]. Various PHY and MAC layer solutions for mMTC (massive

MTC) and uMTC (ultra-reliable MTC) are discussed in [20] where they conclude that higher-layer considerations play an important role to ensure lean signaling by enabling longer sleep cycles etc.

Levanen et al. [21] concentrate on the proposed 5GETLA radio interface and show that latencies below 1ms for payloads of size 50kb are achievable provided a bandwidth of at least 100MHz is available. Though the targeted latency is of the same order as required by industrial control, they do not consider reliability guarantees or retransmissions. A discussion of the feasibility, requirements, and design challenges of an OFDM based 5G radio interface suitable for mission-critical machine type communication (MTC) concluded that multiple receive antennas are critical for interference mitigation [22]. In similar spirit, coverage and capacity aspects concerning both noise-limited and interference-limited operations for MTC were considered in [23]. Various PHY and MAC layer solutions for mMTC (massive MTC) and uMTC (ultra-reliable MTC) are discussed in [20] where they conclude that higher-layer considerations ensure lean signaling by enabling longer sleep cycles and other techniques. Without doubt, synchronization is also going to be extremely crucial. Identifying both the requirements needed for different applications as well as constructing algorithms to achieve them are important. Some of the recent works such as [24, 25, 26] have identified the synchronization requirements as well as suggested techniques to achieve them. For example, one can identify the source of the synchronization signal as either the master clock or a node synchronized to the master clock sending out secondary synchronization signals and give more weight to the more trusted source of synchronization signal.

1.4.2 Metadata optimization and Coexistence

Efficient communication of short packets in the communication-theoretic context was discussed in [27, 28, 21] where the key insight is that when packets are short, the resources needed for metadata (like preamble, header, etc.) transmission should be considered as payload is now comparable in size to traditional metadata. Coexistence of URLLC with other traffic such as mobile broadband is crucial to enable heterogeneous usage of channels [29] where they find that having the mobile broadband scheduler have knowledge of the URLLC application's state would enable good co-existence. Network slicing to support heterogeneous traffic in the context of 5G namely, eMBB, URLLC and mMTC has been studied in [30, 31] where they find that non-orthogonal slicing guided by reliability diversity (i.e., URLLC first followed by the rest) can provide significant gains in terms of performance tradeoffs while compared to orthogonal slicing. Other studies have focussed on hand-off and mobility management of these kinds of traffic for $5G$ [32, 33].

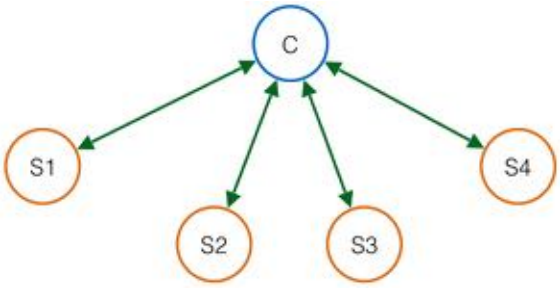
1.4.3 Security and Positioning

Other major concerns that 3GPP as well as several industry groups have is with security and privacy especially given dense deployment as well as devices mainly being embedded systems with limited GUI [15, 34, 35, 36]. Vehicle-to-everything communication is a near-future

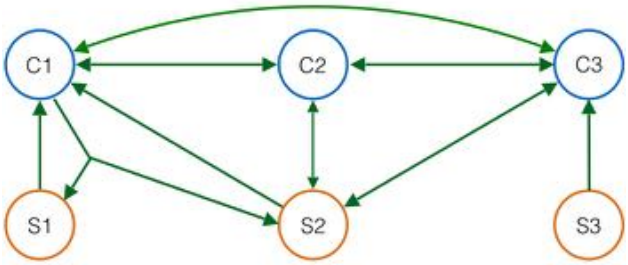
application that requires URLLC [37, 15, 38]. In addition to answering questions about designing ultra-reliable, low-latency wireless communication, they demand higher degree of accuracy in positioning as well since many algorithms like platooning and self-driving require accurate knowledge of vehicle's own as well as neighbor's positions. Thereby we see that applications demanding URLLC may have high demands on other systems as well and URLLC might be crucial to support multiple systems.

1.5 Problem Setup

The thesis considers a popular current application that requires ultra-reliable, low-latency communication as a motivation as well as proxy for the future applications that URLLC may enable. Throughout this thesis, the setup of the network is abstracted as follows. We target a local wireless domain where nominally all nodes are in range, but fading might cause a pair of nodes to be unable to hear each other. The traffic patterns (what we deem the "information topology") of interest consist of steady streams of messages, each originating at possibly different nodes within the network, and each stream subscribed to by some (possibly different) subset of nodes within the network. Within a short period of time, deemed a "cycle time," every stream needs to deliver one packet reliably to its subscribers. The information topology can be arbitrary – something naturally centralized like a star topology as shown in Fig. 3.1a (e.g. with a central controller talking to many sensor/actuators collecting streams of measurements and sending streams of commands) or something more generic as in Fig. 3.1b.



(a) Star message flow topology



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

We explore the performance of different schemes (including the ones we propose) with parameters in the neighborhood of a practical application, the industrial printer case described in Weiner et al. [6]. This application is modeled as a central system such as the one shown in Fig. 3.1a. In this particular scenario, the SERCOS III protocol [39] supports the printer's required cycle time of 2 ms with reliability of 10^{-8}. Consequently, we target a 10^{-9} probability of failure. The printer has 30 moving printing heads that move at speeds up to 3 m/s over distances of up to 10 m. Every 2 ms cycle, each head's actuator receives 20 bytes

from the controller and each head's sensor transmits 20 bytes to the controller. The amount of information transmitted by the controller in a single cycle is $20 \times 30 = 600$ bytes. The total amount of sensor information transmitted by the heads to the controller in a single cycle is also $20 \times 30 = 600$ bytes. Therefore a total of 1200 bytes or 9600 bits of information is sent during a cycle of 2 ms which corresponds to a desired goodput of $\frac{9600}{(2\times10^{-3})}$ bit/sec $= 4.8$ Mbit/sec. If we assume access to a single dedicated 20 MHz wireless channel, this 4.8 Mbit/sec corresponds to an overall net spectral efficiency of $\frac{(4.8\times10^{6})}{(20\times10^{6})} = 0.24$ bits/sec/Hz. Throughout this thesis, we explore the performance of different schemes in ranges similar to the ones found in the printer application.

1.6 Dissertation outline and previously published work

The remainder of this dissertation proceeds as follows: In Chapter 2 we provide background information on related topics including the evolution of communication for industrial control, recent efforts in 5G standards, cooperative communication and channel modeling. In Chapter 3 we introduce the two main cooperative communication based schemes of this thesis: "Occupy CoW" and "XOR-CoW". We analyze their performance and explore different optimizations. Next in Chapter 4 we take a critical look at the main characteristics of wireless channels that impact the design and performance of these communication schemes. In Chapter 5 we identify the key parameters needed to bound unmodeled uncertainty and analyze the impact of various unmodeled error events on the performance of cooperative communication based protocols. In Chapter 6 we develop intelligent relay selection schemes that can predict channel quality and pick a small number of good relays. In Chapter 7 we present the experimental results that build a better understanding of real-world events as well as validate our models.

In this thesis, the material in Chapter 3 is adapted from [40, 41, 42], the material in Chapter 4 is adapted from [43, 44], the material in Chapter 5 is adapted from [45, 46] and the material in Chapter 6 is adapted from [43].

This thesis may undergo some minor changes and the reader is encouraged to look at this URL [47] for the most updated version.

Chapter 2

Related Work and Background

This thesis brings together ideas from a variety of areas including cooperative communication, network coding, finite blocklength coding, channel modeling and machine learning. Specifically, we use cooperative-communication-based techniques to combat fading; network coding to optimize for bi-directional traffic; and leverage channel dynamics knowledge to build intelligent channel quality prediction schemes to aid in relay selection. In this chapter, we briefly review some of the literature in these areas and we first begin with looking at some of the recent developments in the 5G and Ultra-reliable low-latency communication (URLLC) domain.

2.1 Recent developments in proposed 5G protocols and URLLC

Latency and reliability have risen in importance as 5G wireless is discussed, taking their place alongside a focus on increasing capacity and energy efficiency while also using mmWave frequencies [25, 4, 48]. Unlike previous 3GPP systems that attempted to provide a 'one size fits all system', the 5G system is expected to be able to provide optimized support for a variety of different services. One important driver for very short round-trip time (RTT) latencies, of the order of 1ms, is to support tactile feedback to wireless users, enabling immersive VR applications [49]. There has been an explosion of work in the area of URLLC – including identifying the domains that may be targeted and system requirements specific for those domains [5, 2].

The latest version of 5G technical specifications released by 3GPP has already scoped out the modulation [50], channel coding [51] and physical layer procedures for control [52] and data [53] in the context of mobile broadband. In the next phase of 5G standards body discussions (starting with Release 16 which would begin during the later half of 2018), there will be significant focus on enabling URLLC through 5G [15]. They envision a combination of heterogeneous networks that includes both the traditional user to base-station connection and user-to-user connection bypassing the need for a base-station to interface between them

specifically in the context of URLLC. The reliabilities and latencies targeted by 5G standards for URLLC are in the range of 10^{-5} and 1 to 10ms respectively. In addition to studies conducted by 3GPP, associations such as 5GAA (5G Automotive Association) has brought together automotive technology, and telecommunications industries (ICT), to develop end-to-end solutions for future mobility and transportation services. Studies by 5GAA including [54, 55] again stress the need for URLLC for vehicle-to-everything communication and propose specifications for different varieties of signals (such as red-light jumping to detect a potential crash). There are several questions that need to be answered before a 5G standard for URLLC can materialize which we have summarized briefly in Chapter. 1.6.

Several recent works have been considering strategies for enabling wireless URLLC. Authors in [56, 57] consider using optimized coded interface diversity to ensure reliability. They find that optimizing based on the characteristics of different interfaces (such as correlated failures) can outperform choosing strategies universally. In the similar spirit, [58] considers using packet duplication in the higher layers (as it is already being supported by current 5G standards) to essentially capture interface diversity.

Schemes that rely on the channel state information to maximize the data rate or choose 'leader' nodes have been considered in the context of URLLC. In [59], the authors consider a centralized control system where the controller performs pilot-assisted channel estimation to adapt the transmission rate to each device based on the quality of its channel. In [60], the authors consider a leader selection scheme based on channel state information. Both of these studies assume that the channels remain static for the duration of a cycle and have promising results. Prediction in the data-plane layer for improving bandwidth reservation has been studied in [61] where the authors find that their prediction based method can save up to 66.7% of the bandwidth compared with the method that is not aware of burstiness.

2.2 Industrial automation

Communication in industrial control systems have traditionally been wired. Point-to-point wired systems were replaced by the *fieldbus* systems (industrial computer networks) to mitigate the large number of physical points of failure introduced by connectors and wire harnesses, resulting in a highly unreliable system. Some of the popular fieldbus systems serving these applications are SERCOS, PROFIBUS and WorldFIP [62, 63, 64]. The main objective of the fieldbus system is to provide reliable real-time communication. But as industries are aiming to connect control systems globally, fieldbuses are becoming increasingly inadequate to meet the flexibility demands [65]. This has prompted a paradigm shift towards installing wireless communication systems in industrial control environments as it reduces bulk and installation costs in addition to easing of deployment in remote areas. Due to widespread use of fieldbus systems, several wireless extensions have been looked in [66, 67, 68]. Although these are technically feasible, deploying them is not straightforward as random access techniques like CSMA and Aloha are employed. The amount of collisions in these techniques could result in unbounded transmission delays which are intolerable in tight-control loops

[69].

The first process application that truly employed wireless communication was monitoring. There have been many studies on design and implementation of Wireless Sensor Networks (WSN) [70, 71, 72] and they have been successfully deployed. As the main objective of a WSN is to conserve energy, sensor nodes spend most of their time in sleep state. This hinders the straightforward adoption of wireless sensor networks in tight-control loops [73].

Some wireless technologies have been specifically tailored for use in industrial control *e.g.,* Wireless Interface for Sensor and Actuators (WISA)[74], ZigBee PRO [75], WirelessHART [76], ISA 100 [77] to name a few. WISA is a great attempt at meeting stringent real-time requirements but fails to achieve interoperability and multipath routing. Additionally the reliability of WISA (on the order of 10^{-4}) does not work for control[78]. ZigBee PRO does not employ good diversity techniques and hence does not provide good reliability [79]. Both ISA 100 [77] and WirelessHART [76] provide secure and reliable communication, but cannot meet the latency bounds as *each* packet is 10ms long. The median latency for a successful delivery is in the order of 100ms and the protocols are heavily optimized for power consumption [80]. We would need a *reliable and real-time* protocol if we want to have a drop-in replacement for existing fieldbus like SERCOS III, which provides a reliability of 10^{-8}.

The IWLAN standard, which is based on a combination of PROFINET with 802.11n WLAN, attempts to resolve this issue by adding proprietary scheduled polling called iPCF (industrial Point Coordination Function) [81, 82, 83, 84]. To deal with a deep fade, IWLAN has to rely on handing over the faded node to a redundant access point, and this handover is sped up by using proprietary industrial-automation oriented enhancements to the 802.11n protocol. Even with these enhancements, handover can only occur at the time-scale of tens of milliseconds [81].

The focus of majority of mechanisms for addressing the QoS requirements in wireless monitoring and control industry is mainly on higher layer techniques, like MAC layer contention, channel hopping and multipath routing [65] and less on PHY layer. Most of these do not exploit diversity techniques like spatial and cooperative diversity, multipath routing etc. This leads us to the next part of related works where we explore the cooperative communications and multi-user diversity techniques in wireless communications literature.

2.3 Cooperative communication and multi-user diversity

The key to getting reliability in wireless communication is to harness diversity [10]. Highly-reliable WSNs use techniques like channel hopping and contention-based medium access control (MAC) to harvest time and frequency diversity, and multi-path routing as an indirect way to harvest spatial diversity [65]. Unfortunately, low-latency applications like ours cannot use time diversity since the cycle times of single-digit milliseconds could very well be shorter than channel coherence times of tens of milliseconds. Techniques like Forward

Error Correction (FEC) and Automatic Repeat Request (ARQ) also do not provide much advantage in the face of fading [85]. Later in this thesis, we demonstrate that frequency-diversity based techniques also fall short, especially when the required throughput pushes us to increase spectral efficiency. Even beyond the issue of poor performance, there is the issue of availability — exploiting frequency diversity requires us to crucially depend on nature to provide enough multipaths with a large enough delay spread to actually create frequency diversity [10]. Consequently, our protocol leverages spatial diversity instead to combat fading.

The size of the networks targeted in this thesis is moderate (10 - 100 nodes active at once). Therefore, there is an abundance of antennas in the system and we can harvest some resulting diversity. Multi-antenna diversity is mainly of two types: a) sender diversity where multiple antennas transmit the same message through independent channels and b) receiver diversity where multiple receive antennas harvest copies of the same signal received via independent channels. Researchers have studied these techniques in great detail; so our treatment here of the literature is limited. Cooperation among distributed antennas can provide full sender-diversity without the need for physical arrays [86]. Even with a noisy inter-user channel, multi-user cooperation increases capacity and leads to achievable rates that are robust to channel variations [87]. The prior works in cooperative communication tend to focus on the asymptotic regimes of high SNR. By contrast, we are interested in low - moderate SNR regimes (around 10 dB) since we envision some of these applications to have battery operated devices.

Multi-antenna techniques have been widely implemented in commercial wireless protocols like IEEE 802.11. Sender-diversity harvesting techniques using relaying coupled with a time division multiple access (TDMA) based scheme have been explored for industrial control [85, 88]. Unfortunately, as we discuss later and can see in Fig. 3.5, strict TDMA for relays can scale poorly with network size since relaying for one message consumes many slots to get enough diversity to attain high reliability. To scale better with network size, our protocol uses simultaneous transmission by many relays, using some distributed space-time codes (DSTCs) such as those in [89, 90, 91], so that each receiver can harvest a large diversity gain. While we do not discuss the specifics of space-time code implementation, recent work by Rahul et al. [92] demonstrates that it is possible to implement schemes that harvest sender diversity by using concurrent transmissions.

2.4 Control and communication co-design

This thesis's approach to enabling wireless industrial automation is to maximize the reliability of communication while simultaneously reducing latency. For completeness, we mention that a second approach towards achieving successful wireless industrial automation would be to adapt control algorithms to (the reliability and latency guarantees provided by) wireless communication and to co-design the two modules. Fundamental limits for control and estimation of systems over both noiseless rate-limited channels [93, 94] and noisy channels [95] have been established. A series of works [96, 97, 98] established the limits of control and

estimation over packet dropping networks and it was recently shown that control and communication co-design could provide unbounded performance gains in such settings [99].

The literature on adapting control to wireless communication has generally focussed on leveraging the optimization paradigms of control-theoretic synthesis. Works like [100, 101, 102] combine data rate and quantization with performance optimization and dealing with packet drops. A holistic view of network parameters including the placement of controller functionality has been studied in [103, 104]. Finally, there are even more completely integrated approaches like the wireless control network idea proposed in Pajic et al. [105] wherein the wireless network itself is modeled as the controller with the network topology providing an implementation constraint and the unreliability of the links viewed through the lens of robust control techniques [106].

This thesis focuses exclusively on improving communication. This has two motivations. First, it follows the principle of layering as it allows unmodified control laws (which might not have been developed using any particular synthesis methodology or even stated performance criteria) to operate with a new communication layer [107]. Second, it establishes a baseline upon which we can study the gains achievable through co-design, which warrants further investigation.

2.5 Network Coding

The seminal work of Ahlswede *et al.,* [108] showed that regarding information to be multicast as a "fluid" to be routed or replicated in general is not optimal and employing coding at nodes can lead to efficient use of bandwidth. This idea was further studied in [109], where a forwarding architecture for wireless mesh networks to improve throughput by introducing a coding layer in between the IP and MAC layers was proposed. They provide a practical implementation of network coding into the current network stack, addressing the common case of unicast traffic, and dynamic and potentially bursty flows. Recent results in [110] show that using randomized space-time block coding (RSTBC) in two-way relay networks improves throughput by exchanging data through a bi-directional relay network. Like most works using network coding, we aim to increase throughput which translates to lower latency. Fig. 3.13 illustrates how we use network coding combined with simultaneous retransmissions in our work. Essentially, if there is a natural viability for XORing then, only those nodes with the necessary packets help by broadcasting the XORed packet.

The wireless communication system framework proposed in this thesis combines cooperative communication and network coding techniques to achieve the desired QoS requirements by exploiting multi-user diversity and distributed space-time codes (such as those in [89, 90, 111], so that each receiver can harvest a large diversity gain) to achieve high-reliability and low latency. The key idea here is that relays simultaneously broadcast coded packets (as long as they are coding the same set of packets).

2.6 Finite-Block-Length Coding

Coding for short packets is an important question in URLLC. Recent works have been looking at the feasibility of using different coding techniques for URLLC. [112, 113, 114]. In this section, we briefly provide pointers into the relevant background underlying finite block length effects on performance of error correction codes. A useful communication-theoretic perspective is to decompose the required SNR into three parts: (1) capacity: how much does the rate fundamentally require? (2) gap-to-capacity: given the target reliability and the specific code being used, how many extra dB do we need beyond capacity? (3) fading-margin: how many dB do we need to absorb bad wireless fades?

Although it is useful to be able to think about these separately, they clearly interact with each other at the system level. For example, if overall "goodput" is what is desired and the higher layers will use ARQ to achieve high reliability, then lowering the target reliability on a link comes at the cost of more retransmissions and hence less overall rate. In [115] the authors propose that links which fail about 10% of the time (allowing more aggressive code rates) result in the best goodput. A similar finding is reported in [116] from a channel dispersion perspective. The question this thesis addresses is whether a similar story holds when we have a diversity-oriented cooperative communication protocol with a low latency requirement.

Most of the theoretical results about cooperative communication assume infinite block-length codes. However finite block-lengths matter in this context [116, 117]. In the wireless context, the impact on the diversity multiplexing tradeoff was studied in [118, 119], the effect of outdated CSI was studied in [120], the effects of queue constrains were studied in [121], and effect of coherence time on diversity was studied in [122]. There are other notable works which have focussed on interesting aspects of finite block length coding [123, 124, 125, 126, 127, 128]. The recent paper [129] looked at a very similar problem involving the effects of finite-block-length codes as well as CSI estimation-overhead in the context of low-latency applications and presented simulation results where the finding was that the average PER increases in the number of participating terminals unless the terminals also act as potential relays.

2.7 Channel Modeling

Advancement in wireless communication technologies have been made possible by studies about channel characteristics in indoor and outdoor environments and extensive modeling [130, 131]. Characteristics of wireless channels like propagation loss in different environments have been instrumental in estimating coverage area for both traditional systems like cellular communication as well as newer technologies that leverage TV White Space bands [132, 133, 11]. Recently, wireless channel characteristics are being considered for building applications that leverage backscattering to model objects in the environment, building indoor positioning applications as well as novel low-power communication technologies [134]. As advancements in hardware have enabled building mmWave radios, studying character-

istics of wireless channels in indoor and outdoor environment in the mmWave band have become crucial to develop exciting applications including tether-less AR/VR and vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication to enable autonomous vehicles [135, 136, 137, 138]. Most of these works have focused either on large scale statistics of channels (relevant for problems like capacity estimation) or dynamics in relatively larger scale for instance in the order of seconds [139]. However, in this thesis (Chapter 4), we focus on temporal and spatial channel characteristics at both small as well as large scales that capture various (potentially rare) events of interest that can ultimately cause a failure event.

2.8 Learning and Communication

Advancements in machine learning and applications of machine learning techniques in various fields such as computer vision has paved the way for significant development in various fields. Usage of machine learning models for wireless communication and networking in general has been garnering a great interest. Predicting traffic, usage and caching to avoid congestion has long been a part of networking – especially wired. With video streaming over wireless becoming increasingly popular, there is an added layer of complexity due to the nature of wireless medium, thus paving the way for proactive wireless networking, in which a wireless network can predict its users' behavior and adapt its operation [140, 141, 142, 143, 144]. Similarly, there is potential for better resource allocation through learning network topologies as well as link quality – [145, 146, 147] are just a small sample of works that have considered such strategies (including neural networks and reinforcement learning) for wired as well as wireless networks. More comprehensive surveys of using machine learning techniques for next-generation wireless networks as well as wireless sensor networks can be found in [148, 149, 150, 151, 152, 153]. Recent works have focused more on the use of machine learning in the context of IoT and specifically URLLC. A multi-armed bandit based reinforcement learning approach is proposed to achieve the optimum harmonization of feedback and feedback-less transmissions in [154]. Balancing computation and communication at fog networks on-the-fly to minimize service latency is considered in [155]. Machine learning methods to predict the outcome of the decoding process ahead of the end of the transmission are considered in [156]. There is also immense interest in using machine-learning-based-architectures for traditional radio design [157, 158].

In this thesis, machine learning techniques are applied primarily to predict channel quality. Other studies have also dealt with channel coefficient prediction. Particularly, [159] considers local polynomial method for channel coefficient prediction for efficient resource allocation to maximize capacity. However, they assume small node mobility which is different from the case considered in this thesis. Prediction of channel quality under Gaussian process models is considered in [160], however this work assumes (partial) knowledge about radio position which is not an assumption made in this thesis. Specifically, in this thesis, we focus on predicting wireless channel quality with high-reliability with minimal information (such

as maximum speed of a node) about the system and using channel state information from the near-past.

URLLC is a hard engineering problem and it requires us to carefully think about each component. As mentioned earlier, there are several works looking at specific system design aspect like short packet signaling, error correction codes for short block lengths etc. However, this thesis takes a big-picture approach to addressing URLLC and as a result brings together ideas from all the areas mentioned in this chapter to build towards a viable solution.

Chapter 3

Cooperative Communication Protocol Design

What are the key ingredients needed to design a communication system that caters to the requirements of ultra-reliability and low-latency? What are some of the avenues for optimization and how much can they buy us? These are the questions addressed in this chapter. Crucially, we focus primarily on combating fading by abstracting away more nuanced effects to shine light on the main pain points.

We introduce a communication protocol framework for industrial control and IoT applications that is designed to meet the stringent Quality-of-Service (QoS) requirements of low-latency and high-reliability. The protocol relies on multi-user diversity to achieve reliability without relying on time or frequency diversity created by natural motion, multipath or frequency selectivity. Our key findings are shown in Fig. 3.5, where the minimum SNR required to achieve a cycle failure probability of 10^{-9} is used to compare different protocols as the size of the network grows. We see that a one-hop scheme, one that does not use cooperative communication (the top blue line), requires an unreasonably high nominal SNR. Even idealized hybrid automatic repeat request (HARQ — the dashed red line) cannot eliminate the need to use high power to overcome a one-in-a-billion fading event. Harnessing diversity is required to do better. The purple dotted curve shows what idealized frequency hopping could achieve, assuming that nature has guaranteed sufficient frequency diversity (the number of independently faded subchannels required is labeled along the curve). For large enough networks, the black line in Fig. 3.5 shows that only slightly worse performance is available by using cooperative communication where a subset of the nodes (the number of active relays is marked on the curve) take turns to relay messages that they have heard. This does not count on the multipath environment guaranteeing a lot of frequency diversity and instead harnesses the spatial diversity that independently located nodes bring. A further 20dB of gain is possible by moving to the OccupyCoW protocol described in this chapter that combines relaying with simultaneous transmission of messages by relays, and this is what is shown by the yellow and green curves. The difference between the yellow and green curves is what can be gained by optimizing the protocol parameters, and is not as significant

by comparison.

The protocol can be optimized for bi-directional traffic by leveraging the fact that the paths for success are the same in either directions. Therefore, XOR-ing the two packets can either be used to save time (thereby, freeing up time for other less important traffic) or can be used to lower the minimum SNR required to meet the protocol performance requirements.

The rest of the chapter is organized as follows. Section 3.1 describes the setup of the network that is considered in this chapter. In the first part of the chapter, we consider the multi-user-diversity-based protocol framework only. Section 3.2 describes Occupy CoW – the multi-user-diversity-based protocol framework we propose. Section 3.3 compares the performance of our protocol to hypothetical frequency-diversity-based schemes as well as to schemes that do not leverage simultaneous transmissions. Section 3.4 examines the impact of fine-tuning the protocol parameters and explores duty-cycling to reduce power consumption and what this suggests for implementation. In the second part of the chapter, we combine the multi-user-diversity-based protocol framework with network coding to introduce the XOR-CoW protocol in Section 3.5. It gives rise to non-trivial improvement in the presence of bi-directional traffic and we examine its performance in Section 3.7. All the formulas used to generate the plots are included in Appendix A and B.

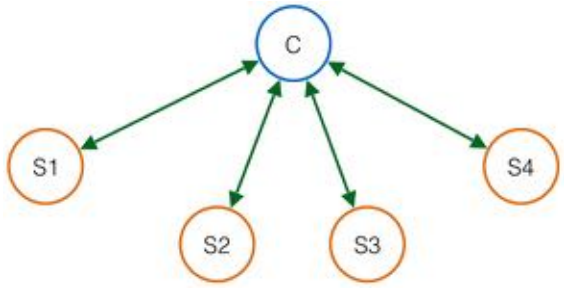
A note about using repetitions in time and frequency

This chapter and the protocol proposed here primarily focuses on using spatial diversity and does not include any repetitions in time or frequency. However, the final design that we propose in Chapter 5 includes repetitions in time and frequency to guard against unmodeled errors such as mis-synchronized clocks or stray interference events. The analysis of the scheme and the operating points will be explored in detail in Chapter 5. We briefly mention it here for providing the big-picture view but omit further discussions about time/frequency repetitions to keep the focus in this chapter on fading.

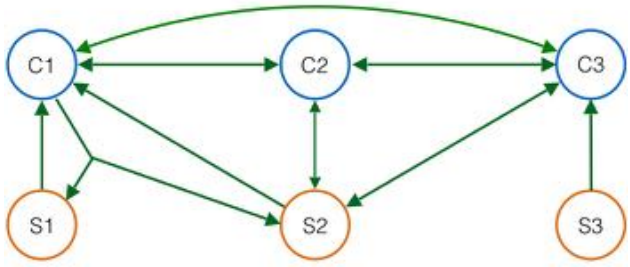
3.1 Problem Setup

We target a local wireless domain where nominally all nodes are in range, but fading might cause a pair of nodes to be unable to hear each other. The traffic patterns (what we deem the "information topology") of interest consist of steady streams of messages, each originating at possibly different nodes within the network, and each stream subscribed to by some (possibly different) subset of nodes within the network. Within a short period of time, deemed a "cycle time," every stream needs to deliver one packet reliably to its subscribers. The information topology can be arbitrary – something naturally centralized like a star topology as shown in Fig. 3.1a (e.g. with a central controller talking to many sensor/actuators collecting streams of measurements and sending streams of commands) or something more generic as in Fig. 3.1b.

The potential for deep fading is what frustrates the simple strategy of just giving each message stream its own time-slot. For example in Fig. 3.1a, if the communication link



(a) Star message flow topology



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

from 'C' to 'S3' were deeply faded, we would have a failure. We combat this by employing cooperative communication – if any of the other nodes can hear C, it can relay the relevant message streams. As reviewed in Section 2.3, cooperative communication has been well studied in the wireless literature. In this thesis, we specifically adapt it to the ultra-reliability low-latency regime.

3.2 Occupy CoW Design

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

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

- Fixed schedule: Once an initial schedule (or order) of packets has been determined, all packets are transmitted once. Then, in the same order, all nodes that have the corresponding packet simultaneously retransmit it. This is a two hop version of the protocol. For three hops, all nodes that now have the corresponding packet simultaneously retransmit it again. We have restricted the number of hops to three. This is because

in local networks where nominally nodes are in-range and thus presumably connected to each other, there is negligible improvement in performance (say SNR reduction) from going to higher hop counts. However when the networks are fundamentally wide (some flows need at least 2 hops to reach their destinations under nominal channel conditions), then going to higher hop counts would be necessary.

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

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

The protocol itself is information-topology independent, but star-topology examples will be used to explain fixed schedules in Sec. 3.2.2 and adaptive schedules in Sec. 3.2.3.

3.2.1 Resource assumptions

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

- All the nodes share a universal addressing scheme and order. Each node knows the initial order of messages being transmitted so there is no confusion about what is transmitted next. Each message packet also has the destination address attached to it. This ensures all nodes who could potentially relay the message knows the destination.
- 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 [161]. Thus we can schedule time slots for specific packets and nodes can simultaneously transmit if so desired.

Algorithm 1 Occupy CoW protocol

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

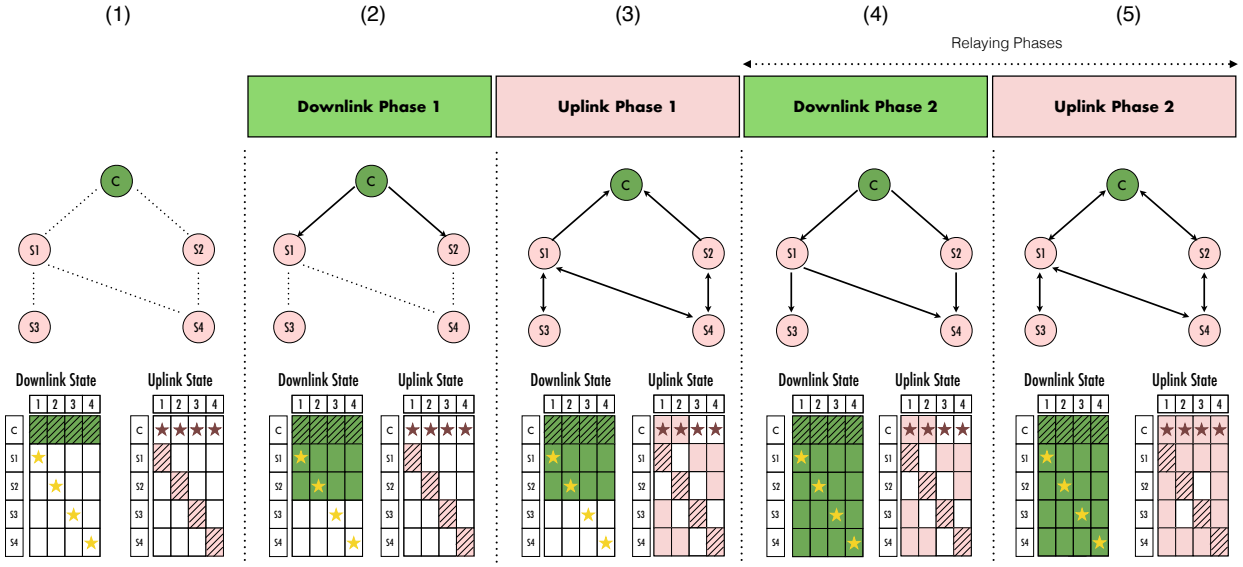


Figure 3.2: Simple example with one controller and 4 nodes. The graph illustrates which links are active during that phase. The downlink and uplink tables at each stage represent the information each node has at the end of that phase. Striped cells indicate message origins and starred cells indicate message destinations. Explained in detail in Sec. 3.2.2.

- The protocol relies on time/frequency synchronization to achieve *simultaneous retransmission* of messages by multiple relays. We assume that if k relays simultaneously (with consciously introduced jitter[1] or some other DSTC) transmit, then all receivers can extract signal diversity k without having to know in advance who is relaying or how many simultaneous transmissions they are receiving.
- (Only for adaptive-schedules) Nodes are capable of decoding variable-rate transmissions [162].

3.2.2 Fixed Schedule Example

For simplicity we focus on a simple star information topology as in Fig. 3.1a. A central controller (C) that must transmit m distinct bits (downlink messages) to each of the four nodes. Each of nodes (S1-S4) must transmit m distinct bits (uplink messages) to the controller. We define a cycle failure to be the event that at least one node fails to receive its downlink message, the controller fails to receive an uplink message from any of the nodes, or both. While there is no qualitative or quantitative difference between downlink and uplink packets, we use this terminology for ease of exposition.

[1]This jitter or explicit delay transforms spatial diversity into frequency-diversity as shown in [91]. This scheme does not require a relay to know who else is relaying alongside it, and having a long enough OFDM symbol suffices.

We will now run through a fixed schedule two-hop version of Occupy CoW on this network using Fig. 3.2. Column (1) in the figure has two components in it — the top figure shows the available communication links depicted by the dotted lines (the rest are faded out) and the bottom comprises two tables for the downlink and uplink information of each node. The table on the left is the downlink information state of each node (including the controller) and the table on the right is the uplink information state. Striped cells indicate message origins and starred cells indicate message destinations. For instance, since S1 is interested in downlink message 1 from the controller, the corresponding box in the downlink table is starred, and similarly for S2-S4. On the uplink, the controller is interested in the uplink packets from nodes S1 to S4, but these nodes do not care about each others packets, leading to stars only in the top row.

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

Phase I

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

Phase II

In this phase (also divided into Downlink and Uplink), nodes *simultaneously* transmit packets to help other nodes[2].

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

The key point here is that S3's downlink packet is retransmitted by both C and S1 using a DSTC. Since S1 has a good channel to S3 (and S4) they both can decode this. The same

[2]Section 3.3.4 discusses a version of the protocol where relays take turns instead of simultaneously relaying.

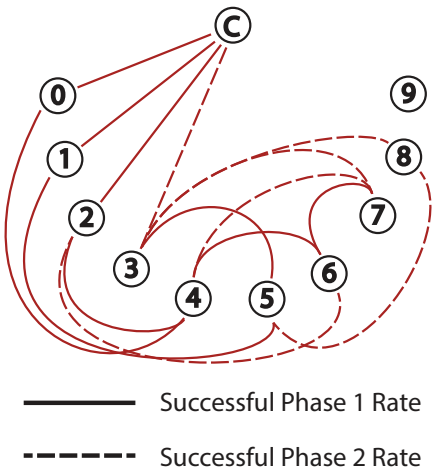


Figure 3.3: Network realization of the adaptive schedule example. The links that are present under different rates are depicted.

ensues for S4's downlink packet. At this stage, all nodes (S1 to S4) have received their downlink packets.

The final phase is Uplink Phase II (column (5)). This phase is similar to Downlink Phase II as nodes that have a message simultaneously broadcast it. S1's uplink packet is retransmitted by C, S1, S3 and S4 simultaneously using a DSTC and S2 is able to decode it. A similar procedure happens for S2's uplink packet and S1 and C are able to decode it.

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

3.2.3 Adaptive Schedule Example

We again consider a star information topology for this example. There is one controller and 10 nodes (S0 - S9). The controller has m bits of information for each node and each node in turn has m bits of information for the controller. In this example, we will consider an adaptive schedule three-hop protocol. 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. Note that the relays that help are not taking turns to help. They transmit the message *simultaneous* using a distributed space-time code (DSTC) and that is key. We describe the example in detail below.

The link realization of the network is shown in Fig. 3.3. The controller (C) has direct links to nodes S0 - S2 at the rate of Phase I. The rates of other phases depend on the number

of nodes that succeeded in Phase I – thus links that were bad under the initial rate could be good under the new rate (for example the link between C and S4). Fig. 3.15 walks through this example step-by-step and shows the information state at each node. For compactness, we have merged the two uplink and downlink tables for each node into a single table. During the downlink phases, the downlink part of the table is shown and during the uplink phases, the uplink part of the table is shown. For this example, we allocate time equally for all message phases (Downlink Phases I, II and III and Uplink Phases I, II and III) and by reciprocity assume that links present in Downlink Phase I are present in Uplink Phase I and so on.

Phase I

This phase is just like its counterpart in the fixed-schedule case — all messages get transmitted for the first time in their allotted slots. Phase I is divided into two phases – Downlink Phase I (length of T_{D_1})) and Uplink Phase I (length of T_{U_1})). In Downlink Phase I, the controller transmits the downlink packets of each of the nodes. One can further optimize this by combining multiple packets from a single node into one larger packet for practical purposes (as shown in Fig. 3.15). The controller combines the individual messages into a single packet and broadcasts it at the rate $R_{D_1} = \frac{m \cdot n}{T_{D_1}}$. In the instance depicted in Fig. 3.15, Column 1, only S0, S1, and S2 successfully receive and decode the controller's packet. Note that these three "direct links" to the controller are also depicted in Fig. 3.3. At the end of Downlink Phase I, S0, S1, and S2 have decoded both their individual messages as well as the messages intended for all of the other nodes. This is followed by Uplink Phase I. In this phase the individual nodes transmit their uplink messages in separate packets in their assigned slots.

In Fig. 3.15, Column 2, again only S0, S1, and S2 successfully transmit their messages to the controller. When a node is not transmitting, it is trying to listen for other messages – thus S4 and S0 are able to hear each other, and so on. In Fig. 3.3, we can also see the nodes which can hear each other even though they do not have anything to say to each other. All successes thus far have been due to *direct* connections between nodes and the controller. Due to this, we refer to these types of successes as "one-hop" successes.

Scheduling Phases

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

During three scheduling phases (total length T_S and each sub-phase of length $T_S/3$) the controller and the other nodes transmit short acknowledgments of $2n$ bits corresponding to n downlink packets and n uplink packets. Each phase is divided equally among the $n+1$ nodes, resulting in a scheduling rate of $R_S = \frac{2n \cdot (n+1)}{T_S/3} = \frac{6n \cdot (n+1)}{T_S}$. In Scheduling Phase I, all nodes take turns transmitting their acknowledgment (ACK) packets. For example, the

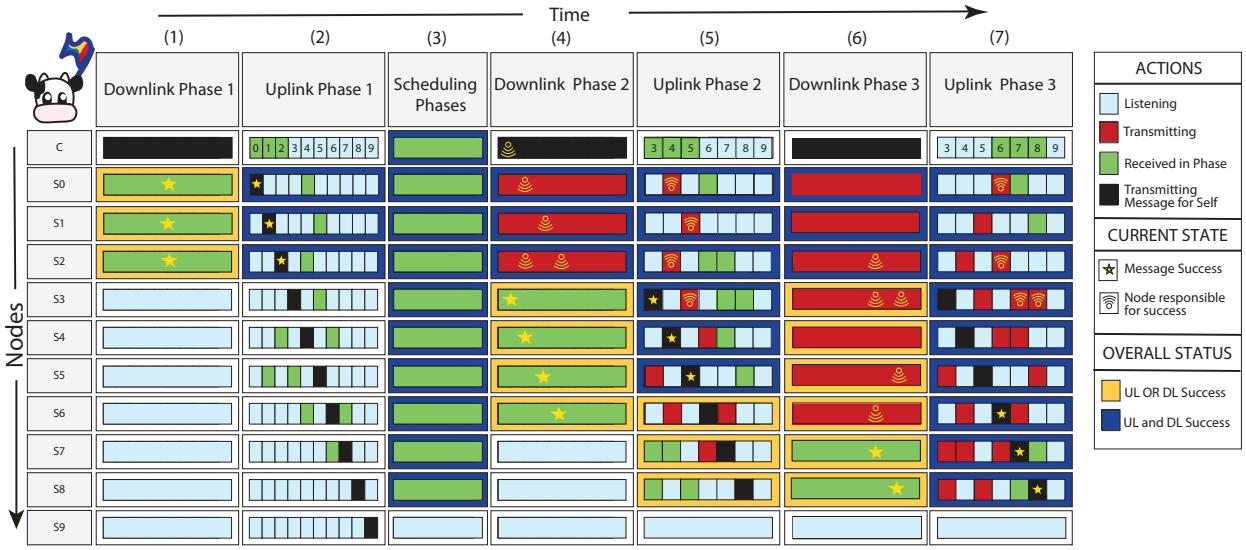


Figure 3.4: The seven phases of the Occupy CoW protocol illustrated by a representative example. The table shows a variety of successful downlink and uplink transmissions using 0, 1 or 2 relays. S9 is unsuccessful for both downlink and uplink.

controller could go first, then S0, then S1 and so on. In this example, the controller's ACK packet would be 1111111111110000000 with the first 10 ones corresponding to the downlink packets (known by assumption), the next 3 ones indicate that the controller has the uplink packets of S0 - S2 and the rest of them are zero to indicate that the controller doesn't have those packets. Similarly S0's ACK packet is 1111111111000100000 with the first 10 ones corresponding to the downlink packets (as S0 has decoded all downlink packets), the next one is for its own uplink packet, the next three zeros for the uplink packets of S1 - S3 are followed by a one for S4's uplink packet and the rest are zeros corresponding to S5's - S9's uplink packets. After all ACK packets have been transmitted once, Scheduling Phase I ends.

In Scheduling Phases II and III, these short ACK packets are retransmitted in a round-robin fashion by the nodes which have heard them using a DSTC in a fashion identical to fixed-schedule Occupy CoW Phase II. For example, C's ACK packet is simultaneously transmitted by C, S0, S1, S2 and S3, S0's ACK packet is transmitted by C, S0 and S4 and so on. These ACK packets are relayed once again in Scheduling Phase III so that all packets reach all nodes. At the end of Scheduling Phase III, all nodes have 'global ACK information' with high probability and are ready to adapt the retransmission schedule so that slots are not wasted on already successful data packets in Phases II and III. Fig. 3.15, Column 3 shows the information state of the nodes after the end of the scheduling phases. All nodes except S9 have received every ACK packet and know the schedule for the rest of the cycle. However, S9 has not received the scheduling information and therefore does not transmit anything for the rest of the cycle in order to avoid any interference to other packets.

Phase II

After the Scheduling Phases, we have Phase II of data transmission. The messages that have already succeeded are the downlink and uplink packets of S0 - S2. Thus, the retransmission schedule only allocates time for the downlink packets of S3 - S9 and the uplink packets of S3 - S9. For illustrative purposes, we divide this phase further into two sub-phases – Downlink Phase II (length T_{D_2}) and Uplink Phase II (length T_{U_2}). In general, if a_D packets have succeeded in Downlink Phase I and a_U have succeeded in Uplink Phase I, then the rates of transmission in these phases are: $R_{D_2} = \frac{m \cdot (n - a_D)}{T_{D_2}}$ and $R_{U_2} = \frac{m \cdot (n - a_U)}{T_{U_2}}$. The relaying in these phases is the same as the relaying in Phase II of the fixed schedule protocol – except with a modified schedule. Because the schedule has adapted, it is possible that nodes that were initially unable to directly connect to the controller may now be able to, *if* the rate during any of these phases is lower than that of the first. This may occur if enough nodes are successful in the first phase since fewer messages must now be sent or if the time allocated for the phases T_{D_2} or T_{U_2} is greater than T_{D_1} or T_{U_1} respectively resulting in a lower rate.

Downlink Phase II is depicted in Fig. 3.15, Column 4. We see that node S3 gets its downlink message directly through the controller (due to reduced rate), and this is reflected in the dashed representation of the connection between node S3 and the controller in Fig. 3.3. As S0 and S2 are able to reach S4, it successfully receives the controller's message in two hops via S0 and S2 and so on. At the end of Downlink Phase II, nodes S0, S1, S2, S3, S4, S5, and S6 have successfully received their downlink messages. Uplink Phase II is similar and is depicted in Fig. 3.15, Column 5 and the same set of nodes' uplink packets have successfully been delivered to the controller.

Phase III

Again, this phase is divided into Downlink Phase III (length T_{D_3}) and Uplink Phase III (length T_{U_3}) with rates $R_{D_3} = \frac{m \cdot (n - a_D)}{T_{D_3}}$ and $R_{U_3} = \frac{m \cdot (n - a_U)}{T_{U_3}}$ respectively. In these phases, 'three-hop' successes occur. For example, in Fig. 3.15, Column 6, S8 successfully receives its downlink packet through S5 (the full path is C $\rightarrow$ S1 $\rightarrow$ S5 $\rightarrow$ S8). The uplink counterpart is similar to downlink and at the end of Phase III, all nodes except S9 have received their downlink packet and have successfully relayed their uplink message to the controller. The example depicted in Fig. 3.3 and 3.15 is a failed instance of the protocol since the node S9 has not received its downlink message and the controller has not received S9's uplink packet.

3.2.4 Information topology-dependent optimization

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

For example, each node can piggyback a one bit ACK for their downlink packet onto their uplink message. Then the extra scheduling phases can be simplified to a single phase where the controller processes all the ACKs (received as well as not received) into a single

packet that just lists which messages require retransmission. Then, all the nodes that can hear the controller get to know the schedule. These nodes can then modify the downlink packet (culling already successful messages and appending the global schedule to it) and simultaneously broadcast it. The nodes that can hear this first set of relays can then not only decode the downlink messages (despite not knowing the schedule) but also figure out the schedule itself so that they can help in the next phase. At this stage, nodes only reachable via three hops do not have the schedule and to propagate the information to them, we switch the order of Uplink Phase II and Downlink Phase III (Downlink Phase III directly follows Downlink Phase II). The nodes reachable by two hops broadcast the downlink messages (with embedded schedule) again so that it can be heard by the nodes only reachable by three hops. Thus, even though all the nodes did not know the schedule at the beginning of Downlink Phase II, they do get to know it by the end of Downlink Phase III and that is sufficient for enabling Uplink Phases II and III. As you can see, this optimization exploits the star nature of the information topology to shorten the scheduling phase and furthermore, the total traffic dedicated to scheduling is substantially reduced.

3.3 Analysis of Occupy CoW

We explore the Occupy CoW protocol with parameters in the neighborhood of a practical application, the industrial printer case described in Weiner et al. [6]. In this particular scenario, the SERCOS III protocol [39] supports the printer's required cycle time of 2 ms with reliability of 10^{-8}. Consequently, we target a 10^{-9} probability of failure for Occupy CoW. The printer has 30 moving printing heads that move at speeds up to 3 m/s over distances of up to 10 m. Every 2 ms cycle, each head's actuator receives 20 bytes from the controller and each head's sensor transmits 20 bytes to the controller. The amount of information transmitted by the controller in a single cycle is $20 \times 30 = 600$ bytes. The total amount of sensor information transmitted by the heads to the controller in a single cycle is also $20 \times 30 = 600$ bytes. Therefore a total of 1200 bytes or 9600 bits of information is sent during a cycle of 2 ms which corresponds to a desired goodput of $\frac{9600}{(2\times10^{-3})}$ bit/sec $= 4.8$ Mbit/sec. If we assume access to a single dedicated 20 MHz wireless channel, this 4.8 Mbit/sec corresponds to an overall net spectral efficiency of $\frac{(4.8\times10^{6})}{(20\times10^{6})} = 0.24$ bits/sec/Hz.

3.3.1 Behavioral assumptions for analysis

The following behavioral assumptions are added to the resource assumptions in Sec. 3.2.1.

- We assume a fixed nominal SNR and independent Rayleigh fading on each link. We assume that each node has perfect local receiver CSI knowledge of the 'good' channels to itself i.e., those net channels on which messages may be decoded. No assumptions are made for knowing the CSI of deeply faded channels. No global CSI knowledge

is assumed. We defer the issue of the overhead associated with acquiring this local knowledge to future work.

- We assume a single tap channel — performance would improve if we reliably had more taps/diversity. Because the cycle-time is so short, the channels' coefficients do not change in a cycle and hence we use the delay-limited-capacity framework [163, 164].

- A link with complex fade h and bandwidth W is deemed good (no errors or erasures) if the transmission rate R is less than or equal to the link's capacity $C = W\log(1 + |h|^2 SNR)$. Consequently, the probability of link failure is defined as

$$p_{link} = P(R > C) = 1 - \exp\left(-\frac{2^{R/W} - 1}{SNR}\right). \tag{3.1}$$

  From the above equation we see that if R decreases, then the probability that the capacity C is less than R also decreases (the capacity C did not change, only R did). In other words, a channel which was unable to support a given rate might be able to support a lower rate.

- We also assume channel reciprocity – if a channel has fade h between node A to B, then it is also h from B to A as well.

- If there are k simultaneous transmissions[3], then each receiving node harvests perfect sender diversity of k. For analysis purposes this is treated as k independent tries for communicating the message that only fails if all the tries fail.

- We do not consider any real implementation effects on decoding to abstract away some of the nuanced effects. This is partially justified in spirit by Yang et al. [166] where they show that the channel dispersion is zero for quasi-static fading channels. However, we will explore finite-blocklength effects in Chapter 5. At the most basic level, we can think about a code in terms of its gap to capacity at the desired reliability. If a code is 3dB away from capacity, then one can add 3dB to all power requirements calculated assuming infinite-blocklength and the protocol should work. However, the effects are more nuanced. For infinite-blocklengths, codes either work or don't work depending on the rate and channel capacity. In the case of finite-blocklength the performance degrades more smoothly. Even a link that cannot deliver the final target reliability for the error correcting code under consideration still might be enough to allow a node to decode and become a potential relay. More potential relays mean that we can more easily count on getting higher receiver power in the relaying phase – thus getting higher reliability in the relaying phase. This tradeoff creates "partial credit" which allows us

[3]The cyclic-delay-diversity space-time-coding schemes we envision make the effective channel response longer. This can push the PHY into the "wideband regime", and a full analysis must account for the required increase in channel sounding by pilots to learn this channel [165]. 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.

to perform better than what a conservative Shannon-capacity-plus-gap-based analysis would suggest. We revisit this in Chapter 5 and find that the demands are different in different phases of a diversity-seeking cooperative protocol. A related assumption is that no transmission or decoding errors are undetected [167] — a corrupted packet can be identified (say using a 40 bit hash) and is then completely discarded.

We derive the probability of failure for a two-hop downlink scheme in Sec. 3.3.2, a union bound of the failure probability for a generic information topology in Sec. 3.3.3, a similar bound on the probability of failure for relaying with non-simultaneous transmissions in Sec. 3.3.4 and the probability of failure for frequency hopping repetition coding in Sec. 3.3.5. These equations are used to derive results in Sec. 3.3.6. Additional derivations are found in Appendix A.

3.3.2 Two-hop downlink (star information topology)

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

3.3.2.1 Fixed schedule scheme

In the fixed schedule two-hop scheme each message gets sent twice whether or not it was successful in the first try. The first phase's rate $R_{D_1} = \frac{n \cdot m}{T_{D_1}}$ and the corresponding probability of link failure is p_1. The second phase rate $R_{D_2} = \frac{n \cdot m}{T_{D_2}}$ (as all messages get sent two times) and the corresponding probability of link failure is p_2. Let the nodes successful in Phase I be in a set $\mathcal{A}$ (with cardinality represented by the random variable A and a representing a specific size). The nodes in the set $\mathcal{S} \setminus \mathcal{A}$ succeed in Phase II only if they connect to either the controller or at least one of the nodes in $\mathcal{A}$. The Rayleigh fading assumption tells us that the probability that a link fails in Phase II given it failed in Phase I is given by $p_c = \min\left(\frac{p_2}{p_1}, 1\right)$. Then, the probability of not connecting to $\{\text{controller} \bigcup \mathcal{A}\}$ in Phase II is $p_2^a \cdot p_c$. The probability of 2-phase downlink system failure is thus:

$$\mathrm{P}(\text{fail}) = \sum_{a=0}^{n-1} \left(\binom{n}{a} (1-p_1)^a (p_1)^{n-a} \right) \left(1 - (1 - p_2^a \cdot p_c)^{n-a}\right). \tag{3.2}$$

3.3.2.2 Adaptive schedule scheme

In the adaptive schedule two-hop scheme only the messages that were unsuccessful in Phase I get sent again in Phase II. The first phase is exactly like the fixed-schedule scheme. The time allocated for Phase II and the number of first phase successes a dictate the Phase II rate $R_{D_2} = \frac{(n-a) \cdot m}{T_{D_2}}$. The corresponding probability of link failure is denoted $p_2(a)$ (the (a)

is used to indicate that it is a function of a). As in the fixed-schedule case, the probability that the controller to node link fails in Phase II given it failed in Phase I is given by $p_c(a) = \min\left(\frac{p_2(a)}{p_1}, 1\right)$. Then, the probability of not connecting to $\{\text{controller} \bigcup \mathcal{A}\}$ in Phase II is $(p_2(a))^a \cdot p_c(a)$. The probability of 2-hop downlink system failure is thus:

$$\text{P(fail)} = \sum_{a=0}^{n-1} \left(\binom{n}{a} (1-p_1)^a (p_1)^{n-a} \right) \left(1 - (1 - (p_2(a))^a \cdot p_c(a))^{n-a}\right). \tag{3.3}$$

Notice that in the above derivation, we omitted the role of scheduling information even though it is actually crucial for adapting the rate of transmission in Phase II. This is because we assume that the scheduling phases are allocated sufficient time such that the scheduling phase rate is lower than the rates of transmission in any of the other phases. Therefore any scheduling error in the protocol is also going to manifest as a delivery failure for a message packet. A property of ACK packets is that they want to reach all the nodes in the network. Therefore, a more detailed analysis for scheduling failure is as derived in the next subsection discussing the union bound, which is how we can upper bound the probability of failure for a generic topology (Sec. 3.3.3).

3.3.3 The union bound and generic information topologies

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

$$\begin{aligned} q_s &= \text{P(direct link)} \times \text{P(success|direct link)} + \text{P(no direct link)} \times \text{P(success|no direct link)} \\ &= ((1-p) \times 1) + \left(p \times \left(\sum_{j=1}^{n-2} \binom{n-2}{j} (1-p)^j p^{n-2-j} \left(1 - p^j\right) \right) \right). \end{aligned} \tag{3.4}$$

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

$$\text{P(failure)} = s \times d \times (1 - q_s). \tag{3.5}$$

As mentioned in Sec. 3.3.2.2, ACK information from each node has to disseminate throughout the network – therefore if there are n nodes in the network, there are n ACK messages, and the number of subscribers for each is $n-1$. Consequently, the probability of ACK dissemination failure can be bounded by the union bound derived in this section. Equations for other error probabilities are derived similarly and can be found in Appendix A.

3.3.4 Non-simultaneous relaying

In the design of Occupy CoW, we employed simultaneous relaying using a DSTC to harvest the benefits of spatial diversity while staying within latency limits. To tease apart the impacts of relaying and simultaneous transmission, it is useful to analyze relaying without simultaneous transmissions. To have relays taking turns within a fixed-schedule scheme, the basic requirement is making the time-slots shorter. Suppose that we have r potential relays for every data packet, designated in advance for every message stream — these will be the only nodes that will listen for this stream and each will have an assigned slot in which to potentially relay it if they heard it. This means that if there are k (either 2 or 3) hops, then each data packet will have a footprint of $1 + (k-1)r$ time slots. This means that for s message streams, if the total cycle time is T and each data packet is m bits long, then the link data rate is $R = \frac{s \cdot m \cdot (1+(k-1)r)}{T}$.

Consider a single message stream and let $q_s(p, r)$ denote the probability of success to a single destination where p is the probability of link error given the rate and SNR. The analysis for the two-hop case follows the union-bound case in Eq. (3.4) with the number of potential relays r playing the role of $n-2$ above. Consequently:

$$q_s(p, r) = ((1-p) \times 1) + \left(p \times \left(\sum_{j=1}^{r} \binom{r}{j} (1-p)^j p^{r-j} \left(1 - p^j\right) \right) \right). \tag{3.6}$$

The union bound argument applies and so Eq. (3.5) continues to bound the probability of error, just with the slightly revised expression in Eq. (3.6) for q_s.

3.3.5 Frequency-hopping schemes

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

$$\mathrm{P}(\text{fail}, k) = 1 - \left(1 - (p_{sc}(k))^k\right)^s. \tag{3.7}$$

The same tension exists. A larger k allows us to harness more diversity while also forcing the instantaneous rate to be higher.

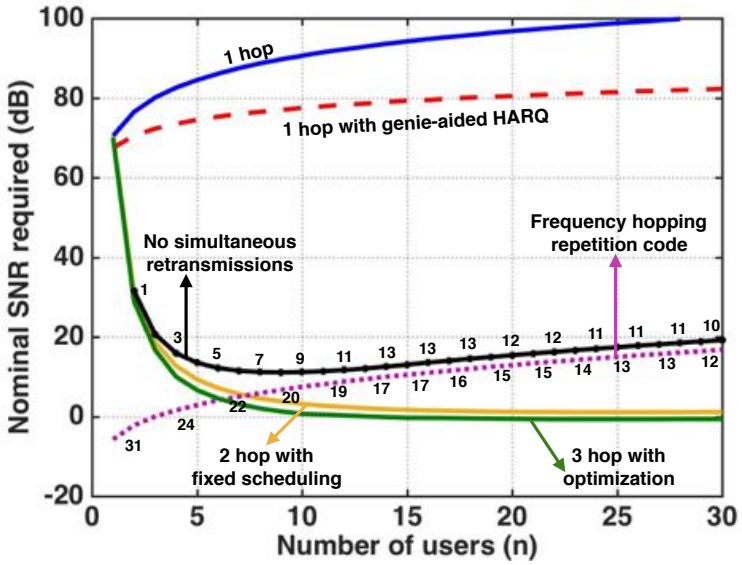


Figure 3.5: The performance of Occupy CoW as compared with reference schemes for $m = 160$ bit messages and $n = 30$ nodes with 20MHz and a 2ms cycle time, aiming at 10^{-9}. The numbers next to the frequency-hopping scheme represent the amount of frequency diversity needed.

3.3.6 Results and comparison

Following Weiner et al. [6] and the communication-theoretic convention, we use the minimum SNR required to achieve 10^{-9} reliability as our metric to compare fixed-schedule 2-hop and adaptive-schedule 3-hop Occupy CoW to four other baseline schemes. We calculate the minimum SNR required by various protocols to meet the specs in the following fashion. Assuming a fixed nominal SNR, we calculate the probability of failure for the protocol under consideration. Then, we search for the smallest value of nominal SNR that meets the reliability requirement of 10^{-9}. Fig. 3.5 looks at performance (the minimum SNR required on the y-axis) for a star information topology with a central node sending $m = 160$ bit messages to n other nodes and receiving the same size messages from them. All this has to be completed within 2ms and a bandwidth of 20MHz. Initially the minimum required SNR for Occupy CoW decreases with increasing n, even through the required throughput increases as $m \cdot n$, but the curves then flatten out. The gains of multi-user diversity eventually give way and the required SNR starts to increase for large n as the required spectral efficiency increases.

The topmost blue solid curve in Fig. 3.5 shows performance of the protocol restricted to just the first hop of Occupy CoW with one slot per message. The required SNR shoots off the figure for two reasons: (a) because the throughput increases linearly with the number of nodes and (b) to have the system probability of failure stay controlled with more messages

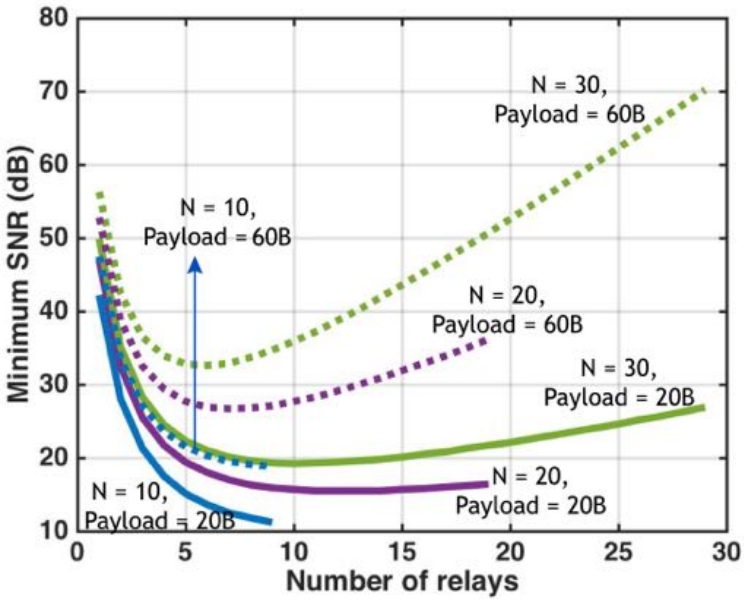


Figure 3.6: For non-simultaneous relaying, the minimum SNR required to achieve a 10^{-9} probability of system failure for different network and payload sizes as the number of nominated relays vary.

to transmit, each individual message must be that much more reliable. The second scheme (red dashed curve) is purely hypothetical. It allows each message to use the entire 2ms time slot for its own uplink and downlink message but without any relaying and thus also no diversity. This bounds what could possibly be achieved by using adaptive HARQ techniques and shows why harnessing diversity is essential. This is rising only because of effect (b) above.

The third reference scheme is the non-simultaneous relaying scheme described in Sec. 3.3.4 and plotted in Fig. 3.5 by the black curve with markers. We see that this curve is always above the Occupy CoW lines — showing the quantitative importance of simultaneous relaying. The curve is annotated with the best number of relays r that minimizes the SNR required. As r increases, the available spatial diversity increases, but the added message repetitions force the link data rates higher.

Fig. 3.6 explores the effect of the number of relays allocated on the required SNR for the scheme in Sec. 3.3.4. For a network size of $n = 30$, a payload size of 60B per message would select $r = 6$ as the optimal number of relays. Reducing the network size to 10 makes $r = 9$ be the optimal number of relays. Compare this to a payload size of 20B and $n = 30$ — not only is the optimal number of relays the same, the entire curve is very close to that for $n = 10$ with payload $60B$. This is because for the same number of relays, the link data rates are the same and the factor of 3 difference in the number of message streams demands a factor 3 reduction in the probability of error per message — which for nine relays is accomplished for

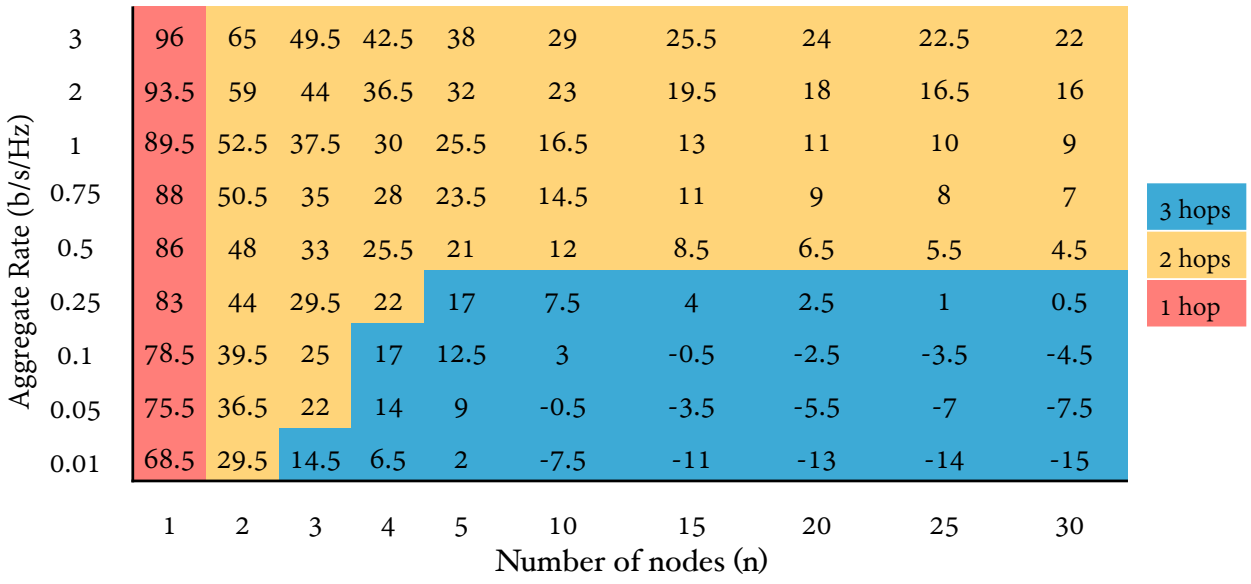


Figure 3.7: The number of hops and minimum SNR to be operating at to achieve a high-performance of 10^{-9} as aggregate rate and number of users are varied. Here, the time division within a cycle is unoptimized. Uplink and downlink have equal time, 2-hops has a 1:1 ratio across phases, and 3-hops has a 1:1:1 ratio for the 3 phases. The numbers here are for a star information topology but as the next figure shows, they would not be much different for generic topologies.

less than 1dB. Given a large enough network, the optimum number of relays seems to depend primarily on the aggregate rate. For a high-aggregate rate, we choose a smaller number of relays and for a lower one, we pick more relays.

The last reference scheme (the purple dotted line in Fig. 3.5) represents the hypothetical frequency-hopping described in Sec. 3.3.5. As the number k of frequency hops increases, the available diversity increases, but the added message repetitions force the instantaneous link rates higher, just as additional relays do for non-simultaneous relaying. For low n we prefer more frequency hops because of the diversity benefits. The SNR cost of doing this is not so high because the throughput is low enough (requiring a spectral efficiency less than 1.5bits/s/Hz) that we are still on the cusp of the energy-limited regime of channel capacity. For fewer than 7 nodes, this says that using frequency-hopping is great — as long as we can reliably count on 20 or more guaranteed independently faded sub-channels to repeat across. After 7 nodes, notice the frequency-hopping scheme is paralleling the non-simultaneous relaying scheme in Fig. 3.5. However, frequency hopping is optimized with more diversity and lower SNR because harnessing multiuser diversity requires the first-hop to actually reach enough relays to be able to use the reserved slots while frequency-diversity is just assumed to always be available. Fig. 3.5 also compares the fixed-schedule two-hop Occupy CoW protocol with equal phase lengths to an adaptive three-hop scheme optimized

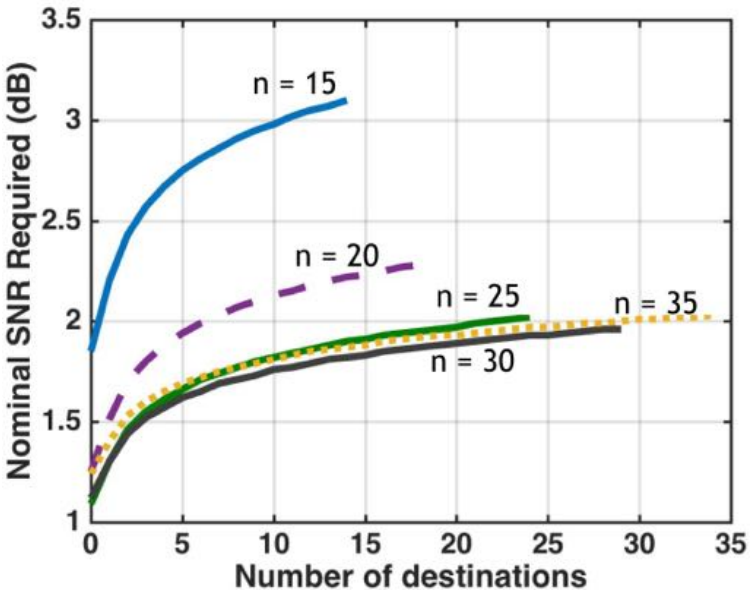


Figure 3.8: Number of destinations vs SNR required for different network sizes for $m = 160$ bit messages and $n = (15, 20, 25, 30, 35)$ nodes with 20MHz and a 2ms cycle time, aiming at 10^{-9} probability of failure. The SNR values at "0" destinations represents the SNR required for a star information topology.

to minimize SNR. We see that these are very close to each other and the choice between these is not as important as harnessing diversity and taking advantage of simultaneous transmissions. This is discussed in detail in the next Section 3.4.

It turns out that the aggregate goodput required (overall spectral efficiency considering all users) is the most important parameter for choosing the number of relay hops in our scheme. This is illustrated clearly in Fig. 3.7. This table shows the SNR required and the best number of hops to use for a given n. With one node, clearly a 1 phase scheme is all that is possible. As the number of nodes increases, we transition from 2-phase to 3-phase schemes being better. For $n \geq 5$, aggregate rate is what matters in choosing a scheme, since 3-phase schemes have to deal with a $3\times$ increase in the instantaneous rate due to each phases' shorter time, and this dominates the choice. In principle, at high enough aggregate rates, even the one-hop scheme will be best with enough users. But when the target reliability is 10^{-9}, this is at absurdly high aggregate rates[4]. In the practical regime, diversity wins.

We now consider the case of a generic non-star topology using Eq. (3.5). Figure 3.8 considers the SNR required for a varying number of destinations for different network sizes. The number of destinations per message ranges from 1 to $n-1$. For comparison purposes, at "0" destinations, we have plotted the SNR required for the star information topology. There

[4]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.

is an SNR 'penalty' for each message having multiple destinations but even when everyone wants to hear everything, this penalty is quite modest. The case of $n-1$ destinations is similar to simply reducing the tolerable probability of failure by a factor of $1/n$. The extra SNR required is on the order of 1dB for medium to large network sizes because of the ample diversity available.

3.4 Optimization of Occupy CoW

3.4.1 Phase-length Optimization

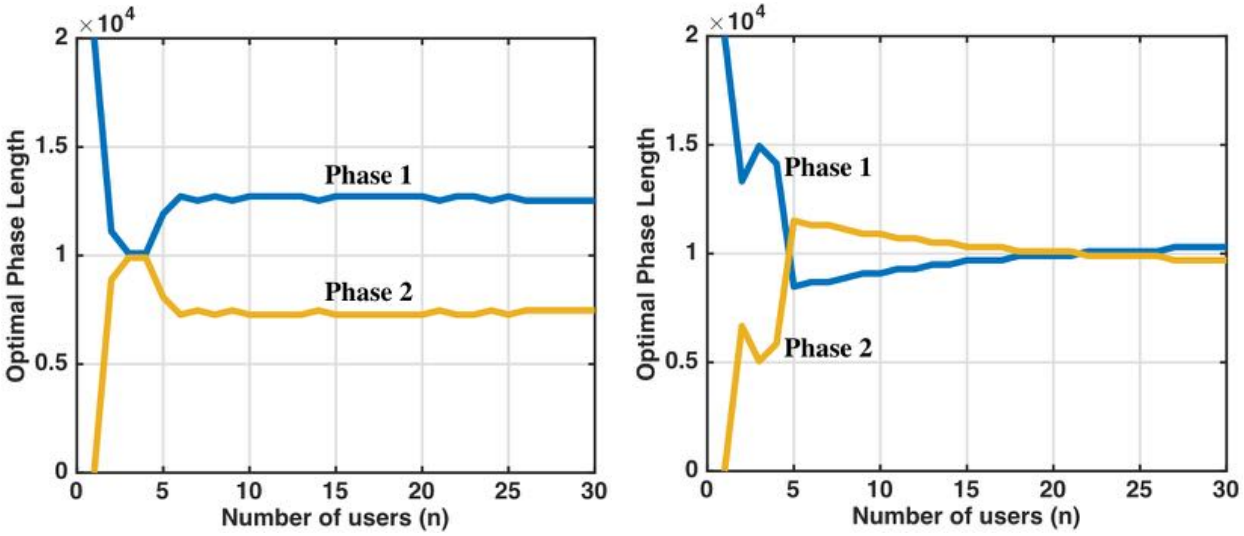
The protocols we have described come with the choice of the number of phases (2 or 3) and a choice of fixed or adaptive schedule. Furthermore, there is the choice of the time allocated for different phases. How does one pick the 'right' parameters? Does it matter? To answer that, we compare the performance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

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

3.4.2 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 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 $T_{D_1} + T_{D_2} = 1\text{ms}$ and $T_{U_1} + T_{U_2}x = 1\text{ms}$. 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 3.9a 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



(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.

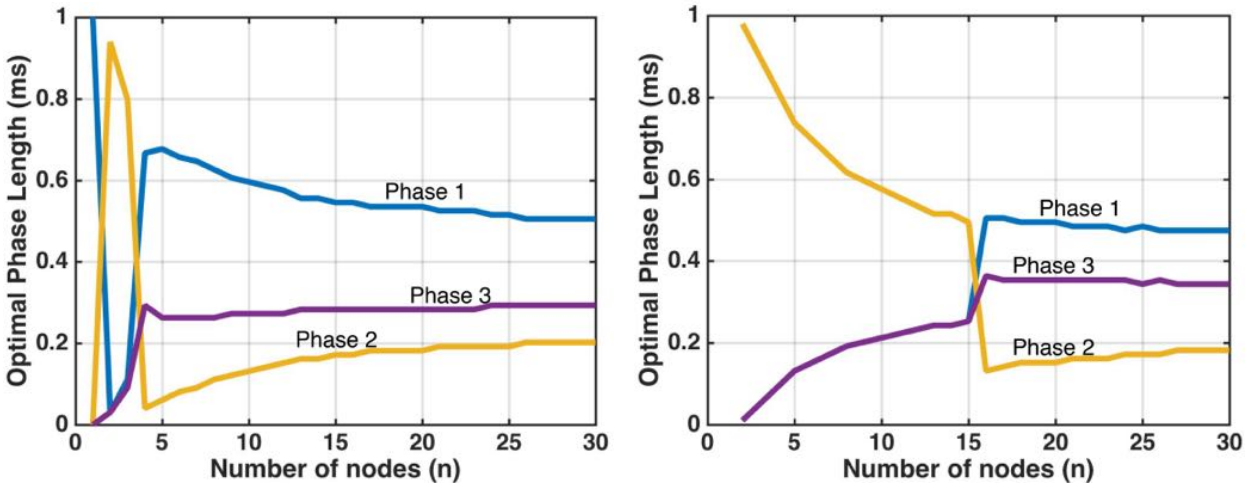
Figure 3.9: Optimal phase allocation for 2-hop protocol. Parameters: 160 bit messages, 30 users, 2×10^4 total bits.

size is reduced, we can maintain a coding rate comparable with phase I with a smaller time.

Uplink: Figure 3.9b 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.

3.4.3 Phase length allocation in 3-hop protocol

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



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

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

Figure 3.10: Optimal phase allocation for three-hops with 160 bit messages, 30 users, 2×10^4 total cycle length.

phase I, II and III of uplink be T_{U_1}, T_{U_2} and T_{U_3} respectively such that $T_{U_1} + T_{U_2} + T_{U_3} = 1$ms.

Downlink: Figure 3.10a shows the optimal allocation of time for phase I, II and III for downlink. The optimization suggests that phase I should be the longest, phase II the shortest and phase III in between (except for network sizes 1 and 2 where the optimal strategies are 1 hop and 2 hop respectively). Phase III is longer than Phase II to make sure that the messages reach everyone possible as more links open up during phase III. Phase I is longest to ensure that the messages are initially successfully decoded by enough nodes to ensure maximal spread.

Uplink: Figure 3.10b shows the optimal allocation of time for phase I, II and III for uplink. In uplink, the critical paths are the ones connecting to the controller rather than the inter-node links. Hence, phase III is allocated more time than in the downlink Phase.

3.4.4 How much SNR does optimization save?

For concreteness, let us consider the downlink side. (Uplink is similar.) Figure 3.11 considers different phase length allocations including the optimal phase length allocation and several other suboptimal allocations. For a star network of 30 nodes, we see that the difference between the various allocations and schemes is minimal. While the best SNR is -1.3db (solid blue curve with markers), the SNR required at phase allocation $10 : 3 : 4$ with adaptive

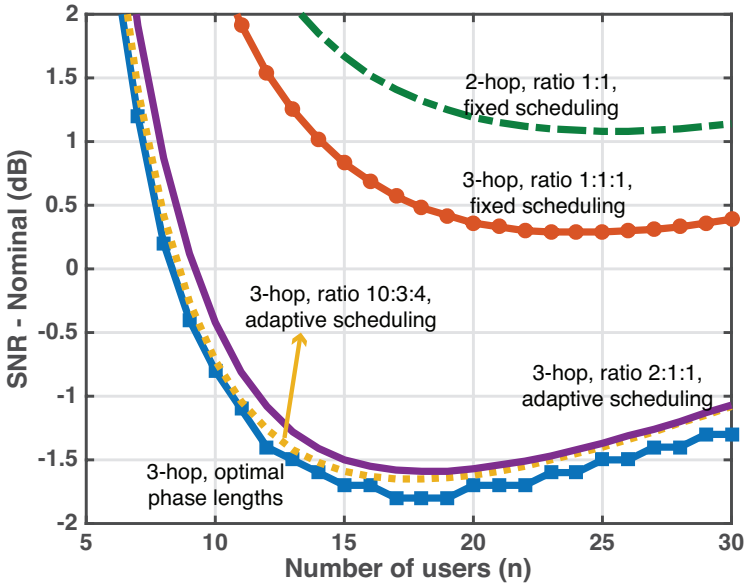


Figure 3.11: Comparing the SNR required for optimum downlink phase length allocation and a few non-optimal allocations.

scheduling is -1.08db (dotted yellow curve), and the SNR required for the simple allocation of $2:1:1$ with adaptive scheduling is -1.06db (solid purple curve). Even for a naive scheme of $1:1:1$ with fixed scheduling, the SNR required is 0.39dB (solid red curve with markers). Note that these results are for a star information topology which is the 'best' case in terms of the SNR required. As the benefits of optimization are marginal in the best case, the benefits in a generic topology are even more negligible. Furthermore, adaptive scheduling is a harder problem in a non-star topology as one cannot mostly piggyback the relevant ACK or scheduling information onto packets that would be sent anyway, as discussed in Section 3.2.4. Consequently, we conclude that though we have many knobs to turn which can optimize the performance of the protocol in terms of required SNR, the benefits for that metric are not going to be that substantial. This is not to say that there might not be other reasons for wanting to use adaptive scheduling — e.g. to support additional best-effort traffic by harvesting time-slots that are not needed for relaying time-critical packets. However, that is beyond the scope of this thesis.

3.4.5 Power consumption and the effect of duty cycling

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

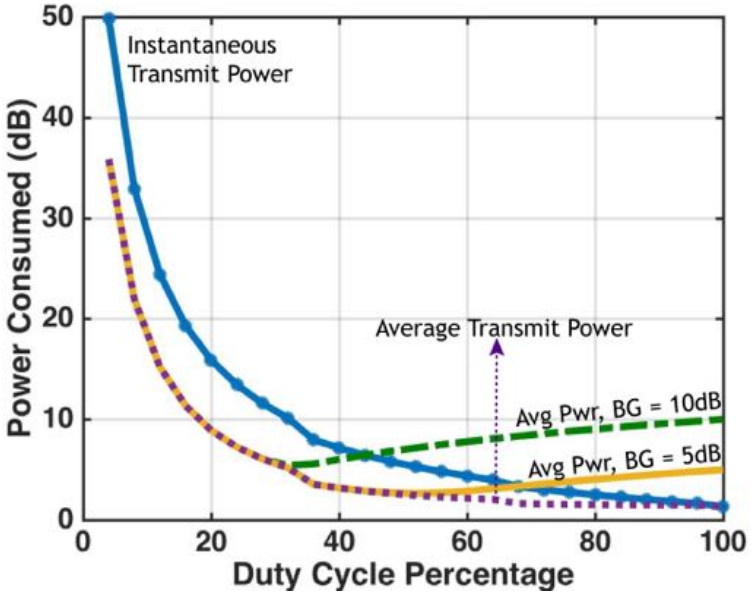


Figure 3.12: Effect of duty cycling percentage (i.e. time awake) on the power required for different on-time percentages for $m = 160$ bit messages and $n = 30$ nodes with 20MHz and a 2ms cycle time, aiming at 10^{-9} probability of failure.

question, we first modify the protocol to handle duty-cycling by using the ideas used to understand non-simultaneous relaying.

We dedicate a percentage of nodes per message as pre-allocated potential relays (say $x\%$). These wake up during the message's transmission – they either listen for the message or simultaneously re-transmit the message if they have it. The equation (3.4) can be modified so that the maximum number of relays is not $n-2$ but $r = \left\lceil \frac{x\times(n-2)}{100} \right\rceil$. Thus the probability of success of a single message-destination pair $q_{ds}(r)$ is:

$$q_{ds}(r) = ((1-p)\times 1) + \left(p \times \left(\sum_{j=1}^{r} \binom{r}{j} (1-p)^j p^{r-j} \left(1-p^j\right) \right) \right) \tag{3.8}$$

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

$$\mathrm{P(failure)} \leq s \times d \times (1 - q_{ds}(r)). \tag{3.9}$$

Fig. 3.12 shows the power consumed to reach the target reliability as a function of the time awake (duty cycle percentage). The blue curve plots the power consumed by a node when awake (in units of received SNR) in order to meet the required reliability. The purple dotted line takes into account the percentage of time the node is asleep, and plots the

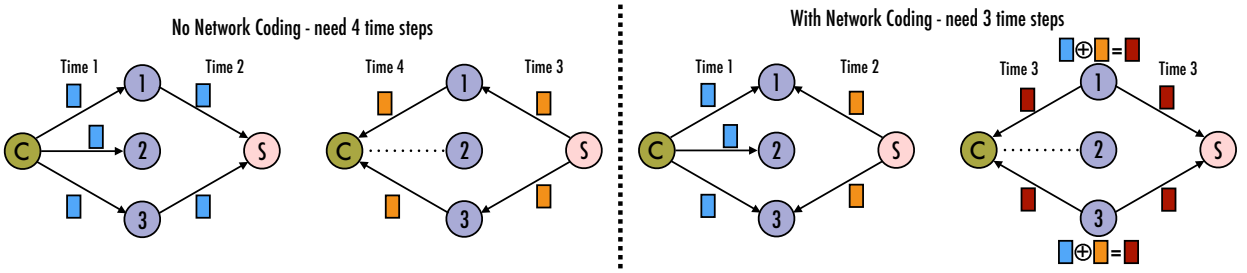


Figure 3.13: Illustration of network coding along with simultaneous retransmissions where the C and S nodes have information to convey to each other through 3 relays 1 - 3. The bold lines are active links and the dotted lines are inactive links. The blue packets are the downlink packets, the orange packets are the uplink packets and the maroon packets are the XORed packets. The XOR scheme can communicate the same amount of information in a shorter time because the uplink and downlink demands are satisfied simultaneously.

average transmit power used. Because this is minimized at 100% duty cycle, if transmit power consumption were all that mattered, it would not be worth going to sleep at all. To get a more refined answer, we recognize that there is some level of background power consumption in the wireless circuitry which accounts for listening and encoding/decoding processes whenever the node is awake [168]. The green line is the average total power consumed assuming a background power consumption of 10dB (i.e. the background power is the same as what the transmit power would be to give a 10dB SNR.) and the yellow line is for 5dB background power consumption. These plots reveal an easy rule of thumb about the desired operating point – operate with a duty-cycle percentage such that the transmit power required is equal to the background power.

3.5 Network Coding Based Optimization

In this section, we use network coding with Occupy CoW. Network coding is generally used to increase network throughput, sometimes at the cost of increased latency, but we show how to use network coding to use this improved throughput to decrease latency and reduce SNR required to meet the specifications.

How does network coding work? Consider two nodes (say A and B) that have messages to each other i.e., node A has a message for B and node B has a message for A. If the direct channel exists (link AB), then A's message to B as well as B's message to A succeeds in reaching the destination. If there is no direct channel, then A's message to B may succeed if there is at least one node (say C) that has connection to both A and B. If there exists such a node, then both A's message to B *and* B's message to A succeeds via the same node (or set of nodes). Essentially, when there is a bi-directional traffic, the paths of 'success' in both direction are the same. When we have such bi-directional traffic patterns, then relay nodes can 'XOR' the packets and broadcast the resulting packet *simultaneously* using a DSTC as

shown in Fig. 3.13. This is what we leverage in XOR-CoW – opportunistically network code packets. We show that integrating network coding with cooperative communication brings down the SNR required to meet the QoS requirements even more than Occupy CoW protocol approach under ideal conditions.

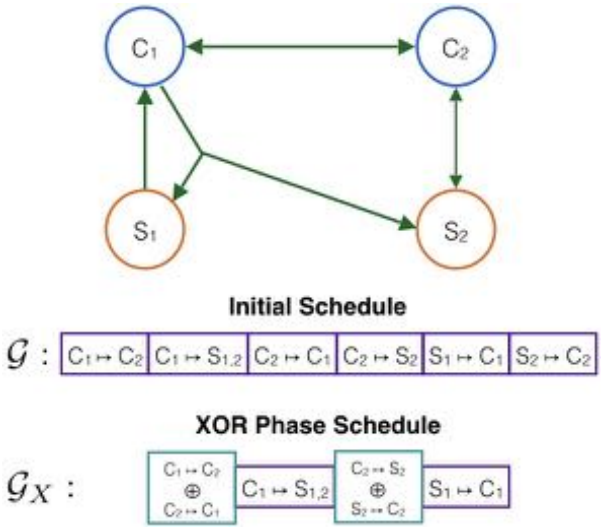
In this section, we will modify Occupy CoW to include network coding to create XOR-CoW protocol. We analyze the performance of XOR-CoW under a communication theoretic and delay-limited capacity framework. We compare it with Occupy CoW and optimize its parameters to show that XOR-CoW is relatively insensitive to parameter choices just like Occupy CoW. Most of the benefit comes from cooperative communication and network coding, so implementing more complicated schemes is not justified.

3.6 XOR-CoW Framework

The XOR-CoW protocol exploits multi-user diversity as well as side information at destination nodes by using simultaneous relaying combined with network coding to enable ultra-reliable communication. The general setup considered is that the network consists of n nodes and each message stream (size m bits) must reach its possibly many destinations within a cycle of time T. The resource assumptions are the same as ones made for Occupy CoW in Sec. 3.2.1 Network coding also provides throughput benefits (and as a result reduction in latency or reduction in SNR needed) when the traffic patterns are multicast (messages need to reach multiple destinations). We consider this scenario in detail in Section. 3.6.1. Since XOR-CoW protocol is a modification of Occupy CoW, the effects of having non-simultaneous relaying (as seen in Sec. 3.3.4), increasing the number of destinations (as seen in Fig. 3.8) and duty-cycling (as seen in Sec. 3.4.5) are similar and hence omitted. We only briefly discuss the issue of optimizing phase to show the similarility in effects and to emphasize choosing a simple design over minute tweaking which could be cumbersome to implement.

3.6.1 XOR-CoW for Generic Information Topology

The XOR-CoW scheme for a generic information topology can be summarized as follows. All nodes know the information topology – the origin and destinations of the messages. Therefore, all nodes know which messages can be XORed. The schedule of messages $\mathcal{G}$ are determined and all nodes know the schedule. For the first phase, the schedule is simple: each message stream is allocated one slot. However, in the second phase (XOR phase), the schedule $\mathcal{G}_X$ is different: whenever bi-directional traffic exists in the information topology, allocate one slot for those two messages in $\mathcal{G}_X$, else allocate one slot for that single message in $\mathcal{G}_X$ (as shown in Fig. 3.14a). In the first phase, nodes take turn according to the schedule to transmit the messages. All nodes listen when they are not transmitting. In the XOR phase, all nodes that can transmit a message (or an XORed message) transmit according to the XOR phase schedule *simultaneously* using a DSTC. In the following section, we



(a) Schedule during first phase and the XOR phase for a generic topology. Pairs of message streams that are inherently bi-directional i.e, $(C_1 - C_2, C_2 - C_1)$ and $(C_2 - S_2, S_2 - C_2)$ are the only ones that are XORed (shown in teal colored boxes).

(b) Fixed and flexible scheduling for the star topology example considered in Fig. 3.15. The green boxes correspond to the downlink packets from the controller to the client nodes (the destinations are labelled: S_i). The pink boxes correspond to the uplink packets from the client nodes to the controller (the origins are labelled). The purple boxes correspond to the XOR packets where the label corresponds to client node whose DL and UL packets are XORed.

Figure 3.14: Scheduling for generic and star topology

focus on the star topology as network coding yields maximum benefits when the traffic is bi-directional [169, 170].

3.6.2 XOR-CoW for Bi-directional Information Topology

In this section, we consider bi-directional topologies wherein if a node A has information for node B, then node B also has information for node A. A simple case of bi-directional traffic is the star topology which we will consider here for exposition purposes. A centralized control system can be modeled as a star topology where the network consists of a central controller C and n client nodes. In each 'cycle' of time T, the controller has m distinct bits of message for *each* client node (downlink messages - DL) and each client node has m distinct bits of message for the controller (uplink messages - UL). As in [42], we assume that while normally, the controller and all the nodes are in-range of each other, bad fading events can cause transmissions to fail. Successful nodes, namely those that have received both the downlink message from the controller and the uplink message for a client node in need, XOR the uplink and downlink messages together to form a single packet. They then broadcast the XORed packet simultaneously. The controller uses the XORed packet as well as the downlink information that it already has to decode the uplink packet. The destination node uses the XORed packet as well as the uplink information that it already has to decode its

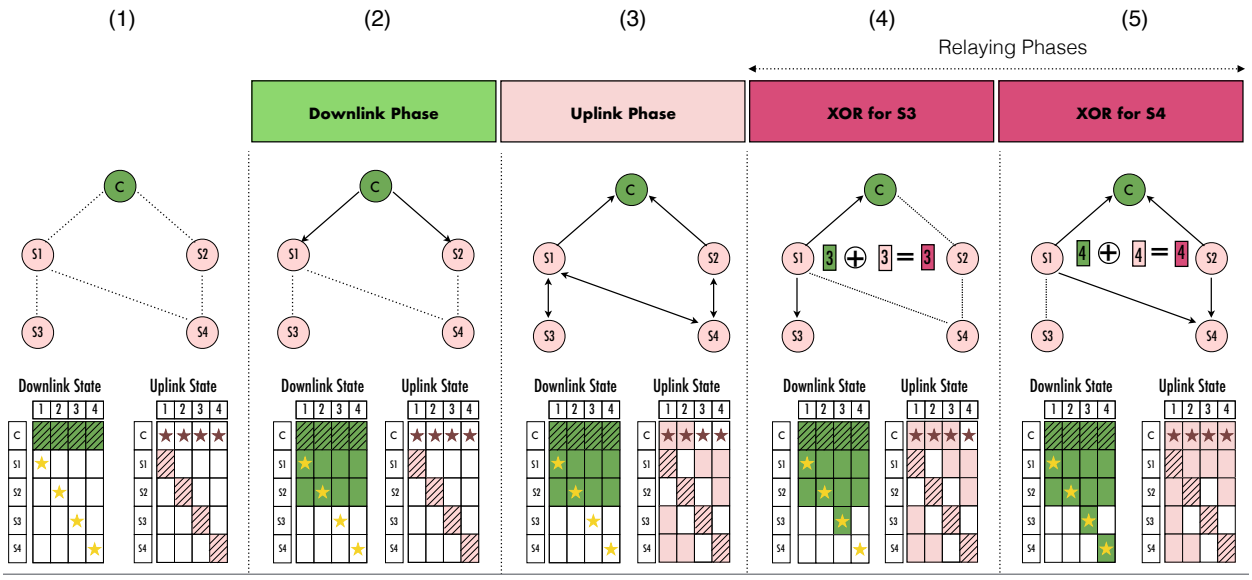


Figure 3.15: Simple example of XOR-CoW with one controller and 4 nodes. The graph illustrates which links are active during that phase. The downlink and uplink tables at each stage represent the information each node has at the end of that phase. Striped cells indicate message origins and starred cells indicate message destinations.

downlink packet.

This scheme has three phases: downlink phase, followed by uplink phase and then the XOR phase. Let the time allocated for the downlink phase be T_D, the uplink phase be T_U and the XOR phase be T_X such that $T_D + T_U + T_X = T$. We will describe the protocol with the aid of Fig. 3.15 where the network consists of one controller and 4 nodes (S1 - S4). To the left of the figure are the downlink buffers at each node (controller and clients) and to the right of the figure are the uplink buffers, also at each node. They get populated as messages are decoded. Initially, the controller's downlink buffer is full as it is the origin of all downlink messages (shown by the striped buffers) and its uplink buffer is empty. S1 - S4 start with their corresponding uplink buffer being full (shown by the striped buffers) and their downlink buffers are empty. The starred messages are those that each user is interested in receiving. The controller is interested in the uplink messages of nodes and the nodes are interested in receiving the specific downlink message intended for them.

Schedules:

There are two versions of the XOR-CoW protocol that can be employed: a) fixed schedule protocol and b) flexible schedule protocol. The difference between these two mainly lies in the relaying phase – do all nodes get another shot at getting their message across or *only* those in need? This is illustrated in Fig. 3.14b.

1) Fixed schedule: In this scheme, time is allocated equally for all nodes in the XOR phase – such that they get another shot at sending their messages. Since the schedule is predetermined, the time at which the message of a particular node is to be transmitted is also known to all users and there is no real need for a scheduling phase to determine the schedule for the XOR phase.

2) Flexible schedule: In this scheme, time is allocated equally only for the nodes which need help in the XOR phase (and no time is given for the messages that have already reached the destination). This scheme requires a scheduling phase since the relays need to be told about the nodes that need help.
Keeping these schemes in mind, we describe the protocol under these schemes.

3.6.3 Downlink and Uplink Phases

During these phases, all the nodes are listening whenever they are not transmitting. The downlink phase is common in both the fixed and flexible scheduling schemes. The cycle starts with a downlink phase in which the controller broadcasts a single packet consisting of all m-bit messages to all n nodes at rate $R_D = \frac{m \cdot n}{T_D}$. In Fig 3.15 column 2, S1 and S2 successfully decode the entire downlink message. Their starred buffers are filled along with the downlink buffers corresponding to other nodes.

Fixed Schedule Scheme: This is followed by the uplink phase, in which the individual nodes transmit their messages to the controller one by one according to a predetermined schedule at rate $R_U = \frac{m}{T_U/n} = \frac{m \cdot n}{T_U}$ by evenly dividing the time slots among all nodes. In Fig 3.15 column 3, the controller successfully decodes the uplink messages of S1 and S2 and the starred uplink buffers of the controller corresponding to these nodes are filled. Since all nodes are listening whenever they are not transmitting, S1 receives the uplink messages of S3 and S4 while S2 receives the uplink message of S4. The nodes which have successfully received the downlink message as well as successfully transmitted their uplink message to the controller are referred to as **strong nodes**. In Fig. 3.15, S1 and S2 are the strong nodes.

Flexible Schedule Scheme: In the uplink phase of the flexible scheduling scheme, the nodes also transmit a one bit ACK to the controller (indicating whether they've successfully received the downlink packet or not). Therefore, the individual nodes transmit their messages (including one bit for an ACK) to the controller one by one according to a predetermined schedule at rate $R_U = \frac{m+1}{T_U/n} = \frac{(m+1) \cdot n}{T_U}$ by evenly dividing the time slots among all nodes.

3.6.4 Scheduling Phase

This phase is *crucial* when the flexible scheduling scheme is employed. In this phase the controller transmits acknowledgments to the strong nodes (at the same rate as the downlink

phase). This is just 2 bits of information per node for downlink and uplink. The common-information about the system's state enables the strong nodes to share a common schedule for relaying messages for the remaining nodes. Note that the schedule only reaches the strong nodes but the nodes which need help do not know the schedule. How will they know which message is intended for them without the knowledge of the schedule? This can be addressed by building in identification of the destination node in the packet such that the nodes can figure out which packet was addressed to them while keeping the transmission rate the same. This approach has been discussed in detail in [171]. Therefore we assume that the nodes know which packet was meant for them.

3.6.5 XOR phase:

Depending on the scheduling scheme, the time allocated for this phase can either be equally divided among all nodes – corresponding to the rate of transmission is $R_X = \frac{m \cdot n}{T_X}$, or only those that need help – corresponding to the rate of transmission is $R_X = \frac{m \cdot n_1}{T_X}$ where n_1 are the number of unsuccessful nodes. In either case, the strong nodes XOR the downlink and uplink messages of each of the unsuccessful nodes they've heard. During the slot of an unsuccessful node (say node Y), all the strong nodes that have successfully heard node Y act as simultaneous broadcast relays and transmit the XORed packet using a DSTC.

In Fig. 3.15, S3 and S4 are the unsuccessful nodes. In the XOR slot allocated for S3 (Fig. 3.15 column 3), S1 XORs the downlink and uplink packet of S3 (represented by the purple packet) and broadcasts it. Using the downlink packet of S3, the controller can now recover the uplink packet. Using its own uplink packet, S3 can now recover the downlink packet. The process for S4 is similar and the difference lies in the fact that S1 and S2 simultaneously transmit the XORed packet for S4.

3.7 Analysis of XOR-CoW

In this section, we analyze the performance of XOR-CoW. The performance of XOR-CoW's performance for a generic information topology is the same as the performance of Occupy CoW for a generic topology. The behavioral assumptions are the same as the ones made for Occupy CoW in Sec. 3.3.

3.7.1 XOR-CoW probability of failure

The complete analysis of the performance of the XOR-CoW protocol is described in Appendix B. In this section we mainly present the results and state two theorems which are useful in understanding the results.

Theorem 1. *If an instance of fixed schedule two-hop Occupy CoW protocol (i.e., no rate adaptation in the relaying phases) with equal downlink and uplink phases ($T_{D_1} = T_{U_1} =$*

$T_{D_2} = T_{U_2} = T_M$) succeeds, then there is a common downlink and uplink success path for each node in the network.

Proof. If a node successfully decoded the downlink message in one hop, its uplink message also gets through successfully to the controller in one hop (due to channel reciprocity). If a node successfully decoded the downlink message in two hops via a relay Z, then the same relay helps uplink as well – again due to channel reciprocity. □

Theorem 2. *If an instance of fixed schedule two-hop Occupy CoW protocol with equal downlink and uplink phase 1 ($T_{D_1} = T_{U_1} = T_{D_2} = T_{U_2} = T_M$) and a given SNR succeeds, then the fixed scheduling version of XOR-CoW with downlink and uplink phase lengths both equal to T_M and XOR phase length also equals to T_M succeeds at the same SNR.*

Proof. From Theorem 1 we know that the paths for downlink and uplink success when $T_{D_1} = T_{U_1} = T_{D_2} = T_{U_2} = T_M$ are the same – i.e., either they directly succeed to the controller or they have the same relay helping in both downlink and uplink. These relays essentially have the capability the XOR the packets as they have both the packets as well as good links for transmission. Hence, as long as the rate in the XOR phase stays the same (this is ensured by $T_D = T_U = T_X = T_M$), the XOR-CoW protocol also succeeds at the same SNR. □

A corollary of Theorem 2 is that while two-hop Occupy CoW would require time $4 \times T_M$ to succeed, XOR-CoW succeeds in time $3 \times T_M$ – i.e., a throughput improvement of $\frac{4}{3}$.

3.7.2 Results and comparison

We explore the performance of XOR-CoW with parameters taken from a contemporary practical application, the industrial printer case described in [6]. The SERCOS III protocol [39] supports the printer's cycle time of 2 ms with system error probability of 10^{-8}. We target the following system requirements for the application: 30 moving printing heads that move at speeds up to 3 m/s over distances of up to 10 m. Every cycle lasts 2ms and in each cycle the controller transmits 20 bytes of actuation data to each head and each of the 30 sensors transmit 20 bytes of sensory data to the controller. Assuming access to a single 20MHz wireless channel, this 4.8 Mbit/sec throughput corresponds to an overall spectral efficiency of approximately 0.25 bits/sec/Hz. SERCOS supports a reliability of 10^{-8} and for our protocol we target a reliability of 10^{-9}.

We define the cycle failure probability as the probability that any packet transmitted during the cycle did not reach at least one of its destinations. We use the minimum SNR required to achieve 10^{-9} reliability as our metric to compare XOR-CoW to other schemes. Fig. 3.16 compares the performance of the following protocols a) XOR-CoW, b) Occupy-CoW (the cooperative-communication-based protocol not employing network coding), and c) Frequency hopping based protocols. We see that optimized version of Occupy CoW (the best performance that can be obtained without using network coding) and XOR-CoW

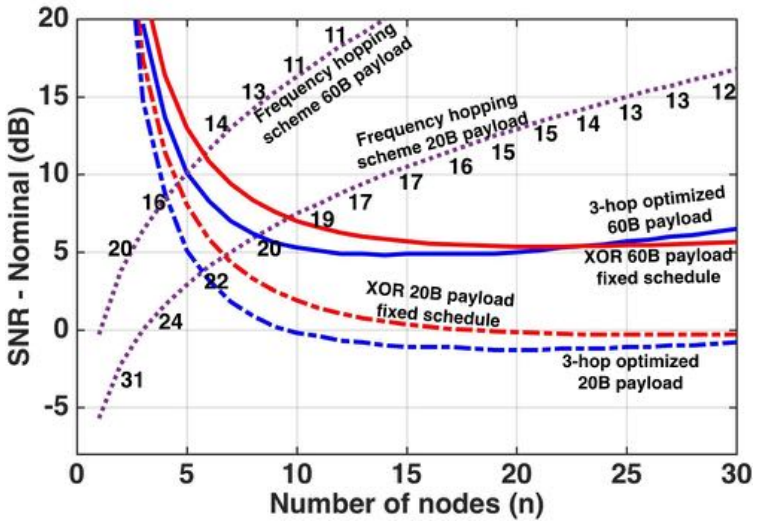
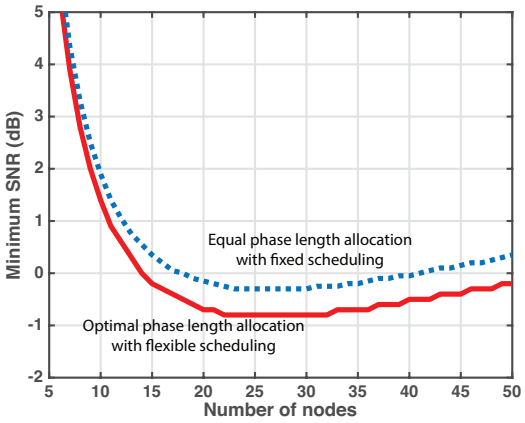
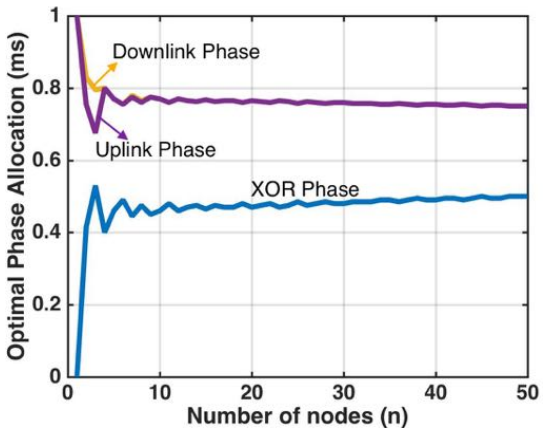


Figure 3.16: The performance of XOR-CoW for a star information topology compared with reference schemes for varying network size, and a 2ms cycle time, aiming at 10^{-9} probability of failure for a 20MHz channel. The numbers next to the frequency-hopping scheme show the frequency diversity needed and those next to the non-simultaneous retransmission scheme show the optimal number of relays per message stream.

with a simple equal-time allocation to different phases perform comparably for $m = 160$ bits (the dot-dashed lines). The advantage of XOR-CoW is clear for high aggregate rates and large networks as shown by the solid in Fig. 3.16. We see that XOR-CoW beats the performance of Occupy CoW for $m = 480$ bits and network size > 20 while also being a simpler scheme. The dotted purple curves represent a hypothetical (non-adaptive) frequency-hopping scheme that divides the bandwidth $W = 20$MHz into k sub-channels that are assumed to be independently faded, for $m = 160$ bits and $m = 480$ bits. The curves are annotated with the optimal k. As k (and thus frequency hops) increases, the available diversity increases, but the added message repetitions force each link's instantaneous data rate to be higher. For low n the scheme prefers more frequency hops to exploit diversity benefits. The SNR cost of doing this is marginal because the throughput is low enough that we are still in the linear-regime of channel capacity. For networks with fewer than 7 nodes, this says that using frequency-hopping is great — as long as we can reliably count on about 20 independently faded sub-channels to repeat across, which is not always practical. We will revisit this in detail in Sec. 3.3.



(a) SNR comparison of optimized flexile-schedule XOR-CoW and fixed-schedule XOR-CoW for $m = 160$ bit and varying network size with 20MHz bandwidth and a 2ms cycle time, aiming at 10^{-9}.



(b) The phase allocation for optimized XOR-CoW with flexible scheduling for $m = 160$ bit messages and varying network size with 20MHz bandwidth and a 2ms cycle time, aiming at 10^{-9} is shown.

Figure 3.17: Optimization of XOR-CoW protocol

3.7.3 Optimization

3.7.3.1 Network Coding Optimization

XOR-CoW scheme only allows for the opportunity to XOR two packets and not more. Are we making sub-optimal decisions by restricting to XORing only two packets? We are not and the reason is as follows. In undirected network (wireless networks considered here can be modeled as undirected networks) the throughput improvement that network coding provides when compared to routing only schemes is upper bounded at 2 [172]. We showed in Sec. 3.3 that the throughput improvement for the best case i.e., the star-topology is actually $\frac{4}{3} < 2$.

Furthermore, we can model the generic information topology as a multicast session. It has been shown that asymptotically network coding provides no benefits when compared to a pure routing schemes [173]. Additionally, even if we end up with a network realization which can provide significant network coding benefits (a rare event in itself), the coding points (which perform network coding operation) need to know the state of each packet and the network realization to compute the optimium code. The overhead of acquiring this network information state is significant (similar in spirit to why backpressure routing isn't implemented as-is in current networks).

3.7.3.2 Phase Length Optimization

We consider the XOR-CoW protocol and look at the optimal allocation of time which minimizes the SNR required to meet the performance specifications. Although the phase length

allocations are uneven (as seen in the figure 3.17b), we find that the SNR saving that we achieve by having different lengths is minimal (as seen in the figure 3.17a). The complexity of building a system which can operate at variable rates is extremely difficult and ultimately negates out the small SNR savings achieved by optimization. The strength of the protocol lies in the fact that a simple scheme with equal time allocations with fixed schedule performs almost as good as the optimal scheme – thus paving the way for a practical system.

3.8 Conclusion

This chapter introduces two wireless communication protocol frameworks – Occupy CoW and XOR-CoW (that improves upon Occupy CoW by leveraging opportunities for network coding) for high-performance industrial-automation systems that demand ultra-high reliability and low-latency for many message streams within a network with many active nodes. The protocol frameworks are targeted to a single wireless local domain where all nodes are nominally in range of each other, but can handle any arbitrary information topology in terms of which node is subscribed to which message stream. Harnessing significant diversity is absolutely essential for ultra-reliability and cooperative communication using relaying can access multiuser diversity. To achieve low-latency, simultaneous transmission using a diversity-oriented distributed space-time code is important, especially when the payload sizes are such that spectral efficiency is a concern. This gives a significant SNR advantage over pure frequency-hopping approaches while also not demanding that nature guarantee a lot of frequency diversity. Time diversity is also not viable when the tolerable latency is shorter than the coherence time, leaving multiuser diversity as the only real choice to combat fading.

When the background power used for having the wireless subsystem turned on is significant, it is beneficial to have subsets of nodes go to sleep while relying on others to listen and relay messages. Although this increases the transmit power required, it reduces overall network power consumption. Using network coding in addition to the above techniques does give significant savings and might additionally save on energy too. Simple phase length allocations and a fixed schedule are suffice to achieve our target reliability are reasonable SNR; optimized scheduling and phase lengths only provide marginal savings.

Although the main focus of this chapter has been fading and consequently we use spatial diversity techniques, we acknowledge that there are other events that may cause transmissions to fail. The final design that we propose includes repetitions in time and frequency to guard against *unmodeled errors* such as mis-synchronized clocks or stray interference events.

Chapter 4

Wireless Channel Dynamics for URLLC

4.1 Introduction

The biggest hurdle in enabling ultra-reliable low-latency wireless communication is fading. To mitigate the effects of fading, diversity techniques exploiting time, frequency and space have been used successfully to build in reliability. In Chapter 3, we studied two protocols – Occupy CoW and XOR-CoW that exploits spatial diversity techniques to combat fading. However, we made some assumptions about wireless channels and resources. Most of them can be justified and can be accounted for easily. However, are we aim for ultra-reliability, we cannot ignore the assumptions made about wireless channels. In this chapter we fundamentally ask two questions about wireless channels:

- How correlated are channel fades in space? What happens to the performance of protocols that depend on cooperative communication in the presence of spatial correlation?
- How does a channel quality change temporally? Does a channel quality truly remain constant for the coherence time of the system? If it doesn't how fast is the change and can the changes be predicted?

Although spatial-diversity-based techniques seem to address the needs of URLLC, we need to take an in-depth look at wireless channel dynamics specifically in the low-latency regime. It is imperative to understand the events that may occur in timescales ranging from tens of microseconds (corresponding to the length of a single short packet) to a few milliseconds (corresponding to the cycle time). A knowledge of these events helps focus on issues that might otherwise be overlooked if a traditional quasi-static-channel model was considered. This in turn aids in designing systems that can sufficiently guard against potentially adverse and rare events. To illustrate the above point, let us consider the following scenario. There is a network of nodes that wants to send messages to their destinations wirelessly. To guard against bad fading events, nodes nominate a buddy node to relay their

message. Thus, each message gets two shots at getting to its destination – first, directly from the source to the destination if the corresponding channel is good and second, through the buddy node (relay) if there were good channels between the source and the relay and the relay and the destination. The source nominates the relay based on some past channel state information but the buddy node relays the information at *a future time* – say a millisecond later. The traditional quasi-static-channel model suggests that channels remain static for a period of time given by the coherence time (which depends also on the system dynamics). Let us say that the traditional coherence time was 2ms. Given that the overall failure rate demanded is around 10^{-9}, can the source be one-in-a-billion confident that 1ms from now, that the relay will have a good channel to the destination? Do we trust the model that much?

Schemes that rely on the channel state information to maximize the data rate or choose 'leader' nodes have been considered in the context of URLLC. In [59], the authors consider a centralized control system where the controller performs pilot-assisted channel estimation to adapt the transmission rate to each device based on the quality of its channel. In [60], the authors consider a leader selection scheme based on channel state information. Both of these studies assume that the channels remain static for the duration of a cycle and have promising results. However, if channels could change much more rapidly than what the traditional coherence time would suggest, it could lead to an unmodeled degradation in performance for either scheme.

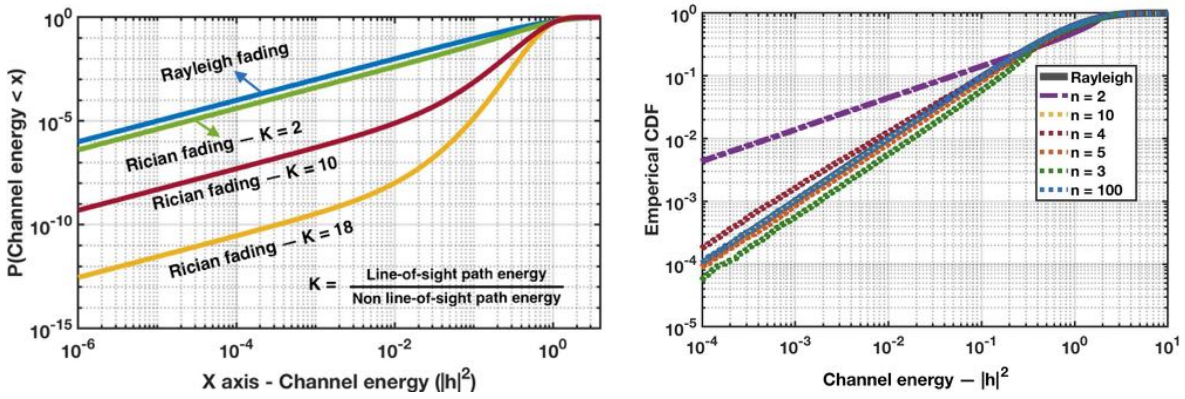
To resolve these questions, in this chapter we take a critical look at the main characteristics of wireless channels that impact the design and performance of different communication schemes. We do this by looking at both the nominal model of fading processes as well as identifying key dimensions of uncertainty to capture the impact of unmodeled effects. We then specifically analyze the impact of these channel characteristics on the performance of two cooperative communication schemes ('Occupy CoW' and 'XOR-CoW') to illustrate the key pain points and reveal where margin needs to be added to the schemes to be able to absorb the effect of both nominal and unmodeled uncertainty. The rest of the chapter is organized as follows:

- In Sec. 4.2 we study the temporal and spatial characteristics of wireless channels. In Sec. 4.5 we study wireless channel dynamics at two main time scales.
  - We study the channel dynamics on the order of tens to hundreds of microseconds which corresponds to the time duration of a short packet. We find that for short packets of duration under 100μs (motion under 0.01λ), channels that are good enough stay quite static. There is no large variation in channel energy within this time if the channels started out to be good. On the other hand, if channels were deeply faded to begin with, even minute changes in energy would manifest as large relative changes but really has no impact on the performance as those were bad channels anyways.

  - We study the channel dynamics on the order of hundreds of microseconds to a few milliseconds which corresponds to the cycle time as well as relaying events. We find that Rayleigh fading processes are not bandlimited. This has significant implications in channel quality prediction and relay selection techniques, which we will explore more in Chapter 6.

- In Sec. 4.6 we study the spatial correlation of wireless channels and understand its impact on the fading distribution. We find that under reasonable conditions, we get fading distributions that are not too far off from an independent spatial fading model. In fact, a channel correlation bounded by 0.2 (corresponding to nodes that are all at least 3λ apart as Eq. 4.6 suggests) can easily be modeled as a drop in nominal SNR of just 0.05dB.

4.2 Wireless Channel Modeling



(a) The CDF of channel energy $|h|^2$ for Rayleigh (no line-of-sight path) and Rician (with line-of-sight path) faded channels for the same nominal SNR. The CDF for Rayleigh faded channels has more mass around 0 (indicating deep fades) than Rician faded channels.

(b) The empirical CDF of channel energy $|h|^2$ for a Rayleigh faded channel obtained through the Jakes's model as given by Eq. (4.1). For different values of number of scatterers n the empirical CDF lingers around the nominal CDF except at $n = 2$.

Figure 4.1: Empirical and theoretical CDF of Rayleigh and Rician fades.

The main characteristics of wireless channels that are of interest in the context of URLLC are the joint distributions of the channel fades across time and space. As mentioned earlier in Sec.2.7, most works in the literature have focused primarily on large-scale statistics, on the order of tens of milliseconds to seconds. The fading process is assumed to be band-limited if the motions have bounded speed [139]. Moreover, often channels are assumed to be static

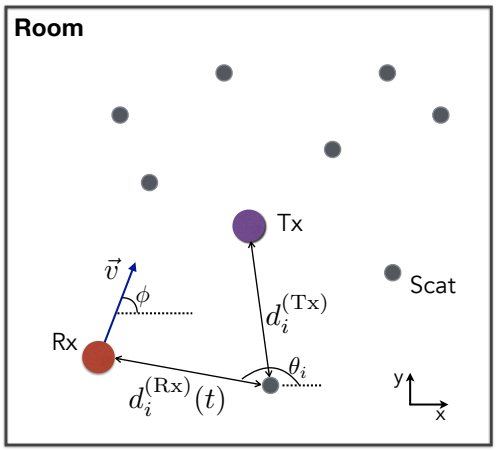


Figure 4.2: Room setup with n static scatterers, a static transmitter and a mobile receiver.

for a duration of time given by the coherence time which is determined by how fast the nodes are moving etc. However, it is imperative to question these assumptions and revisit these concepts keeping in mind that we are operating in low-latency regimes with short packets and aim to deliver high reliability. Knowing which events can lead to failure (such as picking a relay assuming a static channel that actually ends up changing) and effectively addressing them by considering channel dynamics is key to building robust wireless communication systems for URLLC.

In this chapter, we focus on studying Rayleigh fades to understand the worst-case scenario where there is no line-of-sight path in indoor environments. A line-of-sight path makes the fade distribution better (like Rician) as shown in Fig. 4.1a where we see that the mass around 0 is higher in Rayleigh fading than in Rician fading – i.e., the chance of a Rayleigh faded channel being in a deep fade is higher than a Rician faded channel.

Rayleigh faded channels have traditionally been modeled using a sum-of-sinusoids like in Jakes's model [139]. We revisit the classical Jakes's model dynamics of Rayleigh faded channels and question whether the process is fundamentally bandlimited if the motions have bounded speed. We consider only the effects of multipath as we focus on the *variations* at small timescales. The effects of shadowing, diffraction, and other propagation effects can be divided into two categories. One is nominal — when we think about the nominal SNR in this chapter, this is the minimal SNR including all regions of shadowing and normal path loss. Fundamentally, we are interested in modeling the situation where every node can nominally hear every other node — if there were true "dead spots" where shadowing prevents this, this would presumably be known ahead of time and would need to be dealt with using "range-extension" techniques. The other aspect of shadowing and diffraction, namely transitions into and out of shadows, we fold into the unmodeled part of the channel that is addressed in Chapter 5.

Consider a two-dimensional room with n static scatterers distributed uniformly at ran-

dom[1]. Let there be a static single-antenna transmitter in the middle of the room and a single-antenna mobile receiver moving at a constant speed v in some random direction inside the room (illustrated in Fig. 4.2). Let the transmitter be transmitting a tone at frequency f_c (wavelength λ_c). The channel coefficient between the transmitter and the receiver at any time t is given by

$$h(t) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \exp\left(j \frac{2\pi (d_i^{(\mathrm{Rx})}(t) + d_i^{(\mathrm{Tx})})}{\lambda_c} \right) \tag{4.1}$$

where $d_i^{(\mathrm{Rx})}(t)$ is the distance of the scatterer i from the receiver at time t and $d_i^{(\mathrm{Tx})}$ is the distance of the scatter i from the transmitter (both are assumed to not be moving for simplicity). The $1/\sqrt{n}$ normalization in Eq. (4.1) is to keep the marginal variance the same across different numbers n of scatterers. This is because our goal is to understand the reliability impacts of the variability that fading brings — this tells us how much higher we need to make the nominal SNR to be able to absorb the impact of these fades without losing system-level reliability. Eq. (4.1) follows from the results in [174].

4.3 Channel variation as a Gaussian process

We want to understand how channels between a pair of antennas vary as one (or both) antennas move while the environment (scatterers) remains largely stationary. This model captures the small-scale variations that we are interested in, where the nodes are moving at a reasonable speed but for small amounts of time (in ms). The channel coefficient at any point in time is marginally distributed as a complex normal (as the CLT suggests for the expression in Eq. (4.1)), and the channel coefficient process through time can be modeled as a Gaussian process. The parameters that we need to define the Gaussian process are the means and the covariance functions which depend on the distance that the receiver has moved. We assume that the velocity $\vec{v}$ of the receiver is constant over the time durations of interest such that the position of the receiver $\vec{s}(t)$ at time t is given by

$$\vec{s}(t) = \vec{s_0} + \vec{v}t = (x_0 + vt\cos\phi, y_0 + vt\sin\phi) \tag{4.2}$$

where $\vec{s_0} = (x_0, y_0)$ is the initial position of the receiver at $t = 0$ (uniformly distributed in the room), ϕ is the angle of motion of the receiver with respect to the x-axis (uniformly distributed over $[0, 2\pi)$). Let the position of scatterer i be given by $\vec{s_i} = (x_i, y_i)$. The distance

[1] In Fig. 4.1b, we also see that the convergence is very fast with the number n of scatterers. Essentially, it has converged by $n = 3$. However, $n = 2$ needs to be considered as a special case if the deployment environment is one in which that could happen. We do not dwell on that case here, but in effect, its CDF tells us that we need to pay an extra 10+dB of transmit power if we want to avoid deep fades in the context of two-scatterer environments. This is because the case of exactly two scatters results in nulls along long valleys where the two paths have canceling phases. Once we have three or more paths, valleys of nulls can no longer exist — instead, we get isolated nulls. This geometric fact tells us why the convergence is so fast.

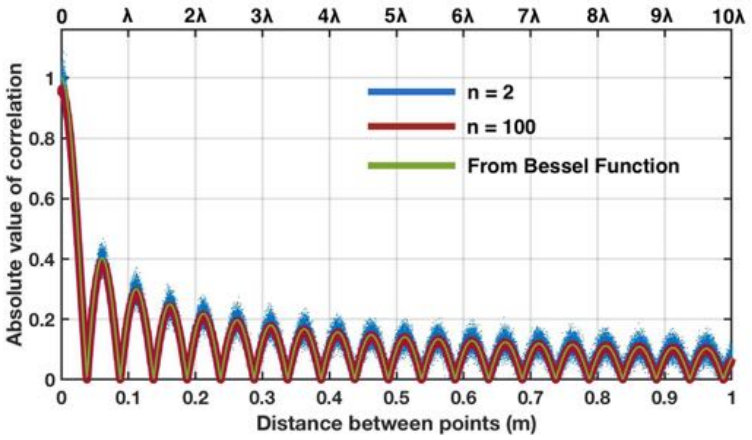


Figure 4.3: Simulated absolute value of the cross-covariance of the fading process for $f_c =$ 3GHz. The curves are exactly as predicted by Eq. 4.6 as the red curve corresponding to $n = 100$ aligns exactly with the green curve which corresponds to the actual Bessel function.

of the receiver from scatterer i at time t is given by

$$\begin{aligned} d_i^{(\mathrm{Rx})}(t) &= \|\vec{s}(t) - \vec{s_i}\| \\ &= \sqrt{(x_0 + vt\cos\phi - x_i)^2 + (y_0 + vt\sin\phi - y_i)^2} \\ &= \sqrt{d_i^{(\mathrm{Rx})}(0)^2 + (vt)^2 + 2vtd_i^{(\mathrm{Rx})}(0)\cos(\theta_i - \phi)} \end{aligned} \tag{4.3}$$

where $d_i^{(\mathrm{Rx})}(0)$ is the distance of the receiver from the scatterer i at $t = 0$ and θ_i is the angle made by the line joining the scatterer and the receiver at time $t = 0$ which is independent of ϕ.

As we are interested in channel dynamics and correlations, it is natural to examine the covariance of the in-phase, $\Re(h(t))$ and the quadrature components, $\Im(h(t))$ of the wireless channel as a function of speed v and time t. Let us denote this covariance by $k(v,t) = \mathbb{E}[\Re(h(t))\Re(h(0))] = \mathbb{E}[\Im(h(t))\Im(h(0))]$. Let us also look at the cross-covariance of the channel coefficient $(h(t))$ given by $\tilde{k}(v,t) = \mathbb{E}[h(t)h^*(0)]$. Since the in-phase and quadrature components are uncorrelated (verified through simulations), $\tilde{k}(v,t) = \mathbb{E}[\Re(h(t))\Re(h(0))] + \mathbb{E}[\Im(h(t))\Im(h(0))] = 2k(v,t)$. We now calculate $k(v,t)$ through $\tilde{k}(v,t)$. We have,

$$\begin{aligned} \tilde{k}(v,t) &= \mathbb{E}[h(t)h^*(0)] \\ &= \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n}\exp\left(j\frac{2\pi}{\lambda_c}\left(d_i^{(\mathrm{Rx})}(t) - d_i^{(\mathrm{Rx})}(0)\right)\right)\right. \\ &\left. + \sum_{i\neq j}\exp\left(j\frac{2\pi}{\lambda_c}\left(d_i^{(\mathrm{Rx})}(t) - d_j^{(\mathrm{Rx})}(0) + d_i^{(\mathrm{Tx})} - d_j^{(\mathrm{Tx})}\right)\right)\right]. \end{aligned} \tag{4.4}$$

As $d_i^{(\mathrm{Rx})}(t)$ and $d_j^{(\mathrm{Rx})}(0)$ are independent for $i \neq j$ and the scatterers are distributed uniformly across the room, the expectation of the cross term in Eq. (4.4) is 0.

$$\tilde{k}(v,t) = \mathbb{E}\left[\exp\left(j\frac{2\pi}{\lambda_c}\left(d_i^{(\mathrm{Rx})}(t) - d_i^{(\mathrm{Rx})}(0)\right)\right)\right] \tag{4.5}$$

Substituting Eq. (4.3) in (4.5), for small movements ($\frac{vt}{d_i} \approx 0$), the covariance function is given by

$$\tilde{k}(v,t) = J_0\left(\frac{2\pi}{\lambda_c}vt\right) \tag{4.6}$$

where $J_0(\cdot)$ is the Bessel function of the first kind (also derived in [175]). Fig. 4.3 shows the simulated absolute value of the expected cross-covariance of the fading process as a function of distance which matches Eq. (4.6). The convergence is rapid with the number of scatterers n.

4.4 Bandwidth of fading processes

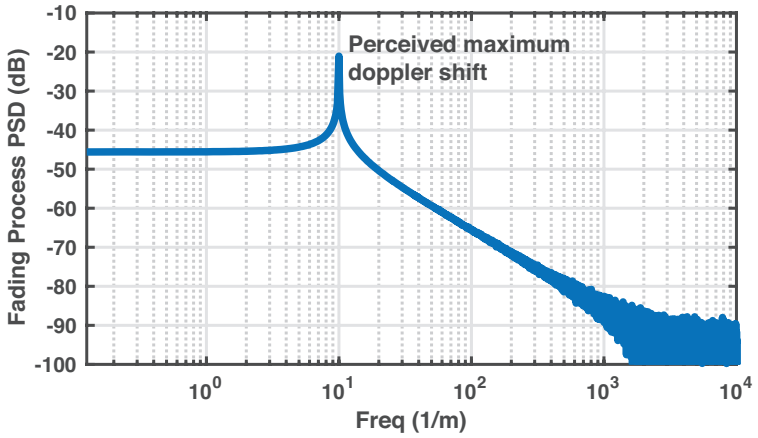
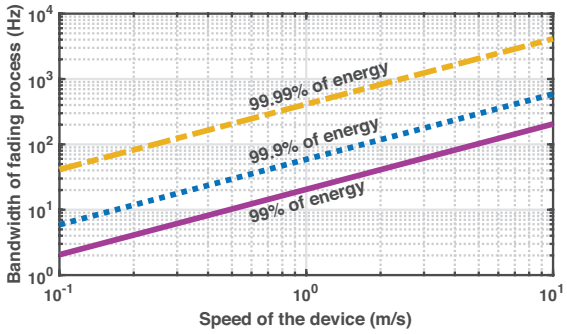


Figure 4.4: One-sided power-spectral density (PSD) of the fading process for $f_c = 3$GHz. Traditionally the PSD has considered to exist only until the perceived maximum Doppler shift but there clearly exists energy beyond this frequency and therefore the process can be considered to not be bandlimited.

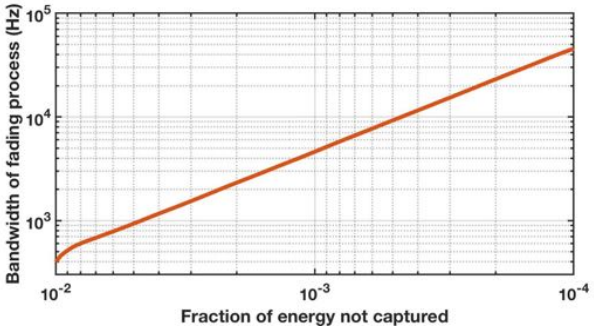
The power spectral density of the complex fading process has indeed been looked at in studies like [175, 139, 176]. However, they make an essential assumption: that the power spectrum is bowl shaped and the contribution of frequencies higher than the perceived maximum frequency is zero – essentially, the fading process is bandlimited. The unilateral Laplace transform of the Bessel function ($\mathcal{L}(J_0(x)) = 1/\sqrt{1+s^2}$) has poles on the imaginary axis. Therefore, the Fourier transform gets tricky – how do we deal with these poles? Studies so

far (such as [175, 139, 176]) seem to have elected to restrict the Fourier transform of the Bessel function until the 'maximum' Doppler shift (i.e., v/λ_c), possibly to address these poles. However, our simulations show that the fading process is not bandlimited – it has energy even beyond the traditionally assumed maximum Doppler shift. This surprising discovery was also supported by looking numerically directly at the Bessel function. The standard assumption of ignoring the response outside the maximum Doppler shift was reasonable when the focus was on the average or typical behavior of the process. However, for URLLC we are interested in rare events with probabilities on the order of 10^{-9} so taking this into account is important.

Figure. 4.4 plots the one-sided power spectral density (obtained through simulations) of the fading process, for center frequency $f_c = 3$GHz. We do see the bowl shape that is traditionally expected until the spatial frequency of 10/m (corresponding to the maximum Doppler frequency) but it clearly doesn't die down to 0 immediately beyond the maximum Doppler frequency, instead decaying at the rate of 20dB per decade.



(a) Bandwidth around the center frequency which captures 99%, 99.9% and 99.99% of the energy of the fading process as a function of node speed.



(b) Bandwidth around the center frequency which captures varying amount of energy of the fading process while the node speed is kept constant at 10m/s.

Figure 4.5: Relationship between the bandwidth of the fading process and the energy it contains.

Traditionally, the bandwidth of a process has been characterized through the amount of energy in it. Fig. 4.5a plots the bandwidth that contains 99%, 99.9% and 99.99% of the energy for various node speeds for center frequency $f_c = 3$GHz. We see the expected linear scaling with speed but we also see the increase in bandwidth with increasing energy content. Specifically, Fig. 4.5b plots the increase in bandwidth with increasing energy captured in it where we see that the 10dB/decade increase is consistent with the behavior in Fig. 4.4 (where there is a 20dB/decade drop and the factor of two comes because of the squaring effect).

4.5 Temporal characteristics of wireless channels

In the URLLC context, there are two time scales of interest – the duration of a single short packet and the overall cycle time. Therefore, we focus on the temporal characteristics at these two time scales to capture the variations within a packet duration and variations within a cycle.

4.5.1 Channel variations within a packet duration

As we are interested in short messages (payloads), the corresponding packet durations will also be small. For payloads of sizes 10s to 100s of bytes, the packet duration is at most 50μs long if we assume the data-rate is on the order of 20Mb/s. Let us consider that a node may move at a maximum speed of 10m/s. How does the channel energy ($|h|^2$) change over the duration of a packet in the above setup?

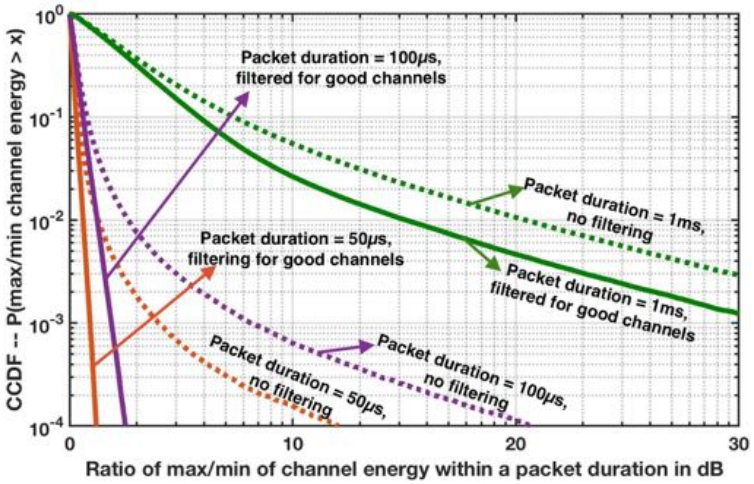


Figure 4.6: CCDF of the ratio of max channel energy by min channel energy in dB within a packet duration for various packet durations. The receiver is traveling at a speed of 10m/s and the center frequency is 3GHz. The dotted curves correspond to all channels and the solid curves correspond to those channels that are good at the beginning of the packet. For short packet sizes of 50μs, there is no discernible change in the channel energy when conditioned on the initial channel being good but as the packet duration increases, we see bigger variations become possible.

Fig. 4.6 studies the variation of channel energy within a packet for a static transmitter transmitting at center frequency of 3Ghz and a mobile receiver moving at speed 10m/s. The orange dotted line corresponding to packet duration of 50μs plots the CCDF of the ratio of maximum channel energy by the minimum channel energy within the given packet duration and the variation seems high. Does this mean that channel energy varies so wildly – the channel energy can fall by more than 10dB within a packet? The answer is: it depends. If

we condition to only look at channels that are good channels (energy above −7dB) at the beginning of the packet, then the variation is extremely minimal (less than 1dB) – as shown by the solid orange curve. This means that a good channel will remain reliably good for short packet sizes. The huge variations in channel energy were due to already badly faded channels – even small variations manifest in a big way.

However, the story is very different for medium to long packet sizes. We look at packets 100μs and 1ms long and see that even after conditioning on looking at the channels that started out good, there is a *significant* variation in their energy 1ms later (the effect is less pronounced for packet duration 100μs). The traditional coherence time for this setup is 2.5ms. However, Fig. 4.6 shows that even good channels do not reliably remain static for 1ms. This suggests that in the context of URLLC, having small packets (on the order of 10μs) can guarantee better stability. This also has significant implications for channel prediction and relay selection as studied in Chapter 6.

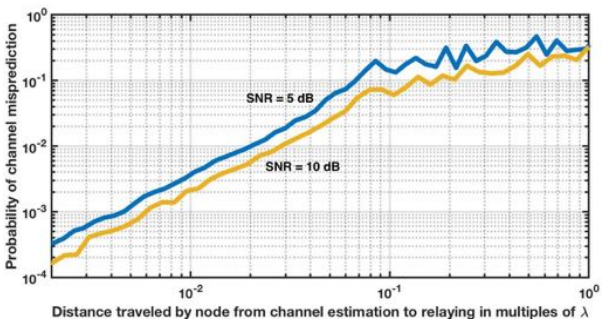
4.5.2 Channel variations within a cycle

We have looked at variation in channels for both small and large packet sizes and saw that for large packet sizes ≈ 1ms, the channel varies significantly. Therefore, simply assuming that channels remain constant for the traditional coherence time duration could potentially lead to severe degradation in performance. As described in Sec. 4.1, there are various schemes and techniques such as relay and leader selection and transmission rate optimization for which knowing the channel variation on the scale of milliseconds is critical. In this section, we study the variations of channels on the order of milliseconds and specifically look at the predictability of channels as a key measure in the context of URLLC.

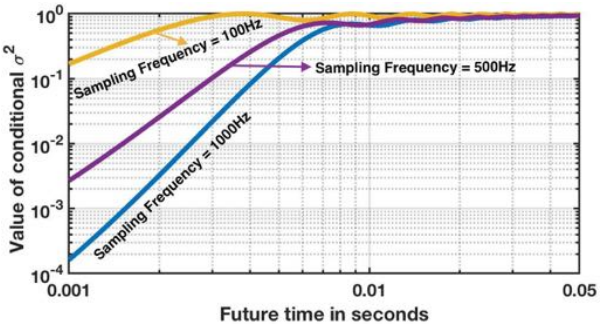
Consider the channel between a pair of nodes, say a source and a destination. Via feedback and/or reciprocity, suppose the source has knowledge of past channels given by $\vec{h} = \begin{bmatrix} h_1 & h_2 & \dots & h_m \end{bmatrix}^T$ from times $\vec{t} = \begin{bmatrix} t_1 & t_2 & \dots & t_m \end{bmatrix}^T$. We want to find the distribution of h_{m+1} at time t_{m+1} conditioned on $\vec{t}$ and $\vec{h}$. We assume that the channel coefficient variation is a Gaussian process, and use simple linear estimation. Therefore, $\{\vec{h}, h_{m+1}\}$ form a multivariate normal and the distribution of h_{m+1} conditioned on $\vec{h}$ is a complex normal distribution. Let

$$\mathbf{K} = \begin{bmatrix} k(v, t_1 - t_1) & k(v, t_2 - t_1) & \dots & k(v, t_m - t_1) \\ k(v, t_2 - t_1) & k(v, t_2 - t_2) & \dots & k(v, t_m - t_2) \\ \vdots & \vdots & \ddots & \vdots \\ k(v, t_m - t_1) & k(v, t_m - t_2) & \dots & k(v, t_m - t_m) \end{bmatrix}$$

be the covariance matrix of the in-phase and quadrature components corresponding to the times of observations so far. Let $\mathbf{K}_* = \begin{bmatrix} k(v, t_{m+1} - t_1) & \dots & k(v, t_{m+1} - t_m) \end{bmatrix}$ be the covariance matrix of the in-phase and quadrature components corresponding to the future time of interest and the times of observations so far. Also, let $\mathbf{K}_{**} = [k(v, t_{m+1} - t_{m+1})] = [k(v, 0)]$ be the variance of the in-phase and quadrature components. Let $\vec{h}_I = \text{Re}\{\vec{h}\}$ be the vector of



(a) The probability of channel quality misprediction as a function of how far the node has moved (at 10m/s) since the last channel measurement. This figure was obtained for predicting the channel using the past 3ms of channel measurements taken every millisecond, and nominal SNRs 5 and 10dB.



(b) The conditional variance of the channel energy distribution predicted by Eq. (4.8) as a function of future time and sampling frequency of the channel coefficient. The higher the sampling rate, the lower the variance. The farther out in the future, the closer the variance becomes to the unconditional variance.

Figure 4.7: The effects of channel changing within a cycle through the perspective of future channel characteristics for center frequency $f_c = 3$GHz.

the in-phase components of $\vec{h}$ and $\vec{h}_Q = \text{Im}\{\vec{h}\}$ be the vector of the quadrature components of $\vec{h}$. Then, the mean of the distribution of the in-phase μ_I and the quadrature component μ_Q of h_{m+1} conditioned on $\vec{t}$ and $\vec{h}$ is given by

$$\mu_I = \mathbf{K}_* \mathbf{K}^{-1} \vec{h}_I, \qquad \mu_Q = \mathbf{K}_* \mathbf{K}^{-1} \vec{h}_Q. \tag{4.7}$$

The conditional variance of both the in-phase and quadrature components is given by

$$\sigma_c^2 = \mathbf{K}_{**} - \mathbf{K}_* \mathbf{K}^{-1} \mathbf{K}_*{}^T. \tag{4.8}$$

As mentioned earlier, the goodness or the quality of a channel is captured by the energy ($|h|^2$) in the channel. The conditional distribution of the energy of the channel $|h_{m+1}|^2$ is given by

$$|h_{m+1}|^2 \sim \text{Rice}(\nu, \sigma_c) \tag{4.9}$$

where $\nu = \sqrt{\mu_I^2 + \mu_Q^2}$, σ is given by Eq. (4.8) and $\text{Rice}(\nu, \sigma_c)$ is the Rician distribution with parameters ν and σ_c. The value of σ_c^2 is a direct indicator of the variability of the channel at the future time t_{m+1}. In addition to the distance into the future, the value of σ_c^2 crucially depends on how fast we are sampling the channel as seen in Fig. 4.7b.

Redefining coherence time/distance:
We are interested in using the above model to understand the predictability of wireless channels. Traditionally, channels have been considered to be static for a period of time

or distance dictated by the system dynamics and carrier frequency. This is the notion of coherence time or distance. This has been a good rule-of-thumb for traditional cellular or WiFi-type systems as they focus mainly on average performance. However, URLLC requires guarantees on worst-case performance which challenges the traditional notion of coherence time or distance. As seen earlier in Sec. 4.5.2, channels simply do not stay static for the traditional coherence time. Essentially, given some past measurements of a channel, we can predict the channel energy in the future time or distance. If the channel energy is greater than some threshold (dictated by the nominal SNR), we label that to be a good channel. However, there is a probability of mislabelling the channel (similar in spirit to demodulation error) which crucially depends on the *future time or distance* through σ_c^2. This error in mislabelling the channel is the fidelity corresponding to the future time or distance. The nearer the future time is, the lower the misclassification probability would be and the farther out the future time is, the higher the misclassification probability would be. To this end, **we propose a more nuanced notion of coherence time or distance: the time or distance over which a channel is predictable to a given reliability.**

Fig. 4.7a shows the distribution of coherence distance in units of wavelengths for a single channel. This was obtained by considering predicting a channel distributed as Eq. (4.9) to be good or bad when operating at nominal SNRs of 5 and 10dB. We see that the prediction is incorrect about 0.2% of the time even when the node has moved only $\frac{1}{100}^{\text{th}}$ of the wavelength. The rule-of-thumb is that for every order of magnitude in distance, the probability of error goes up by about 1.5 orders. It plateaus around the unconditional outage probability. If a node travels sufficiently far, say $\frac{\lambda}{4}$ – it will have little channel correlation from where it began.

4.6 Spatial characteristics of channels

We have focused primarily on the temporal characteristics of wireless channels. As spatial-diversity-based protocols are promising candidates for enabling URLLC, it is essential to understand the spatial correlation of wireless channels. If we end up in a scenario where channels are heavily correlated, then spatial-diversity-based schemes may fail. Our investigations in Sec. 4.3 as well as experimental evidence show that **wireless channels are spatially correlated**. What does this mean for the schemes that want to exploit spatial diversity? Is this a recipe for disaster or is the degradation actually something manageable? The answer to this question depends on how far apart the nodes are. Essentially, there are two scenarios: one (unrealistic and impractical) where all nodes are within a wavelength apart from each other and the other (realistic and practical) where nodes are reasonably spread out in the environment.

Case 1: Nodes are clustered in a single region of radius less than a wavelength
Consider a centralized control system in which the controller has downlink information for

the users and the users have some uplink information for the controller and strict latency and reliability requirements are to be met. If all users are clustered in a single region of radius less than a wavelength, then all the channels to the controller are going to be highly correlated no matter where the controller is positioned. Therefore, this scenario will result in having an overall failure rate greater than the tolerable rate of 10^{-9} – if one node doesn't have a good channel to the controller, it is highly likely that other nodes also don't have good channels to the controller as the channels are very correlated. However, this is a worst case scenario where somehow all nodes land up in a tiny sphere and the *only way* to combat this would be to transmit at a very high power.

Case 2: Nodes are reasonably spread out in space

Again, the channel fades are going to be correlated. However, does this correlation mean that the realizations end up being much worse than if they are independently distributed? Surprisingly, the answer is no. Let us assume that we have the fade realization for the channel between the controller and a point $\vec{p}$ given by h_p and we want to know the distribution of the channel fade between the controller and another point $\vec{q}$. As these are jointly Gaussian, the channel fade between controller and point $\vec{q}$ is given by $h_q|h_p \sim \mathcal{CN}(\rho_{||\vec{p}-\vec{q}||}h_p, \sigma^2(1-\rho^2_{||\vec{p}-\vec{q}||}))$ where $\rho_{||\vec{p}-\vec{q}||}$ is the correlation of the fade distribution which depends on the distance between the two points $||\vec{p}-\vec{q}||$ (as given by Eq. (4.6)), and σ^2 is the unconditional variance of h_q (also of h_p).

Given that we are in the realm of "reasonably spread out in space", we assume that $|\rho_{||\vec{p}-\vec{q}||}| < 0.2$ (the distance between nodes is at least 3 times the wavelength as suggested by Fig. 4.3). In such cases, the conditional variance remains largely unchanged i.e., $\sigma^2(1-\rho^2_{||\vec{p}-\vec{q}||}) > 0.96 \cdot \sigma^2 \approx \sigma^2$. However, the conditional mean can still change significantly from being zero mean to something else. If the channel between the controller and point $\vec{p}$ is deeply faded i.e., $h_p \approx 0$, then the conditional mean of $h_q|h_p$ given by $\rho_{||\vec{p}-\vec{q}||}h_p$ is close to 0. In other words, **even if the channel h_p is deeply faded, then the channel $h_q|h_p$ is distributed approximately as $\mathcal{CN}(0,\sigma^2)$ as if it was independently faded but with slightly lower nominal SNR.** If the channel between the controller and point $\vec{p}$ is not in deep fade, then the conditional mean shifts away from zero but again, the variance remains unchanged. In other words, **if the channel between the controller and point $\vec{p}$ is not in deep fade, then it essentially biases the conditional distribution at $\vec{q}$ towards a good channel but with a slightly smaller variance.**

These findings suggest that the effect of spatial correlation is not necessarily horrible. In fact, a channel correlation of about 0.2 can easily be modeled as a drop in nominal SNR of 0.05dB. Consequently, if the nodes are reasonably spread out, spatial correlation can provide fade realizations that are almost as good as or in some cases better than spatial independence.

4.7 Conclusions and Future Work

In this chapter, we examined channel dynamics in the URLLC context. For the nominal model, we refined the standard Jakes' model-based view of Rayleigh fading and established that as long as the wireless nodes are always separated by a few wavelengths, the assumption of spatial independence essentially holds (with a 0.2dB penalty). Furthermore, we showed that although the traditional view of the fading process as being strictly bandlimited is false, the channel variation within a single short packet is very small once we condition on the channel being good to begin with. Across the entire low-latency cycle, the variation is more substantial and this can have significant implications on the predictability of channels.

In this chapter, we have explored channel variations in a theoretical setting with focus primarily on Rayleigh fading. Though we expect line-of-sight paths to improve channel quality substantially, their dynamics is worth exploring – especially if we care about predictability. Additionally, studying channel dynamics in indoor settings where these applications might be deployed, such as a factory floor or an operating theater is a must. We touch upon this a bit in Chapter 7 where we briefly describe our setting to understand channel dynamics and some initial results. Although briefly mentioned in this chapter, understanding the effects of shadow causing objects on the rate of change of channel coefficients is needed. These objects not only adversely affect the channel quality but can do so for a very long period of time (until this shadow lasts). A careful modeling of such objects and understanding their effects is needed. Another scenario that needs to be examined closely is that of $n = 2$ scatterers case. We briefly mentioned in this chapter that $n = 2$ is a special case and the probability of deep fade is higher than what traditional Rayleigh would suggest. However, because we care about ultra-reliability, it is crucial to understand this case better – maybe the channels are more deeply faded than Rayleigh but it could also be more predictable.

Indoor environments are not the only places applications requiring URLLC are envisioned to operate. There are huge opportunities for outdoor applications such as autonomous vehicles requiring ultra-reliable low-latency vehicle-to-everything communication as well as fully autonomous drone swarms. Although outdoor channel models exist in literature, the focus has primarily been on the channel from a cellular base-station to an end-user. Although there could still be a base-station communicating with either the vehicles or the drones, inter-device communication is going to be extremely valuable for enabling these applications and there is a need to study these channels.

Chapter 5

Robustness of Cooperative Communication Protocols to Modeling Assumptions

We have theoretically studied wireless channel dynamics and their temporal and spatial characteristics. In this thesis we described two candidate schemes Occupy CoW and XOR-CoW in Chapter 3 and analyzed their performance under *ideal conditions*. Other viable candidates are [59, 60]. However, to make any scheme practical, we need to understand the effects of real-world imperfections. Broadly, we can partition the real-world imperfections into two categories:

- Channel fades being drawn from models that are different from the one assumed. This can manifest either as correlated channels or as channels that are deeply faded with a higher probability.
- Uncontrollable events that are difficult to model precisely – such as stray interference events or out-of-sync packet transmissions.

Consequently, we need to understand the effects of different *modeled and unmodeled events* that could potentially cause severe degradation. This is crucial as **wireless systems supporting URLLC applications must build in robustness**. In order to build a robust communication system, we must ask the following questions: a) what events may cause errors, b) how can we model the effects of these errors on the communication system, and c) what avenues does the communication system have to protect against these events?

Before we answer these questions, we need to ask a higher level question: can we protect ourselves against all error events? In other words, are there some events that we just cannot tolerate? Our goal is to make a wireless system ultrareliable to the impairments for which there is hope of being robust to. If a node were to turn into a persistent jammer, we cannot protect against that, the same as not being robust to placing one of the nodes in a Faraday cage. To put it more precisely, we have three avenues to build in robustness – time, frequency

and space. If there are error events that are correlated in *time and frequency* then the hope is to move the system (or at least the nodes) to a different location where hopefully the error events can be uncorrelated. On the other hands, if error events are correlated on all dimensions, then there is no hope. Those kinds of impairments are both unmodeled and irrelevant for wireless ultrareliability.

With the topic of unattainable robustness out of the way, let us start by answering the following two questions: what events may cause errors and how can we bound their effects on the communication system? This will allow us to create an appropriately sized and shaped "uncertainty ball" around the nominal wireless model. The resulting uncertainty-bounding parameters are summarized in Table 5.1.

- The dominant cause of error is deep fading. The frequency of occurrence of deep fades can be modeled using the nominal fading distribution. However, how much do we depend on the accurate knowledge of the fading distribution? What happens if fades actually come from a different distribution? We account for this using an additional probability of error $\leq p_{off}$ that is independent across links and find that its effect is small (i.e. a little bit of added SNR compensates for it) for medium - large network sizes (Sec. 5.1). We have already seen this in Chapter 4 where we see that there is actually some spatial correlation. But this can be viewed as a small unmodeled probability of error.

- Imperfections in different quantities of interest could potentially cause errors as well. What are some of the quantities/measures that can be imperfect? An obvious and significant one is time/frequency imperfection – mis-synchronized transmissions can lead to decoding errors which can effectively destroy an entire slot. How may this happen? The transmitter sends time and frequency acquisition signals for the receiver to lock onto before it sends the actual payload. If the receiver cannot lock to this synchronization signal (say due to random jitter), then this packet transmission could fail. Channel estimation errors also can have this impact. Another similar cause of errors is abrupt channel changes during a packet. (e.g. a transmitter moving in such a way that it transitions to being shadowed to scatterers or loses/gains line-of-sight to the receiver. We have discussed this in Chapter 4 also.) This can also lead to incorrect decoding of the message. We bound these kinds of errors on an independent *per-slot per transmitter basis using p_c while ensuring that when there are more nodes simultaneously transmitting during a slot, the chance of encountering these kinds of unmodeled errors grows* (Sec. 5.2).

- Another potential cause of errors is interference from other devices in the vicinity. This could come from a network nearby in which a node accidentally transmits at a very high power – thereby causing a burst of interference throughout our network. A failure of the error-correcting code due to an unlucky realization of additive noise is similar, and so is motion of the receiver that causes it to abruptly transition into a shadow relative to important scatterers. More practically, these errors can occur in a

correlated way across time slots but could occur independently across frequency and vice-versa. Therefore, we bound these kinds of events as well on an independent *per-slot time and frequency basis using* $p_g^{(t)}$ *and* $p_g^{(f)}$ *respectively, but in a way that does not compound with the number of simultaneous transmitters*. In this thesis, for the purposes of exposition we only consider the errors that are independent across time slots and model it with p_g. However, the results in either case are equivalent – increase in minimum SNR demanded and the number of times a message needs to be repeated are the same – whether the repetitions are in time or in frequency. We briefly revisit frequency repetitions when combined with time repetitions in Sec. 5.4.

We now answer the last question: what avenues does a wireless communication system have to protect against these events? For errors such as the decoding error caused due to packet mis-synchronization causing packet collisions, these can be combated only by doing retransmissions. It is not that the channel between the transmitters and receiver was faded but rather there was an *uncontrollable error* that caused the transmission to fail. Therefore, **retransmissions i.e, time margin and frequency margins** are the way to combat these bad events that are not about SNR.

What about the errors caused due to fading? As mentioned earlier, spatial correlation could potentially lead to slightly worse channel realizations than an independent realization. The only way to improve the channel itself is by increasing the transmit power to get better nominal SNRs. Therefore, to combat unmodeled channel-fade-related events, we use **SNR margins**. Our model based on the behavior of multipath establishes that the multipath fades only change slowly across time relative to the cycle time, and so all temporal correlations can also be treated the same way, using a small SNR margin while assuming that good channels stay static during the cycle.

Another big assumption that we have made so far has been about the error correction codes – that there exist some Shannon capacity achieving coding scheme that operates in the finite-blocklength regime as well. However, it is well-known that finite-blocklengths causes the performance of codes to be far from optimal. Should we worry about these effects and model it carefully or can we simply give the gap-to-capacity penalty and operate at a higher SNR? The answer to this question is slightly more nuanced and is explored in Sec. 5.7 and the flavor of questions we ask is different from unmodeled error events.

With all wireless effects accounted for in either the nominal model or the uncertainty bounds, we can have some confidence that a system which performs well in theory will indeed be ultrareliable in practice. In the subsequent sections, we look at these effects more closely.

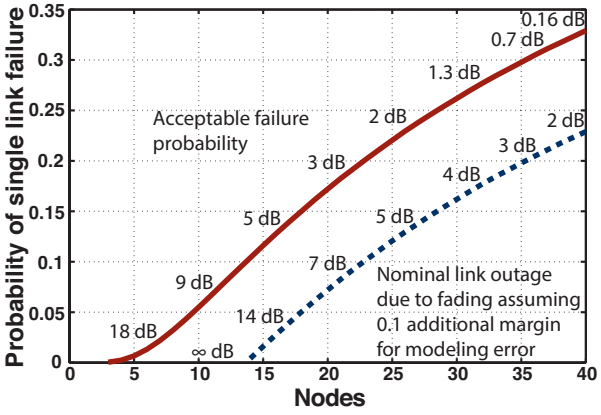
Parameter	Range	Unmodeled events captured
p_{off}	$0 - 0.1$	Imperfections in channel fade modeling. Spatial-correlation-based degradation can also be captured through this term.
p_c	$0 - 10^{-2}$	Errors due to packet mis-synchronizations or channels changing rapidly during a single packet slot. These errors compound with increasing number of simultaneous transmitters.
p_g	$0 - 10^{-2}$	Global errors that are due to burst-interference like events. These errors do not compound with increasing number of simultaneous transmitters. They fundamentally exist at the receivers. These errors are independent across time slots (or in frequency and we briefly comment on it later).

Table 5.1: Uncertainties captured and the parameters associated with them. Except for p_{off}, all of these are essentially independent from time-slot to time-slot if we assume that the communication scheme interleaves repetitions. Some of these might implicitly depend on the length of the time-slots (growing with time-slot length), but this dependence is suppressed here.

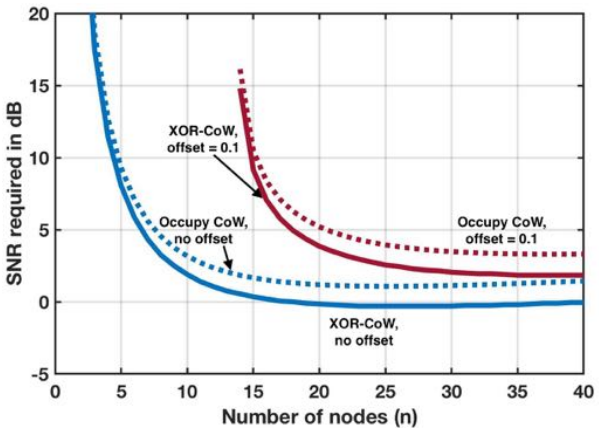
5.1 Effects of uncertainty in channel fade distributions

Bad multipath fades have been modeled to be the dominant cause of potential failures. We assumed that the channel fades themselves come from a Rayleigh distribution. Given the extremely low error probabilities we are targeting in a wireless setting, it is natural to question if we can really trust the fading distribution down to 10^{-9}? What happens if there are unmodeled events (e.g. the exact geometry of the scatterers in the environment) that cause bad fades to occur more often than we had modeled. For instance, in the case of 2 scatterers in the environment, the probability of deep fade is at most 0.01 more than what standard Rayleigh would suggest. We do not have to worry about events like line-of-sight paths that make bad fades occur less frequently. To capture this, we introduce *an extra probability of failure at each link*, p_{off}, on top of the probability of error due to nominal fading, p_w. In this model, the link failure comprises of two parts: one coming from the nominal fading distribution and the other from local modeling error, p_{off}, the total probability of link is $p_{link} = p_w + p_{off}$. This is a local error model where each link gets affected independently – i.e., unmodeled errors themselves are not correlated. Because this bound p_{off} attaches to the individual link fades, we do not assume that it is realized independently across different time-slots in which that same link is active.

Consider the Occupy CoW scheme as described in Sec. 3.2 with n nodes, each sending messages of m bits and total cycle time of T. Nominally, links are modeled as failing if the fade was deep enough i.e., the instantaneous capacity given by $C_{\text{inst}} = W \log(1 + |h|^2 SNR)$ is



(a) The probability of link failure that can be tolerated for Occupy CoW as a function of the number of nodes. The lower curve is 0.1 below and the SNR numbers represent the nominal SNR required to hit that particular link failure probability for Rayleigh fading.



(b) SNR paid to achieve performance robustly in the face of uncertainty. The effects are similar for XOR-CoW. Below 14 nodes, it is not possible to be robust to the $p_{off} = 0.1$ of unmodeled uncertainty specified here.

Figure 5.1: Effects of unmodeled errors on the performance of Occupy CoW and XOR-CoW. We assume the availability of a 20MHz bandwidth channel and every message is of 160 bits long.

less than the rate of transmission. The probability of a bad link under this perfect Rayleigh channel fade distribution model is $p_w = 1 - \exp(-\frac{2^{R_w}-1}{SNR})$ where $R_w = \frac{mn}{T/4}$.

We can look at the *maximum* value of p_{link} that can be tolerated for different number of nodes while keeping the overall probability of failure constant at 10^{-9} in the top curve in Fig. 5.1a. This tells us that if p_{link} is greater than the error in modeling error p_{off}, then increasing SNR to make p_w smaller would be able to digest the modeling error. We see that if $p_{off} = 0.1$, then shifting the maximum tolerable p_{link} down by p_{off} will give us the maximum p_w that can be tolerated which ultimately translates into an increase in the minimum SNR required. We note that for larger tolerable p_{link}, the SNR penalty is smaller (compare the SNR penalty for network size 30 and 15). Note that for smaller network sizes with maximum tolerable $p_{link} < p_{off}$ i.e., $N \leq 13$, there exists no SNR that can robustly support these requirements. The p_{off} of 0.1 alone is too much.

XOR-CoW has similar response to channel fading distribution uncertainty (Fig. 5.1b). We conclude that unmodeled local errors such as not having perfect knowledge of fading distributions do not cause heavy degradation in the performance of schemes that rely on the availability of independently faded links. In fact, channel-correlation-induced extra link failure can be captured in p_{off} as ultimately the effects of both are the same – reduction in nominal SNR while 'preserving' essential independence across space.

5.2 Effects of channel changes within a packet

In Sec. 4.5.1, we studied the temporal characteristics of wireless channels within the packet duration. We saw that good channels tend to remain good for short packets if multipath is the only effect causing fades. However, a rapid change in the channel coefficient within a single packet, say due to crossing into a shadow of an obstacle relative to many scatterers, when the channel is good could lead to decoding errors. What would be the effect of these errors?

Our approach is to bound the probability that a packet in a time-slot is corrupted by some maximum probability of such corruption. There are clearly two qualitatively different kinds of corruption that we need to watch out for. One is where the corruption happens "at the transmitter" — for example, if the transmitter had moved in such a way that its channel rapidly transitioned into or out of a shadow relative to say a line-of-sight path. When multiple nodes are transmitting simultaneously, the receiver is decoding using the combined channel which depends on the DSTC and the individual channel realizations. If any of the channels change during the transmission (causing a corrupt packet to effectively be sent), it could potentially lead to decoding errors. In fact, the more nodes that transmit simultaneously, the more likely one of the channels could change mid-packet causing a decoding error. We bound this error using p_c which has a *cumulating effect* when there are more nodes transmitting during a single message slot. Essentially, if r nodes are transmitting in a single slot then the probability of slot success is $(1-p_c)^r$ However, we assume that these are independent from one message slot to the next.

The other qualitatively different kind of corruption happens "at the receiver" — for example, if the receiver is the one that moves suddenly into or out of a shadow relative to a particularly important scatterer. These errors do not cumulate with multiple simultaneous transmissions. We bound this error using p_g with a probability of slot success being $(1-p_g)$. This is also modeled as being independent from one message slot to the next.

The advantage of these kind of unmodeled uncertainty bounds is that they can encompass many different physical sources of imperfection. For example, channel estimation errors at the receiver can contribute to both p_c and p_g depending on how the pilots and preambles are transmitted. Synchronization errors are clearly a part of p_c. Interference bursts and imperfections of the error-correcting codes are just as clearly a part of p_g. For all of these, the important thing is that these phenomena (just like the feared rapid transition into a shadow) are either finely localized in time or finely localized in frequency.

Because they are finely localized in time, it is imperative to take advantage of time margins here to be robust to them. These unmodeled events are considered as being independent across slots (this is what the assumption of interleaved repetitions justifies at the level of each message), so there is a time diversity of sorts vis-a-vis unmodeled corruptions. This is unlike the traditional notion of time diversity where fading channels change from one slot to the other. Here, the channel quality (being a good/bad channel) remains the same across slots but these other errors happen independently across those same slots. We see that to combat these events, we need to have *multiple relaying slots* for each message, i.e., $k_1 > 1$

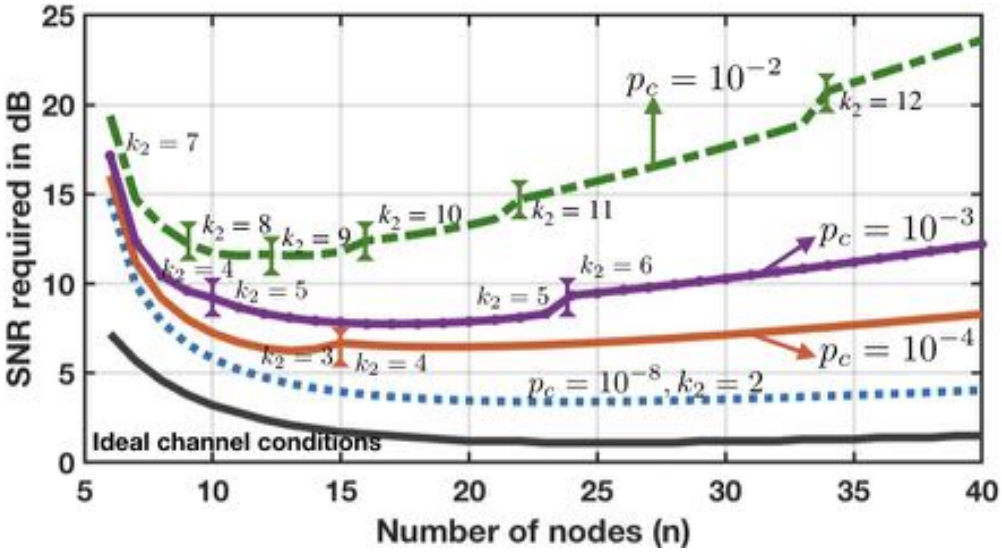


Figure 5.2: Performance of Occupy CoW when the channel changes within a packet (transmitter centric) causing decoding errors. Here $p_g = 0$ and $p_{off} = 0.01$.

and $k_2 > 1$, where k_1 is the number of times a message is transmitted initially and k_2 is the number of times the message is transmitted in the relaying phases. These k_1, k_2 are not the same as hops in a multihop protocol.

We illustrate this in Fig. 5.2 where we see that when p_c is super low $\approx 10^{-8}$, the effect is almost negligible. However, we see a very interesting phenomenon for mid-high bounds $10^{-4} - 10^{-2}$. We optimize over different values for k_2 and pick the one that minimizes the nominal SNR. The curves associated with $p_c = 10^{-2}$, $p_c = 10^{-3}$, and $p_c = 10^{-4}$ are annotated with this optimal number of k_2. The minimum SNR demanded as well as the values of k_2 tell us a story – when there are errors that may occur during a slot and the probability of such errors can compound with the number of simultaneous transmitters then, the only way to combat such errors is by transmitting the message multiple times. However, we are still constrained by the latency limits. Therefore, the available time needs to now be divided up to support multiple retransmissions which brings the spectral efficiency demanding during the slot much higher. For instance, if the initial demanded spectral efficiency (before multiple retransmissions) was 0.25 b/s/Hz and $k_2 = 3$, then the new spectral efficiency demanded that the links have to support is $0.25 \cdot k_2 = 0.25 \cdot 3 = 0.75$ b/s/Hz. It is the increase in demanded spectral efficiency that causes the increase in minimum SNR demanded. Alternatively, we could require the use of wider bandwidth channels.

5.3 Effect of channel change during a cycle

We studied to what extent channels change within a cycle and redefined the notion of "coherence time" in Sec. 4.5.2. In this section, we address the following question: if all available

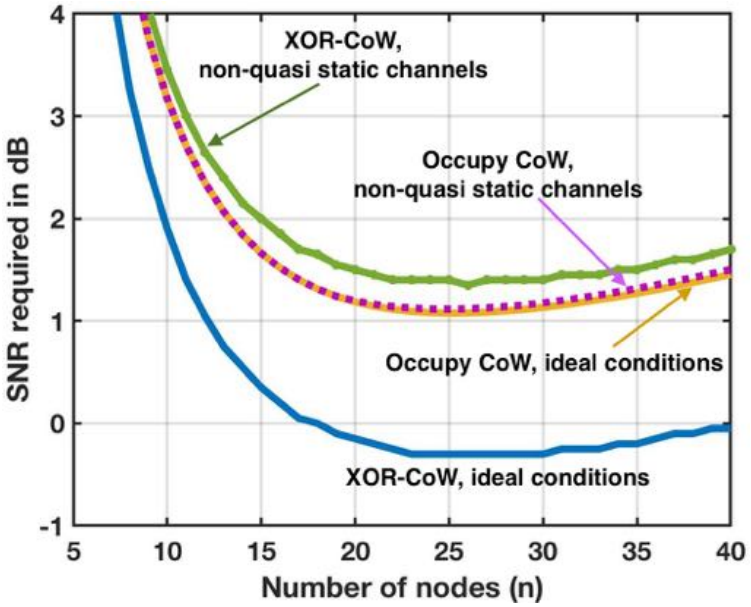


Figure 5.3: Performance of Occupy CoW and XOR-CoW protocols when the channel changes during a cycle but not within a packet. New fades are realized which breaks reciprocity.

relays were to be employed, what are the effects of significant channel changes during a cycle (henceforth referred to as non-quasi-static channels) if channels remain completely static during a single packet transmission (so the effects seen in Sec. 5.2 do not occur)? As mentioned earlier, the easiest way to account for this is to fold these rare events into the p_{off} term earlier. However, it is possible to analyze this even more conservatively. Here, we briefly examine the performance when channels refresh at phase boundaries of the protocol: for eg., it might change between the downlink and uplink phases for the Occupy CoW protocol. This effectively translates any changes during a cycle/phase into an easier to analyze effect.

Such non-quasi-static channels introduce more randomness into the system. In the Occupy CoW protocol, this extra randomness might give some nodes two chances to directly establish a link to the controller, before and after a mid-cycle channel change. This means that the downlink-only or uplink-only performance of a protocol can improve due to the extra diversity introduced by a channel change. However, this breaks the assumption of reciprocity and consequently, the combined performance of uplink & downlink takes a small hit. In the quasi-static case, a path that worked for two-hop downlink to a node was guaranteed to work for two-hop uplink for the same node. In the presence of potentially changing channel fades, this is no longer true. Each node must potentially find two independent paths to the controller — one for uplink & one for downlink.

The hit for the XOR-CoW protocol is more pronounced. The performance hit is due to the decoupling between uplink and downlink — this can lead to a smaller set of nodes that have both uplink and downlink information for any given node — and thus a smaller set of nodes that can help anyone who does not have a direct link to the controller. The

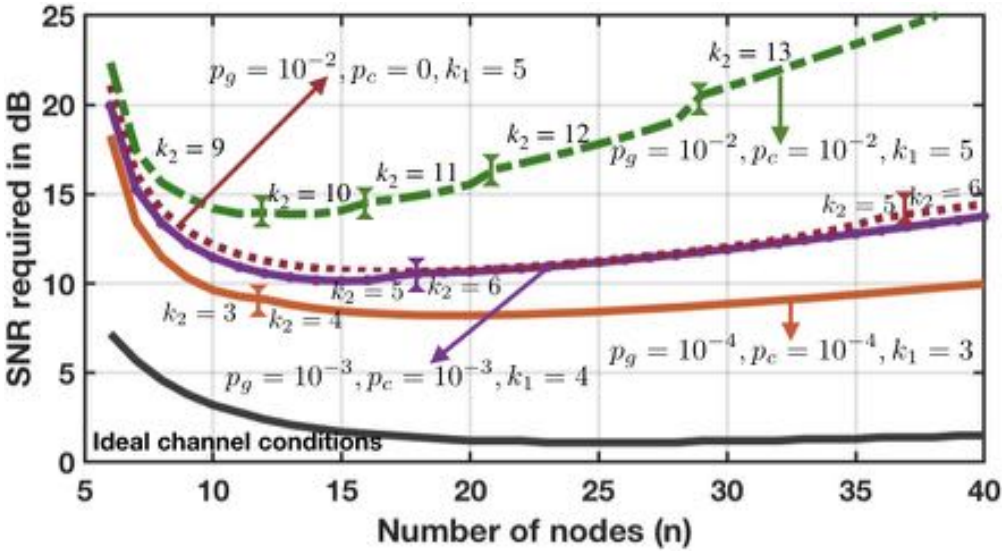


Figure 5.4: Performance of Occupy CoW when receiver-center, transmitter-centric and fading uncertainty errors occur. Here $p_{off} = 0.01$.

degradation in performance is captured in Fig. 5.3. The key takeaway is that this entire effect is small even in the worst case, and only costs an SNR margin of a little over a single dB.

5.4 Combined effect of all error events

Until now, we have analyzed the impact of different kinds of events and phenomenon *individually*. It is important to put together all these events and analyze the combined effects to understand how much it costs to get the robustness we need by budgeting the SNR and time margins appropriately. We capture this in Fig. 5.4 where we account for the following events: a) deep fade causing links to be bad captured by the nominal model for p_w, b) bounded uncertainty in fading model $p_{off} = 0.01$, c) global per-slot bounded badness such as interference, error-correcting-code failures, or receiver shadow transitions that do not cumulate with the number of transmitters p_g (different values explored), and d) bounded per-transmitter badness due to say mis-synchronized packets, channel estimation errors, or transmitters transitioning into shadows p_c (different values explored) which cumulate with the number of transmitters. The exact formula used to obtain the curves can be found in Appendix C.

We immediately notice: the number of retransmissions required in the initial phase k_1 is primarily dependent on p_g as that is the main unmodeled event to guard against in the initial phase as there are no simultaneous transmissions. As p_g and p_c increase, we see increases in the number k_2 of retransmissions in the relaying phase. The increased retransmissions induce a need for higher raw spectral efficiency which drives up the SNR required. In fact, if

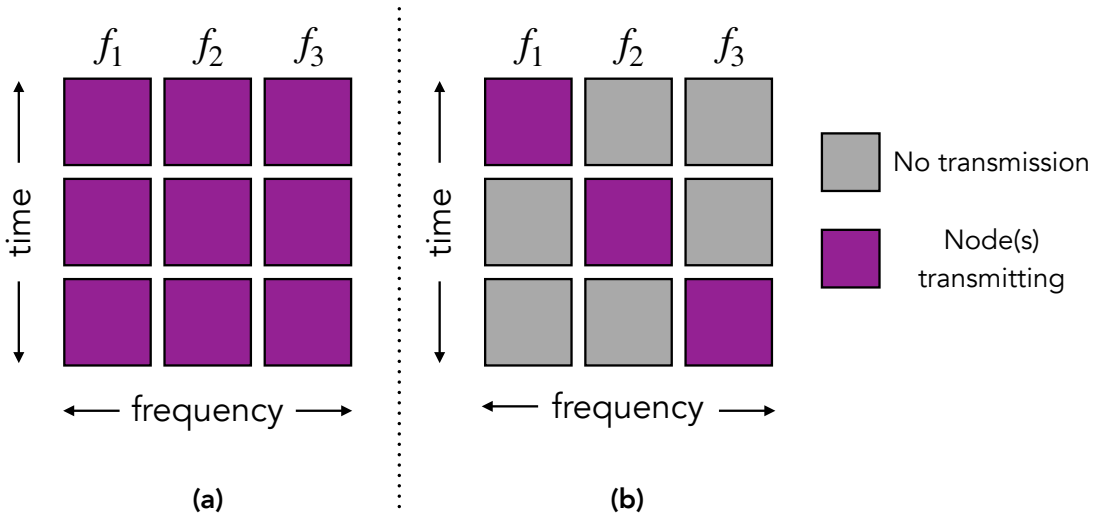


Figure 5.5: Ways of using frequency-time resources when a single message is being transmitted. On the left-hand side, we see an allocation where node (or nodes if this is during relaying phase) transmit in all time slots and in all frequency slots. On the right-hand side, node(s) hop between frequencies during different time slots.

we consider $p_g = p_c = 10^{-2}$ essentially uncontrollable unmodeled events occurring 1% of the time such as shadowing transitions and budget an extra 3dB for finite-blocklength codes, we see that we need to roughly operate in the regime of 15dB to 20dB nominal SNR to be robust to most realistic error events, whereas under ideal channel conditions, we needed to be around 3dB. Most of this is due to the increased bitrate needed to support the repeated transmission of the small packets.

5.5 Time and Frequency repetition structure

We mentioned earlier that global errors like the interference events captured by p_g may occur correlated across time-slots but independently across frequencies. Then, the only way to be robust against these error events is to transmit at different frequencies as well. Now, instead of k_1 repetitions, we divide the available transmission time per-message into $k_1^{(t)}$ slots and the available frequency into $k_1^{(f)}$ sub-bands. Providing a total of $k_1^{(t)} \cdot k_1^{(f)}$ retransmissions in total. If, error events were completely uncorrelated in time and frequency, then the behavior would scale as in Fig. 5.4 with one of $k_1^{(t)}$ or $k_1^{(f)}$ being 1. However, if global errors were correlated either in time or frequency (but importantly not both), then all of these curves would go up – essentially requiring a higher number of total repetitions pushing the demanded spectral efficiency higher.

However, it is important to note that transmitting on different frequencies at the same time is challenging. Similarly listening on all frequencies at the same time is also challenging

(it could cause receiver de-sensitivity). Theoretically, it is possible to have a node transmit simultaneously in all time slots and frequency slots (as shown in Fig. 5.5.a) and be able to harvest the full benefit of multiple (possibly independent) trials. However, if a node wants to use different frequency resources, realistically, it needs to hop between these frequencies (as shown in Fig. 5.5.b). This clearly means that all resources are not used. However, this type of resource allocation still protects against correlated global errors. For instance, let us say that there is an interference event on frequency f_1 and none of the time-slots are usable. Then, transmitting on all available resources such as the one shown in Fig. **??**.a would be wasteful. Although hopping across different frequencies may possibly result in less wasteful transmissions, it does mean that we are less resistant to error events than the scheme using all resources would be (such as the compounding effect captured by p_c). Another advantage of frequency hopping would be that it frees up resources that could be used either by the same network or by a neighboring network supporting less-critical applications. We omit details discussions of SNR demanded and other parameters in this thesis (we conjecture them to be of flavor similar to the results in Sec. 5.4 and Fig. 5.4) but emphasize that system designers should take these into account while designing ultra-reliable low-latency wireless communication systems.

5.6 Why the nominal model matters

So far we have argued why it is important to have picked a nominal model that took some care to understand how the spatial distribution of fades gives rise to reliable multiuser diversity. However, if we had failed to model the effects of spatial correlation carefully or if we had not considered unmodeled events and budgeted for them through time-margins and instead took a pessimistic approach, how would the penalties look like? We briefly consider these two scenarios here and point the reader to Appendix C for detailed analysis.

5.6.1 Effect of spatial correlation of channels

We studied spatial correlation of channels in Sec. 4.6 and saw that the channels are indeed correlated, but since we can assume that nodes are more than 2λ apart, this correlation actually only leads to a small degradation in nominal SNR – of about 0.05dB. So a nominal model of independence is justified. Let us now construct a pessimistic non-nominal model about spatial correlation and understand the effects of such a model.

Suppose, every new channel had a probability q of coming from an independent fading realization and with probability $1 - q$ the channel were identical to a channel that has already been realized, so we get no diversity. This might sound reasonable, but Fig. 5.6 demonstrates how this affects cooperative-communication-based URLLC protocols. The SNR curves decay as the number of nodes increases but a low probability of independence has a severe impact. Especially for smaller networks, around 20 nodes, the SNR penalty is about 40dB. However, from Sec. 4.6 we know that when the nodes are sufficiently separated,

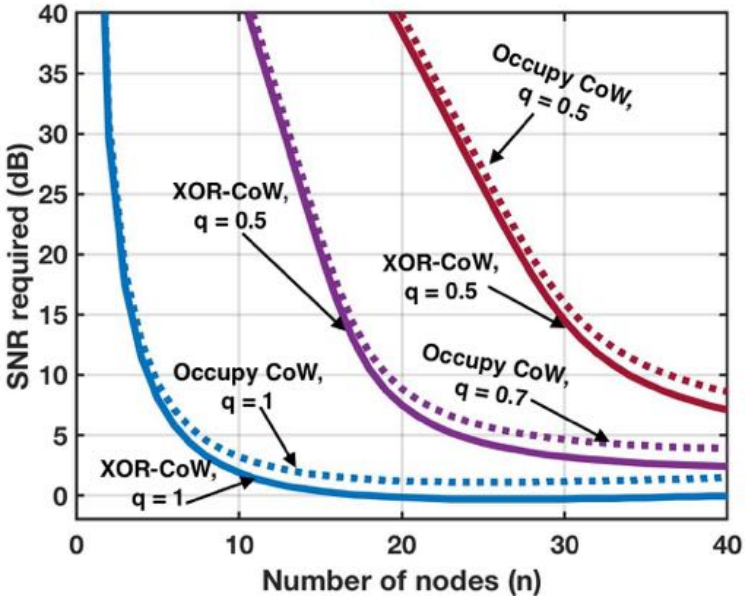


Figure 5.6: Performance of Occupy CoW and XOR-CoW protocols using a pessimistic spatial correlation model. q represents the probability of an independent fade on a channel.

then actual multipath channel realizations, although correlated are not too much worse than completely independent realizations. The equations used to generate Fig. 5.6 can be found in Appendix. C.4

5.6.2 Effect of synchronization impediments

The performance of real systems is far from ideal. Timing and frequency synchronization pose the biggest hurdles in making practical cooperative communication systems as the performance of most synchronization algorithms degrades with more relays [177]. We have so far bounded this effect using p_c which corresponds to unmodeled error terms that cumulate with simultaneous transmitters. Because of this cumulative behavior, we think that a good wireless communication system should try to seek a kind of "sparsity" to be robust, analogous to what the analysis of wideband channels suggests for traditional communication [178]. It is safer and simpler to avoid too many simultaneous transmissions. In Chapter 6, we provide ways to leverage data-driven learning to greatly reduce the need for many simultaneous transmissions. However, implementation constraints might not behave in the cumulative manner that we assume for p_c. Therefore, we analyze the effect of restricting the total number of simultaneous transmissions to some maximum number dictated by the synchronization protocol.

Fig. 5.7 shows the significant impact on performance when the maximum number of simultaneous transmitters for each message is capped. We see the SNR increase with increasing number of nodes because the additional nodes in the system stop being useful as

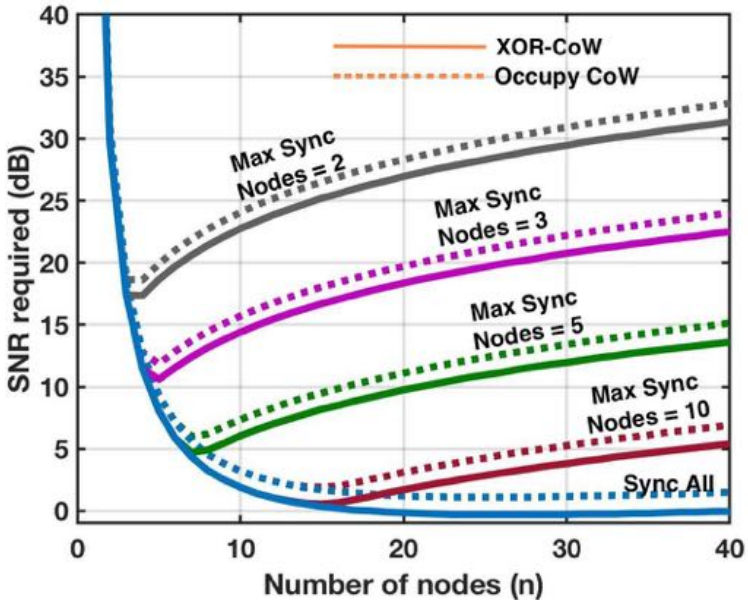


Figure 5.7: Performance of Occupy CoW and XOR-CoW protocols with a cap on the number of nodes that can transmit simultaneously. Here, $k_1 = k_2 = 1$ and there are no repetitions.

new relays which leads to the increase in nominal SNR required. Again, we may be able to combat this through time margins by having multiple relaying slots with a smaller number of transmitters per slot.

5.7 Effects of Finite-Block-Length Error Correction Codes

So far we have ignored the effects of realistic error-correcting codes – instead assuming the existence of perfect capacity-achieving codes. It is well-known that finite-blocklengths causes the performance of the codes to be far from optimal [116]. Should we worry about these effects and model it carefully or can we simply give the gap-to-capacity penalty and operate at a higher SNR? Essentially, to model or not to model, that is the question.

We take the approach of modeling some of the main error events that these codes may face – additive noise. We also analyze in-depth how may we save on the minimum SNR demanded by intelligently selecting channel thresholds. However, our main takeaway is that, the effects of finite-blocklength codes are not too complex and are not going to contribute to any substantial error event. In fact, the effects of additive noise can easily be countered by time and frequency repetitions as has been discussed in this chapter so far. To *emphasize the role of time and frequency repetitions*, we assume that there are no such repetitions in the rest of the section.

Although the approach we describe could be used with any error-correcting code, for

concreteness (and ease of numerical analysis), in this section we look at a concatenated Hamming+Reed-Solomon code because the main purpose is to understand the effect of error-correcting codes. The main insight is that the demands on the error-correcting code are different in different phases. In the initial phases of Occupy CoW and XOR-CoW, the key is to spread the message to many relays as possible and for this, the error-correcting code does not have to be ultrareliable. Moderate reliability is fine. However, when the messages are finally delivered to their ultimate destination, without time or frequency repetitions, there is no diversity with respect to the additive noise and it is vital that the error-correcting code be ultrareliable. Because multiple relays were very likely to have been recruited earlier in the protocol, there is less of a fear of simultaneous deep fades.

5.7.1 Idealized analysis of the 2-hop downlink CoW-protocol

Before considering the situation with finite block-length codes, we first review how the probability of success for downlink is derived in Chapter 3 – to once again draw attention to the key pain points. A note about the notation: we use calligraphic script to denote sets; the random variable associated with the size of a set is in upper case and the instantiation being considered is denoted by lower case.

Denote the set of nodes with direct controller links by $\mathcal{A}$. Other nodes may connect to the controller through these nodes in a two-hop fashion. At its essence, our simple analysis fundamentally examines how the size A of this set $\mathcal{A}$ changes what happens in the second phase. Here, the transmission rate in the downlink phases is $R = \frac{mn}{T/4}$ and hence the probability of a single link outage due to fading (assuming Rayleigh fading and a Shannon capacity-achieving code) is $p = 1 - \exp(-\frac{2^R-1}{SNR})$. Then A follows a Binomial distribution.

Under the ideal conditions that the channels do not change during a cycle and are reciprocal, the probability of cycle failure is the probability that at least one of the nodes in the set $\mathcal{N}\backslash\mathcal{A}$ does not connect to $\mathcal{A}$. Since success means at least one connection is made between $\mathcal{N}\backslash\mathcal{A}$ and $\mathcal{A}$, a conservative assumption is that a node in $\mathcal{N}\backslash\mathcal{A}$ only uses its strongest connection to $\mathcal{A}$ and disregards its other, weaker connections. We call this the 'loudest talker' model because the alternative would be to account for the specifics of the DSTC and how the aggregate SNR behaves. Thus we have:

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

Thus, the probability of cycle failure is given by:

$$P(\text{fail}) = \sum_{a=0}^{n-1} P(A = a) \cdot P(\text{fail}|A = a) \tag{5.1}$$

$$= \sum_{a=0}^{n-1} \left(\binom{n}{a} (1-p)^a p^{n-a} \right) \left(1 - (1 - p^a)^{n-a}\right) \tag{5.2}$$

5.7.2 Effect of additive noise at receivers

Practical receivers introduce some sort of additive noise to the signal. To guard against this, we use error correction codes. Here, the main quantity of interest is the probability of incorrect decoding denoted by $F_{\mathcal{S}}(r_P)$ where $\mathcal{S}$ is the coding scheme under consideration (including the block length) and r_P is the power received at the receiver. This essentially captures the error event when despite the presence of good channel, an error event cause the overall transmission to fail. Although in this section we carefully model these effects, our overall message is that modeling them approximately is good enough – as long as there are enough time and frequency repetitions designed in.

5.7.3 Loudest talker analysis

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

5.7.3.1 Downlink

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

$$P(\text{error}|h_i) = F_{\mathcal{S}}(|h_i|^2 SNR). \tag{5.3}$$

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

$$P(\text{single}) = \int_0^\infty P(\text{error}|h_i) f\left(|h_i|^2\right) d\left(h_i^2\right) \tag{5.4}$$

where $f\left(|h_i|^2\right)$ is the pdf of an exponential random variable. Let the nodes which succeeded in downlink phase 1 be called $\mathcal{A}$ (with cardinality A). Then probability that $A = a$ is a binomial with probability of failure given by Eq. (5.4) as seen earlier. We now consider a node (say j) that hasn't heard its downlink message from the controller. In downlink phase 2, the relays $\mathcal{A}$ will simultaneously broadcast the packet and the node j can only reap the benefits of the loudest link. Let the random variable associated with the channel fade of the loudest link from $\mathcal{A}$ to node Y be denoted by $H_a^{max} = \max\left(|h_{1,j}|^2, |h_{2,j}|^2, \ldots, |h_{a,j}|^2\right)$. The pdf of H_a^{max} (denoted by $f_a^{max}(h^2)$) is given by

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

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

$$P(\text{error}|h_{1,j}, \ldots h_{a,j}) = F_{\mathcal{S}}\left(\left(h_a^{max}(j)\right)^2 SNR\right) \tag{5.5}$$

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

$$P(\text{loudest}|A=a) = \int_0^\infty P(\text{error}|h_{1,j}, \ldots h_{a,j}) f_a^{max}(h^2) dh^2 \tag{5.6}$$

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

$$\begin{aligned} P(\text{fail}) &= \sum_{a=0}^{n} P(A=a) \left(1 - (1 - P(\text{ec}|A=a))^{n-a}\right) \\ &= \sum_{a=0}^{n} \left\{ \binom{n}{a} (1 - P(\text{single}))^a P(\text{single})^{n-a} \right\} \\ &\times \left\{1 - (1 - P(\text{loudest}|A=a))^{n-a}\right\} \end{aligned} \tag{5.7}$$

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

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

$$p_{fade} = P(r_P < r_{th}) = 1 - \exp(-10^{\frac{r_{th} - t_P}{10}}) \tag{5.8}$$

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

$$p_{add} = F_{\mathcal{S}}(r_{th}) \tag{5.9}$$

This threshold of reliability (p_{add}) that divides acceptable from unacceptable is an internal parameter of the analysis that can be optimized to get the best overall bound. Thus, the probability of failure of a link is given by

$$p_{link} = p_{fade} + (1 - p_{fade}) p_{add}. \tag{5.10}$$

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

number of phases of the protocol rather than the number of nodes. Downlink analysis now simplifies to:

$$P(A = a) = \binom{n}{a}(1 - p_{link})^a (p_{link})^{n-a} \tag{5.11}$$

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

$$P(\text{fail}|A = a) = 1 - \left(1 - p_{fade}^a - \left(1 - p_{fade}^a\right) p_{add}\right)^{n-a} \tag{5.12}$$

Combining Eq. (5.11) and (5.12) we get,

$$P(\text{fail}) = \sum_{a=0}^{n} P(A = a)P(\text{fail}|A = a) \tag{5.13}$$

5.7.4 XOR-CoW Loudest Talker Analysis

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

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

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

$$p_{fade,DU} = 1 - \exp(-10^{\frac{r_{DU} - t_P}{10}}). \tag{5.14}$$

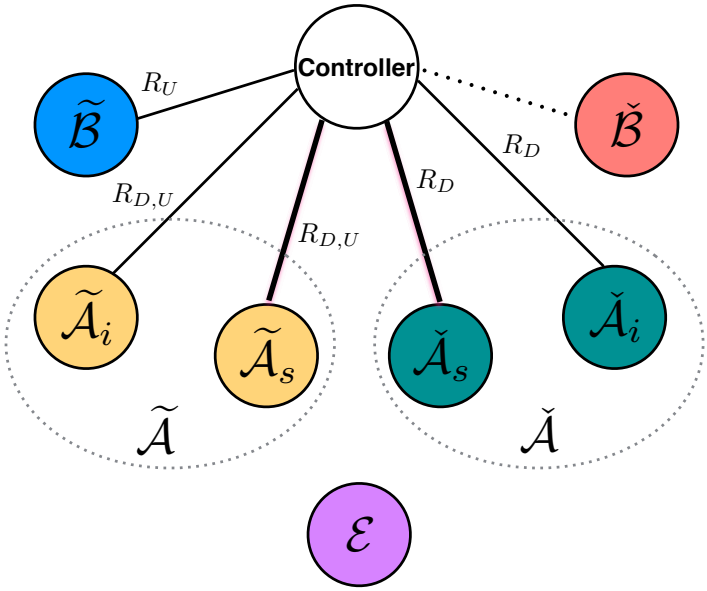


Figure 5.8: The figure shows the different sets and their connectivity to the controller. The interconnection between the sets needed for success are not shown. The rates annotating the links are the rates in which the links to the controller are present. The bold links belong to the superior set which has links to the controller during the XOR phase.

Let $p_{add,D}$ and $p_{add,U}$ be the probability of failure due to additive noise (despite having had enough receive power) in the downlink and uplink phase respectively. The probabilities are different because the blocklengths of the messages in these phases are different. We partition the set of nodes $\mathcal{N}$ (as shown in Fig. 5.8) into different sets for ease of analysis:

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

In order to act as a relay in the XOR phase, a node *must* have the downlink information. Hence only nodes in $\widetilde{\mathcal{A}} \bigcup \check{\mathcal{A}}$ can act as relays in the XOR phase. As we have further restricted the receive power needed to overcome the additive noise threshold in the XOR phase, only a subset of the nodes in $\widetilde{\mathcal{A}} \bigcup \check{\mathcal{A}}$ can help. Let the subset of the nodes in $\widetilde{\mathcal{A}}$ with "superior"

links to the controller be $\widetilde{\mathcal{A}}_s$ (the rest form $\widetilde{\mathcal{A}}_i$) and the subset of nodes in $\check{\mathcal{A}}$ with "superior" links to the controller be $\check{\mathcal{A}}_s$ (the rest form $\check{\mathcal{A}}_i$).

We enumerate the ways in which nodes can succeed.

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

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

$$B(n, m, p) = \binom{n}{m} (1-p)^m p^{n-m}. \tag{5.15}$$

Failure is the event that even one of the nodes did not get its downlink information or wasn't able to transmit its uplink information. We will calculate the probability of failure by unraveling the state space. As mentioned earlier, the probability of having a bad fade in the downlink and uplink phases is then given by Eq. (5.14)

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

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

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

We denote by $\mathcal{A} = \widetilde{\mathcal{A}} \bigcup \check{\mathcal{A}}$ the set of nodes that succeed in downlink. Thus, we get that in addition to having good links, the additive noise at these receivers were low enough to allow decoding. The probability of failing due to additive noise despite having enough receive power in the downlink phase is $p_{add,D}$ which depends on the block length and the coding rate as already discussed earlier. Thus, we get that the probability that $A = a$ conditioned on $G = g$ is given by $P(A = a|G = g) = B(g, a, p_{add,D})$.

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

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

$$\begin{aligned} p_{XDU} &= P(\text{link not good for XOR}|\text{link good for DU}) \\ &= 1 - \exp(10^{\frac{r_{DU} - t_P}{10}} - 10^{\frac{r_X - t_P}{10}}) \end{aligned} \tag{5.16}$$

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

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

$$P\left(\text{success of } \widetilde{\mathcal{A}}\right) = 1.$$

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

$$P\left(\text{success of } \check{\mathcal{A}}_s\right) = (1 - p_{add,X})^{\check{a}_s}.$$

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

$$P\left(\text{success of } \check{\mathcal{A}}_i\right) = ((1 - (p_U)^{a_s})(1 - p_{add,X}))^{\check{a}_i}$$

where $p_U = p_{fade,DU} + (1 - p_{fade,DU})p_{add,U}$.

Consider the set $\widetilde{\mathcal{B}}$ which succeeds if they have a 'superior' link to the controller (with probability $1 - p_{XDU}$) or they connect to $\mathcal{A}$ in uplink phase and have a superior link to the

set in the XOR phase. Let the probability of success for a node (before considering the effect of thermal noise at the receiver) in $\widetilde{\mathcal{B}}$ be $q_{\widetilde{\mathcal{B}}}$. Then we have

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

where $p_U = p_{fade,DU} + (1 - p_{fade,DU})p_{add,U}$. Thus we have

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

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

$$\begin{aligned} f_e = p_U^{a_s} + \Bigg\{ &\left[\sum_{k_s=1}^{a_s}\left(B(a_s,k_s,p_U)p_{XDU}^{k_s}\right)\right] \times \\ &\times \left[\sum_{k_i=0}^{a_i}\left(B(a_i,k_i,p_U)p_{XDU}^{k_i}\right)\right]\Bigg\} \end{aligned} \tag{5.17}$$

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

$$P\left(\text{fail of node in else}\right) = f_e + (1-f_e)\left(1-(1-p_{add,X})^2\right).$$

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

Combining the success equations above we get,

$$\begin{aligned} P(\text{success}|\text{states}) = &P\left(\text{success of else}\right) \cdot P\left(\text{success of } \widetilde{B}\right) \\ &\cdot P\left(\text{success of } \check{\mathcal{A}}_i\right) P\left(\text{success of } \check{\mathcal{A}}_s\right). \end{aligned} \tag{5.18}$$

Finally,

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

where

$$\begin{aligned} P(\text{states}) = &P(\widetilde{A}_s = \widetilde{a}_s, \check{A}_s = \check{a}_s | \widetilde{A} = \widetilde{a}, \check{A} = \check{a}) \\ &\cdot P(\widetilde{B} = \widetilde{b} | G = g, A = a) \cdot P(\tilde{A} = \tilde{a} | A = a) \\ &\cdot P(A = a | G = g) \cdot P(G = g) \end{aligned} \tag{5.19}$$

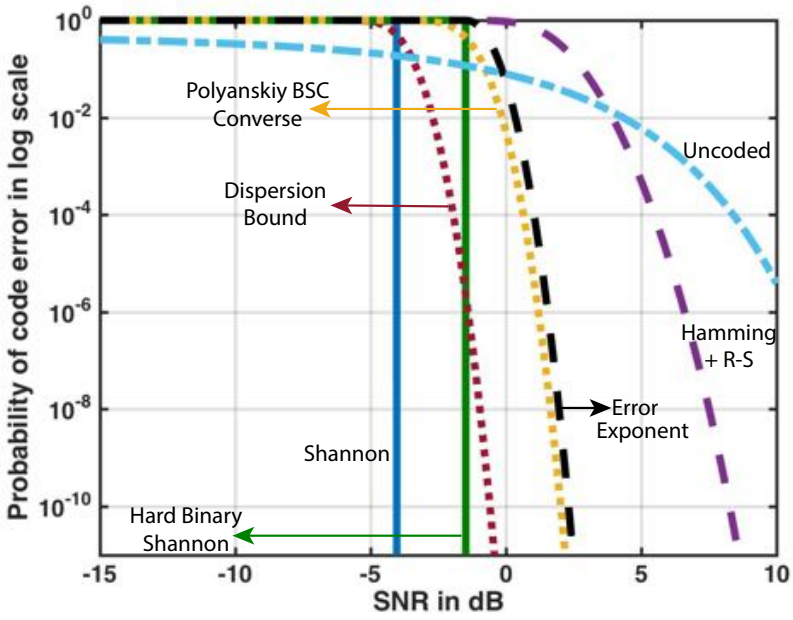


Figure 5.9: Waterfall curves with a block-length of 333 symbols per codeword at a coding rate of $R = 0.48$.

5.7.5 Impact of Finite-Block-Length Error Correction Codes

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

Concatenated Hamming+Reed-Solomon code

A short Hamming code is used to fix isolated bit flips with a Reed-Solomon code wrapper to clean up the rest. In particular, we consider a $(7, 4)$ code, and each of the $16 = 2^4$ Hamming codewords forms a symbol in the Reed-Solomon alphabet. When we need a field size of more than 16, we just group two Hamming codewords together and so up to 256 RS symbols can be obtained by putting two together, and so on. We then generate RS parity symbols such that the coding rate is close to $R = \frac{m \times n}{T/4}$ for any given m, n and T. The exact expressions for decoding error $F_{\mathcal{S}}(SNR)$ as a function of the SNR at the receiver can be computed and we have used the (suboptimal) half-minimum-distance decoding expressions for an underlying BPSK signaling assumption for our plots.

We begin by looking at the waterfall curves (in Fig. 5.9) for various coding techniques

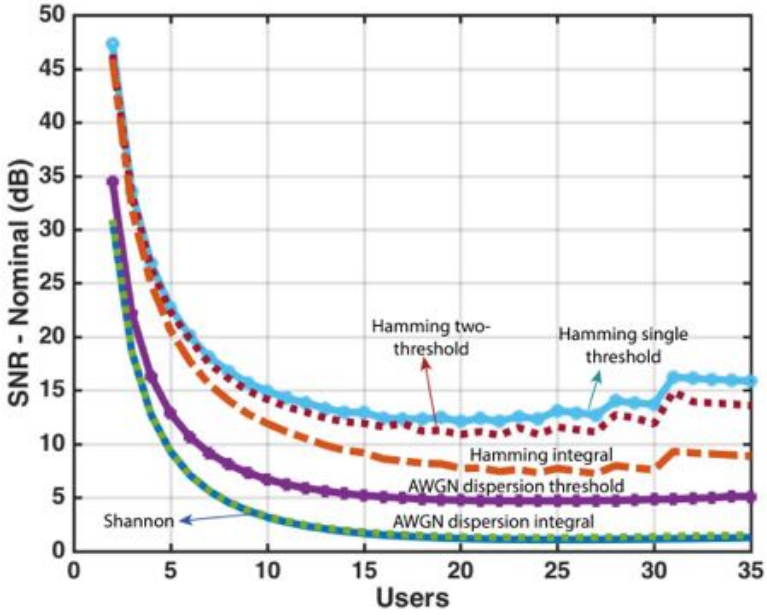


Figure 5.10: For 2-hop downlink, the SNR required for a perfect Shannon capacity code versus the SNR required for using various coding schemes is shown. The reliability is 10^{-9}.

for $n = 30$ nodes which corresponds to a coding rate of $R = \frac{30*160}{10000} = 0.48$ at an 'uplink' blocklength of $10000/30 = 333$ symbols per codeword. As expected, the performance for the simple concatenated Hamming+Reed-Solomon code is much worse than the channel dispersion-based bound [116]. We observe that once in the waterfall region, the block error probability for the Hamming+Reed-Solomon scheme falls rapidly from 10^{-2} to 10^{-10} in a matter of 4dB.

We first look solely at the performance of the downlink 2-hop protocol with uplink-like blocklengths. The reason for considering this particular scenario is that downlink is the simplest to analyze because of the independence of the links used in various phases. This allows us to get a better understanding of various issues that the ultra-reliability requirement causes or introduces. The uplink blocklengths are used because they are shorter and hence more vulnerable to additive noise. Additionally, the uplink blocklengths are the shortest blocklengths so we get an estimate for the worst case scenario that we might deal with. In Fig. 5.10 we consider the following curves: a) the Shannon code curve which gives us the lower bound on the power required; b) the AWGN-dispersion curve derived using the integral model as described in Eq. (5.7); c) the dispersion code curve for the fade + additive noise model from Eq. (5.10) with p_{add} being set to 10^{-10}; d) the concatenated Hamming+Reed-Solomon code curve using the integral model as described in Eq. (5.7); and e) the concatenated Hamming+Reed-Solomon code curve for the fade + additive noise model from Eq. (5.10) with p_{add} being set to 10^{-10}. We heuristically set p_{add} to 10^{-10} since the target probability

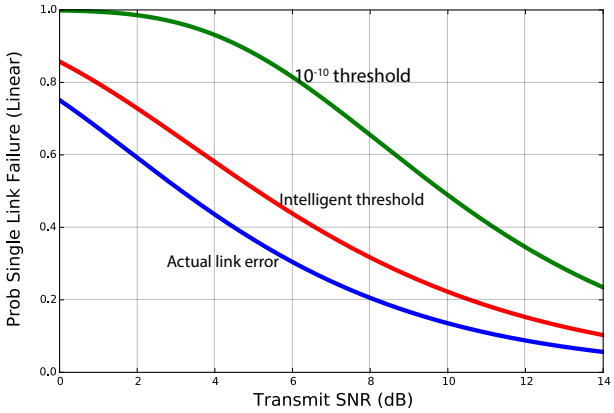


Figure 5.11: Comparison between bounds for a single link failure using different models for coding rate corresponding to $R = 0.48$ and blocklength of 1000/3.

of cycle failure is 10^{-9}.

The underlying reason for the gaps in Fig. 5.10 between the downlink curve calculated with the full integral (as described in Eq. (5.4)) and the one calculated using the $p_{add} = 10^{-10}$ threshold bound (described in Eq. (5.10)) can be seen by examining Fig. 5.11. At any transmit SNR, the thresholding bound significantly overestimates the probability of failure which translates to an increase in transmit power needed to achieve the same performance. A third curve is shown that uses an intelligent search over the value of p_{add} to get as close to the actual value of p_{link} as possible.

Fig. 5.12 shows the transmit SNR required to achieve our target reliability while using the XOR-CoW protocol. As explained earlier, the analysis of the XOR-CoW protocol using the integral approach is computationally intractable and hence not plotted. However we do plot curves with a) p_{add} set to 10^{-10}; and b) where we search over p_{add} for each phase. The line corresponding to the Shannon capacity code gives us a lower bound on the transmit power required. The dispersion-based line corresponding to the adaptive search over p_{add} gives us a good ballpark lowerbound on how a 'good' finite blocklength code can seem to perform using this style of analysis. The performance of the concatenated Hamming+Reed-Solomon code is similar to that in Fig. 5.10. The close match suggests to us that the downlink integral curves are indeed essentially the right answers even for the XOR case.

The more interesting aspect is to look in Fig. 5.13 at the receive SNR thresholds that are selected for the XOR phase vs the downlink/uplink phase when we allow those thresholds to be optimized. By tolerating a lower receiver power for the downlink-uplink phases, we allow for a potentially larger number of relays for the relaying phase even as the resulting probability of additive noise induced error is higher. More relays means that we can more easily count on getting higher receiver power in the XOR phase – thus getting higher reliability in the relaying phase. This allows for lowering the transmit power required by around 4dB which agrees with the number from the waterfall curve at 30 nodes. This possibility of "partial credit" is why the naive prediction of simply adding the capacity-gap to the Shannon-style

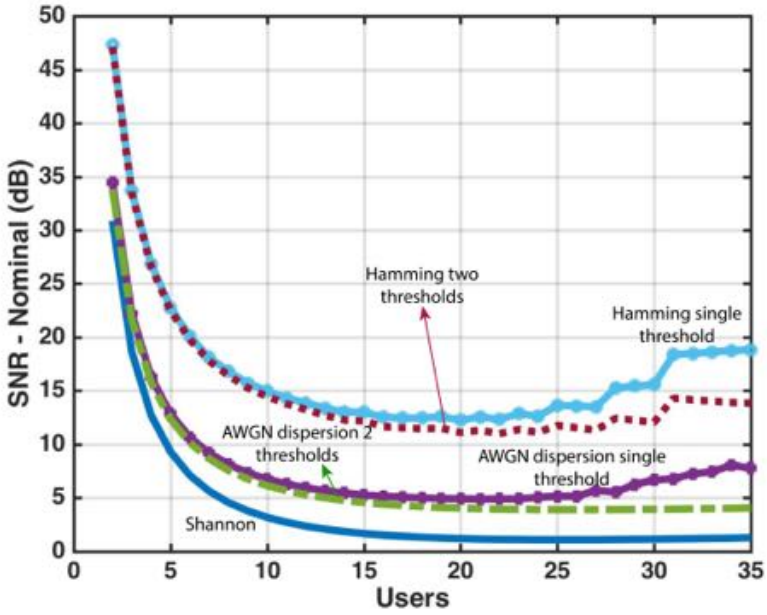


Figure 5.12: The SNR required under the assumption of existence of a Shannon capacity code versus the SNR required for using a practical code for XOR-CoW protocol. The target reliability is 10^{-9}.

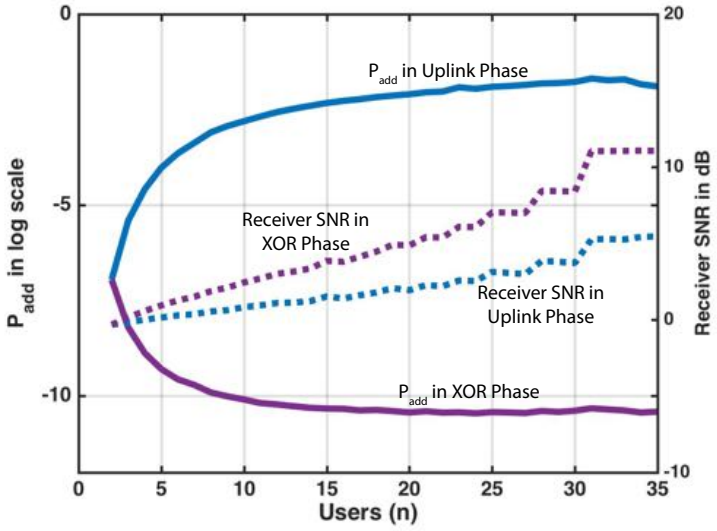


Figure 5.13: Optimized Receiver SNR thresholds (and the corresponding additive noise error probability) for the XOR-CoW protocol using the simple Hamming+RS Code.

analysis is too conservative. The error-correcting code in the early phases is not called upon to hit probabilities of error of 10^{-9}. That is only required at the final phase.

5.8 Conclusions and Future Work

In Chapter 4 we exorcised the fear of unfortunate spatial correlations so we bounded the rest of the uncertainty using three terms, each of which we believe encompassed events that would be independent across time-slots: (a) a bound on the probability of unmodeled receiver-centric temporary outages of time-slots; (b) a bound on the probability of unmodeled transmitter-centric temporary corruptions that would compound if there were multiple simultaneous transmissions in a single time-slot; and (c) a bound on the additional unmodeled probability of fading for a given channel. We argued that between all of these bounds, interference, error-correcting code issues, synchronization issues, shadowing transitions, and other propagation effects are all covered.

To be robust against the first two, a communication scheme has to have time margin by repeating the same message using interleaved slots. By identical logic, we could also have added a fourth uncertainty bound for unmodeled corruption of a frequency slot that might span many time slots. The same schemes would work except they would also need to hop between frequencies as they hopped into time slots. The required number of such hops would not have to increase if the uncertainty bound for frequency corruption was not bigger than the bound for time-slot corruption since what the hopping is seeking is an independent chance to experience something closer to the nominal model. Furthermore, as long as the URLLC system insisted that only one of its own messages was being transmitted at any given time, there is clearly no additional counterpart of p_c for compounding errors for per-channel unmodeled dynamics. However, if multiple messages were transmitted in overlapping times but in different channels, then such a term would need to exist — although it would be small if the frequency slots were very well separated. For the last, an increased SNR margin is required. The use of repetitions also increases the SNR required since messages must be successfully communicated using shorter slots and thus higher spectral efficiency. The combination of explicit modeling of known effects and bounding channel uncertainty in a way that captures the "shape" of wireless protocols allows us to have confidence in ultra-reliability.

Works like [116] and [117] tell us that once blocklengths are short, no code can be perfect. For low-latency communication, short blocklengths are essential. For ultrareliability, multiple phases and the prospect of relaying is essential to harvest the required diversity of fading. Should we worry about these effects and model it carefully or can we simply give the gap-to-capacity penalty and operate at a higher SNR? We answered this question by removing any possibility for time and frequency repetitions to focus on the kinds of events caused by finite-blocklength codes. What we saw is that in the initial phase, the goal is not ultrareliability but reaching the maximum number of relays. The code can therefore be run at a much more moderate error probability — similar to traditional wireless communication systems

that will use ARQs to achieve reliability. The relaying phase must be made as reliable as possible as it is the last chance to succeed. This means that simply adding together the gap-to-capacity to the Shannon bound is too conservative when thinking about ultrareliable low-latency wireless communication. We can do significantly better.

However, the question arises – is this careful modeling necessary? The answer is not really. Yes, we need to pay the gap-to-capacity penalty. However, the careful thresholding is not really necessary. Additionally, codes do not have to be ultra-reliable in the relaying phase. We can build in robustness to these errors by having time and frequency repetitions as we have seen in this chapter.

The fact that URLLC systems can be made robust to unmodeled uncertainties at the 10^{-2} to 10^{-3} level means that such uncertainties can be tracked and learned through data-driven self-monitoring of a wireless system. After all, they will manifest as small anomalies where a packet that didn't succeed in one repetition does in another, despite nominally facing the same channels, multiple times per second. By contrast, a 10^{-9} event will not be seen with any reasonable frequency to support learning. Consequently, learning actually might have an important role to play in allowing systems to reduce their spectral and time footprint while maintaining reliability, if they can assume that these unmodeled uncertainties (such as shadowing or local interference) are not going to change abruptly.

As mentioned in this chapter, spreading the messages out in time and frequency is paramount to protect against bad events. A consequence of this spreading is that resources open up – for instance if only one node is talking at a time in one frequency sub-band, then the other sub-bands are just idle. These resources can be shared with non-critical applications in the environment and leads to interesting questions about co-existence. Although the question of co-existence is not explored in this thesis, it is an important one that needs to be addressed especially if the technology has to penetrate widely.

Chapter 6

Predicting Relay Quality For URLLC

Wireless communication for URLLC is a challenging problem which we have so far addressed with a mixed approach. We first abstracted away lots of effects and considered the main effect a wireless communication system has to fight – multipath channel fading. Then, we looked at channel dynamics in depth to build on the basic channel model and to model events (such as the change in channel quality) that are essential to consider to build a robustly reliable communication system. We then considered the effects of different modeled events and also simplified the structure of 'unmodeled' events by considering their worst case behavior. We looked at the avenues needed to be robust to modeled and unmodeled error events – time margin, frequency margin and SNR margin. We realized that although in theory simultaneous transmissions seem like a great idea, they may introduce or worsen some hurdles and challenges.

Consider the case when synchronization mismatch or channel dynamics may cause an error at the receiver. The higher the number of transmitters, the higher the chance that one of the transmitter may go out of synchronization or that a channel might change. We explored this in detail in Sec. 5.2. However, the disadvantage of having a smaller number of transmitters transmitting is that we have less robustness to channel fading and we need to combat this through SNR margin as explored in Sec. 5.6.2. In fact, if we non-adaptively restrict the number of relays that may simultaneously transmit, then the minimum SNR required to meet the reliability demand grows tremendously – restricting the maximum number of relays to 5 increases the operating SNR by 15dB even without accounting for time and frequency margins (Fig. 5.7). To have a moderate nominal SNR, we need 13 or more simultaneous relays for moderate network sizes of > 20. Not only do we need to combat synchronization mismatch due to inaccuracies in local oscillators by dedicating significant resources to time and frequency synchronization [179], getting a large number of nodes to transmit also wastes energy.

A key insight comes to our aid to combat these challenges – we don't care about having lots of channels, we care about having *good channels*. Whenever we have considered building reliability, we look at the fact that channels could be bad too often and so we need many channels in order to ensure that at least one of them is robustly good. This is why we have so

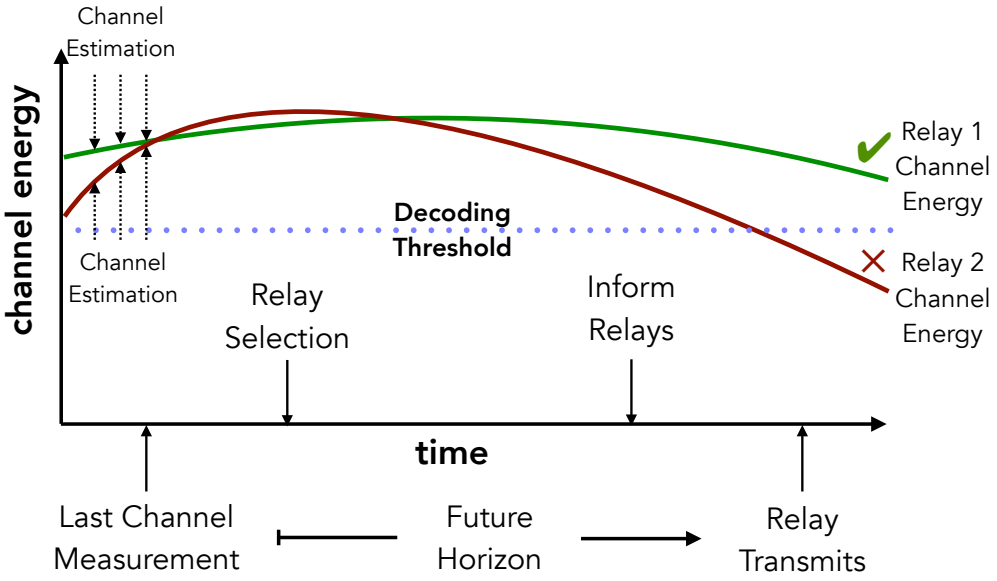


Figure 6.1: After estimating channels, relays are selected and informed of their selection. During the time between channel estimation and the relays transmitting, the channels may change. So relay selection is a prediction problem.

far been considering lots of nodes simultaneously transmitting the same message. However, only the good channels actually deliver the message to the receiver. If there is some way to identify which of the many channels will be good with a high probability, then we don't actually need lots of nodes transmitting. In this chapter, we will explore this very question – can we choose a small set of relays that will have good channels to the destination *reliably* when they actually need to relay?

We focus on selecting nodes that will have a high SNR channel to the destination(s) during the relaying phase. Relay selection schemes have been extensively studied, but these schemes traditionally assume that the channels remain precisely the same from the last channel measurement to when they actually relay. Most studies focus on the decoding error at the receiver based on current channel coefficients [180]. Let's consider using these relay selection schemes without taking channel dynamics into account. If we assume that a channel might change significantly in a millisecond with probability 10%, then to achieve a reliability of 10^{-9}, we will need to choose *at least* 9 relays to be robust to these potential channel changes. This is an improvement over the 13 relays required if they are chosen blindly, but it is still a large number of nodes to have simultaneously transmitting. However, if fading channel dynamics are stable enough to predict channel quality more reliably (say an error rate of at most 10^{-3}), then we could rely on fewer relays.

To facilitate such predictions, we must solve a few important challenges. We only have potential relay's channel coefficient measurements for a certain period of time. Based on these channel coefficient measurements, we decide which nodes(s) to nominate as relays and disseminate that information. The node actually relays at a *future* time. In the time between

the last channel coefficient measurement of the relay and actual relaying, the channel may change (potentially adversely) as illustrated in Fig. 6.1. But how do wireless channels change and how fast? Can channels be reliably predicted based on past channel realizations? If so, for what future horizons and at what reliability? In Chapter 4, we have already studied the dynamics of wireless channels due to the physics of multipath. We will now build upon the models from there and the results to build good relay selection schemes.

6.1 Relay Selection Setup

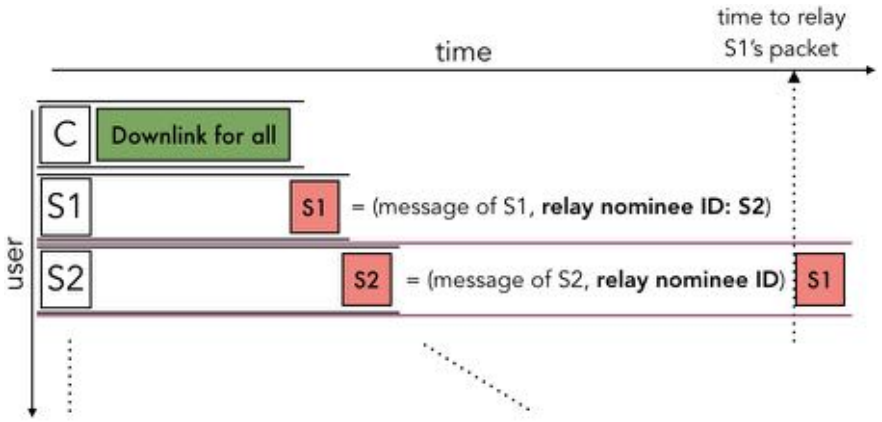


Figure 6.2: Adaptation of the Occupy CoW (as well as XOR-CoW) protocol to account for relay nomination schemes for the star information topology. The users append their uplink message to add channel quality information as well as relay nominee information in it and only those relays transmit during the relaying phase – whether downlink, uplink or XOR-ed relaying phase.

Let us first establish the setup of the relay selection process before we further study the schemes. Each message stream has an origin node and can have possibly multiple destinations. The origin of the message nominates the relay for that message. The nomination criterion should be based on past channel knowledge.

In the case of star-topology where there is downlink traffic from the controller to each node and there is uplink traffic from each node to the controller, the nomination process can actually be further simplified. Lets assume that the controller appends to its downlink packet the last known channel state between itself and **each user**. This way all nodes get to know a slightly old channel state between controller (c) and all n users ($s_1, \ldots, s_n$) – denoted by h_{c,s_i}, $i = 1, \ldots, n$. The users keep track of the channel between themselves and other all users (as well as the controller) by constantly updating the channel measurement. As nodes channel sound before they transmit and all nodes are always listening (as long as they are not transmitting), nodes can decode the channel state information between themselves and other nodes. Thus a specific user s_i, has the channel state information of the inter-user channels

h_{s_i,s_j} where $j \neq i$. However user s_i **does not** have information of channel between s_j and s_k where $j \neq i$ and $k \neq i$. Thus, each user s_i has some knowledge of channels between itself and potential relays and the potential relays and the ultimate destination – the controller.

However, the controller in this case would not be able to easily get access to the channel information between the potential relays and the multiple destinations that its messages need to reach. Therefore, in the case of star topology, it makes sense for the users (not the controller) to nominate relays for both the **downlink and uplink** messages. Each user would then append to its uplink packet, the set of relays that it nominates for the relaying phase. This is captured in Fig. 6.2. In the rest of the chapter, we will only focus on the star-topology setup for the purpose of exposition. Additionally, we will also only focus on relay-based success – we will ignore direct success between a node and the controller and only look at the success rate through the relay. This way we can isolate the relay-selection success and remove the influence of direct path successes.

Its important to clearly define the error event. There is a time difference between when the relay is selected and when the relay *actually relays*. While we select the relay, we are making a prediction about its quality *in the future*. Therefore, **the error event is selecting a relay that ends up having a bad channel in the future.**

6.2 Relay Quality

We begin the investigation by revisiting some of the concepts already seen in Chapter 4 but reproduced here for completeness sake.

Rayleigh-faded channels have traditionally been modeled using a sum-of-sinusoids like in Jakes's model [139] and we consider the same setting here. Let there be a static single-antenna transmitter in the middle of the room and a single-antenna mobile receiver moving at a constant speed v in some random direction inside the room (illustrated in Fig. 4.2 in Chapter. 4). Let the transmitter be transmitting a tone at frequency f_c (wavelength λ_c). The channel coefficient between the transmitter and the receiver at any time t is given by

$$h(t) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \exp\left(j \frac{2\pi (d_i^{(\mathrm{Rx})}(t) + d_i^{(\mathrm{Tx})})}{\lambda_c} \right) \tag{6.1}$$

where $d_i^{(\mathrm{Rx})}(t)$ is the distance of the scatterer i from the receiver at time t and $d_i^{(\mathrm{Tx})}$ is the distance of the scatter i from the transmitter (both are assumed to not be moving for simplicity). The $1/\sqrt{n}$ normalization is to keep the marginal variance the same across different numbers n of scatterers.

The channel coefficient at any point in time is marginally distributed as a complex normal, and the channel coefficient process through time can be modeled as a Gaussian process. The parameters that we need to define the Gaussian process are the means and the covariance functions which depend on the distance that the receiver has moved. We assume that the velocity $\vec{v}$ of the receiver is constant over the time durations of interest such that the position

of the receiver $\vec{s}(t)$ at time t is given by

$$\vec{s}(t) = \vec{s_0} + \vec{v}t = (x_0 + vt\cos\phi, y_0 + vt\sin\phi) \tag{6.2}$$

where $\vec{s_0} = (x_0, y_0)$ is the initial position of the receiver at $t = 0$ (uniformly distributed in the room), ϕ is the angle of motion of the receiver with respect to the x-axis (uniformly distributed over $[0, 2\pi)$). Let the position of scatterer i be given by $\vec{s_i} = (x_i, y_i)$. The distance of the receiver from scatterer i at time t is given by

$$\begin{aligned} d_i^{(\mathrm{Rx})}(t) &= \|\vec{s}(t) - \vec{s_i}\| \\ &= \sqrt{d_i^{(\mathrm{Rx})}(0)^2 + (vt)^2 + 2vtd_i^{(\mathrm{Rx})}(0)\cos(\theta_i - \phi)} \end{aligned} \tag{6.3}$$

where $d_i^{(\mathrm{Rx})}(0)$ is the distance of the receiver from the scatterer i at $t = 0$ and θ_i is the angle made by the line joining the scatterer and the receiver at time $t = 0$ which is independent of ϕ.

We have derived the covariance of the in-phase, $\Re(h(t))$ and the quadrature components, $\Im(h(t))$ of the wireless channel as a function of speed v and time t in Chapter 4 and give only the final result here. Let us denote this covariance by $k(v,t) = \mathbb{E}[\Re(h(t))\Re(h(0))] = \mathbb{E}[\Im(h(t))\Im(h(0))]$. We have,

$$k(v,t) = 0.5 \cdot J_0\left(\frac{2\pi}{\lambda_c}vt\right) \tag{6.4}$$

where $J_0(\cdot)$ is the Bessel function of the first kind.

Consider the channel between a pair of nodes, say a source and a relay. Suppose the source has knowledge of past channels given by $\vec{h} = \begin{bmatrix} h_1 & h_2 & \dots & h_m \end{bmatrix}^T$ from times $\vec{t} = \begin{bmatrix} t_1 & t_2 & \dots & t_m \end{bmatrix}^T$. We want to find the distribution of h_{m+1} at time t_{m+1} conditioned on $\vec{t}$ and $\vec{h}$. We assume that the channel coefficient variation is a Gaussian process, and use simple linear estimation. Therefore, $\{\vec{h}, h_{m+1}\}$ form a multivariate normal and the distribution of h_{m+1} conditioned on $\vec{h}$ is a complex normal distribution. Let

$$\mathbf{K} = \begin{bmatrix} k(v,t_1-t_1) & k(v,t_2-t_1) & \dots & k(v,t_m-t_1) \\ k(v,t_2-t_1) & k(v,t_2-t_2) & \dots & k(v,t_m-t_2) \\ \vdots & \vdots & \ddots & \vdots \\ k(v,t_m-t_1) & k(v,t_m-t_2) & \dots & k(v,t_m-t_m) \end{bmatrix}$$

be the covariance matrix of the in-phase and quadrature components corresponding to the times of observations so far. Let $\mathbf{K}_* = \begin{bmatrix} k(v,t_{m+1}-t_1) & \dots & k(v,t_{m+1}-t_m) \end{bmatrix}$ be the covariance matrix of the in-phase and quadrature components corresponding to the future time of interest and the times of observations so far. Also, let $\mathbf{K}_{**} = [k(v,t_{m+1}-t_{m+1})] = [k(v,0)]$ be the variance of the in-phase and quadrature components. Let $\vec{h}_I = \mathrm{Re}\{\vec{h}\}$ be the vector of the in-phase components of $\vec{h}$ and $\vec{h}_Q = \mathrm{Im}\{\vec{h}\}$ be the vector of the quadrature components

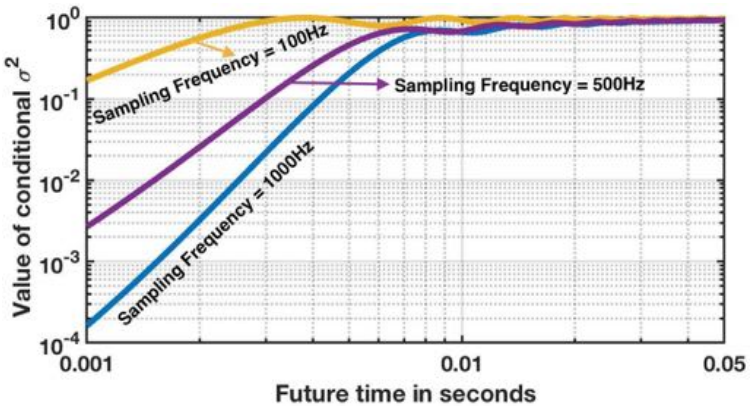


Figure 6.3: The conditional variance of the channel energy distribution predicted by Eq. (4.8) as a function of future time and sampling frequency of the channel coefficient. The higher the sampling rate, the lower the variance. The farther out in the future, the closer the variance becomes to the unconditional variance.

of $\vec{h}$. Then, the mean of the distribution of the in-phase μ_I and the quadrature component μ_Q of h_{m+1} conditioned on $\vec{t}$ and $\vec{h}$ is given by

$$\mu_I = \mathbf{K}_* \mathbf{K^{-1}} \vec{h}_I, \qquad \mu_Q = \mathbf{K}_* \mathbf{K^{-1}} \vec{h}_Q. \tag{6.5}$$

The conditional variance of both the in-phase and quadrature components is given by

$$\sigma_c^2 = \mathbf{K}_{**} - \mathbf{K}_* \mathbf{K^{-1}} \mathbf{K}_*{}^T. \tag{6.6}$$

As mentioned earlier, the goodness or the quality of a channel is captured by the energy ($|h|^2$) in the channel. The conditional distribution of the energy of the channel $|h_{m+1}|^2$ is given by

$$|h_{m+1}|^2 \sim \text{Rice}(\nu, \sigma_c) \tag{6.7}$$

where $\nu = \sqrt{\mu_I^2 + \mu_Q^2}$, σ is given by Eq. (6.6) and $\text{Rice}(\nu, \sigma_c)$ is the Rician distribution with parameters ν and σ_c. The value of σ_c^2 is a direct indicator of the variability of the channel at the future time t_{m+1}. In addition to the distance into the future, the value of σ_c^2 crucially depends on how fast we are sampling the channel as seen in Fig. 6.3.

For a relay to be a good relay, two different channels (source-relay h_{sr} and destination-relay h_{dr} channels) have to be good at the future time when the node relays. We define the badness b_r of a relay r as the probability that either $|h|_{sr}^2$ or $|h|_{dr}^2$ is not a good channel. Since $|h|_{sr}^2 \sim \text{Rice}(\nu_{sr}, \sigma_{sr})$, the probability that the energy is less than the threshold γ is $F_{sr} = 1 - Q_1\left(\frac{\nu_{sr}}{\sigma_{sr}}, \frac{\gamma}{\sigma_{sr}}\right)$ where Q_1 is the Marcum Q-function [181]. The same is true for h_{dr}. Therefore, the badness of a relay r is given by

$$b_r = P(\min\left(|h|_{sr}^2, |h|_{dr}^2\right) < \gamma) = F_{sr} + F_{dr} - F_{sr} \cdot F_{dr}. \tag{6.8}$$

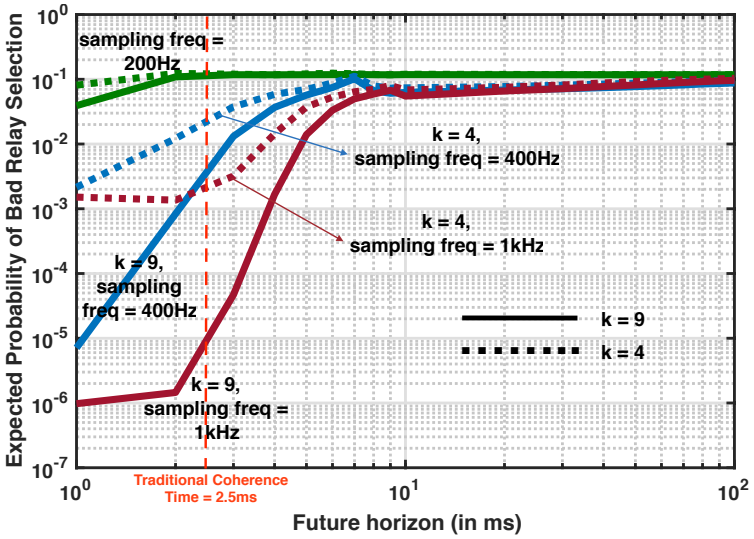


Figure 6.4: Probability of the best relay not being good enough for varying sampling frequency and future horizons. The model order is 3, the number of relays to choose from is $k = 9$ (solid curves) and $k = 4$ (dotted curves), nodes are moving in a random direction at speed 10m/s, nominal SNR is 5dB and the center frequency is $f_c = 3$GHz. The traditional coherence time for a radio moving at these parameters is 2.5ms (corresponding to moving $\lambda/4$ at 10m/s) which is marked on the plot.

Given a choice of k independent relays (numbered 1 to k), we ask the question, what is the probability that the one relay we choose is not good enough. The relay we do choose is the relay with the smaller badness metric. Therefore, the probability that the relay chosen from a set is bad is the smallest badness metric of that set. Therefore, that probability p_{bad} is given by,

$$p_{bad} = \min(b_1, b_2, \ldots, b_k). \tag{6.9}$$

The main variables that determine the value of p_{bad} as defined in Eq. (6.9) are, 1) the sampling frequency, 2) the future horizon, 3) the number of potential relays to choose from (k), and 4) the nominal SNR and the rate of transmission (they together determine the threshold γ). Fig. 6.4 shows the effect of sampling frequency and future horizon. We generate this curve using Eq. (6.9). As the badness metrics themselves are random variables, we simulate the setting to get the expected probability. Consider the solid set of curves for $k = 9$. We see that prediction error for nearer future horizons (under 3ms) for sampling frequency of 1kHz – which is about the bandwidth containing 99.99% fading process energy for nodes moving at a speed of at most 10m/s, is excellent ($< 10^{-4}$). This suggests that nominating at most 2 relays would be sufficient to achieve reliability of 10^{-9}. As the future horizon increases, the prediction error quickly degrades to the unconditional outage probability, which is the error probability if a random relay is picked without any prior knowledge. We also see that the performance when we sample at lower frequencies (400Hz or lower) is bad and cannot be

used in practice. Similar findings hold true for $k = 4$ nodes as shown in Fig. 6.4. This means that channel state information really does need to be steadily monitored and disseminated in a low-latency way within the network.

6.3 Practical relay selection

We have reformulated the relay-selection problem with emphasis on detailed channel dynamics knowledge. The key point that we care about is to select the right relay (or the right set of relays) for each message such that, in the future, when the relays have to transmit, at least one of them have good channel(s) to the destination(s). Our main takeaways are the following:

- **Sampling** above a minimum frequency is crucial. This minimum frequency is determined by the center frequency, the maximum speed at which the nodes are moving **and** the prediction fidelity needed. If we are okay with much lower fidelity, then sampling at a lower frequency would suffice.

- The **future horizon** over which one needs to predict is also obviously important. Random processes like channels can be predicted to much higher accuracy if we are predicting something in the near future – say a few hundred microseconds. However, predicting something that is farther out in the future – say tens of milliseconds is as good as a fresh draw without any prior information. Luckily, low-latency applications like the ones we target need relay qualities to be predicted for milliseconds out in the future.

- The **number of relays available to choose from** is also important for two reasons. First, let us say that we are able to predict the future quality of a relay with arbitrarily high accuracy. If we only get to choose between two relays, then there is high chance that both the relays have bad channels in the future. Even though our prediction accuracy was high and it would predict both of the relays to be bad, it didn't ultimately help us reach our goal – to deliver the packet to the destination. However, if we had more relays to choose from, we would be able to predict the right set of relays with good channels in the future and have a successful transmission. Second, can we truly make a super accurate predictor? The coherence time plot in Fig. 4.7a suggests that, for the future horizons that we care about – in the range of milliseconds can only be predicted at about 10% accuracy. How can we choose relays in an ultra-reliable fashion? Here is where the number of relays again plays an important role. Though our per-relay quality prediction is lower than what we would like, due to the availability of several relays to choose from, the probability of incorrectly predicting the best relay's quality is low (in the order that we can tolerate).

6.4 Visualization of channel variation

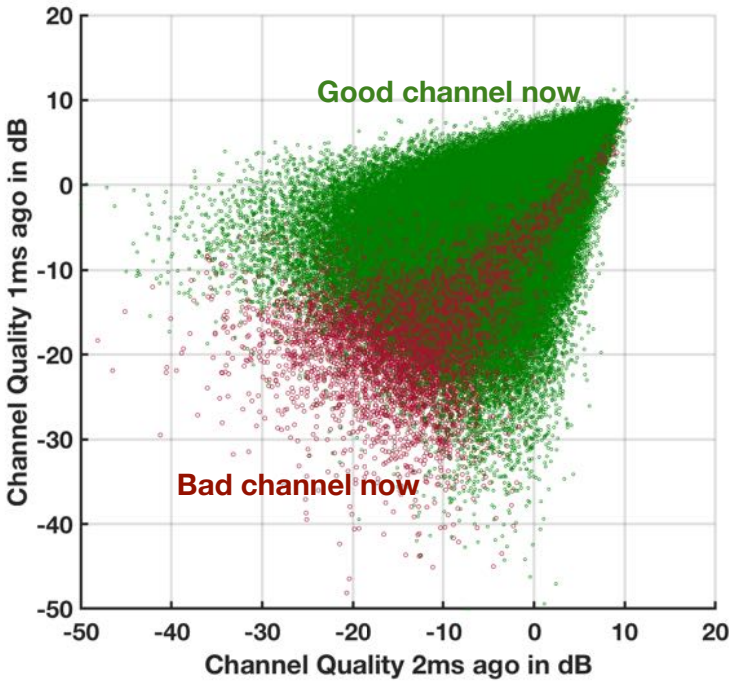


Figure 6.5: A scatter plot of various channel evolutions with a good channel now is marked in green and a bad channel is marked in red. Essentially, it shows how the channel quality now has a non-linear dependency on the channel qualities 1ms and 2ms in the past.

Before we build relay quality predictors, its natural and important to visualize the temporal variation of the channels to fundamentally understand dependencies as well as linear/non-linear aspect of these processes. Visualizations and the patterns that emerge can guide us to build good predictors. Fig. 6.5 plots the variation of the channel quality. The axis represent the natural logarithm of the channel quality ($|h|^2$) 2ms and 1ms ago on the X-axis and Y-axis respectively. The points of the scatter plot represent the individual channel variation and the colors represent whether the channel now is a good channel (in green) or a bad channel (in red).

This simple scatter plot with only two features actually tells us a lot of things. This tells us obviously why just the immediate past (i.e., 1 ms) channel coefficient alone is not a good enough indicator for channel quality in the future. Channels do not just simply follow the past channel. There is an interesting dependency on the channel coefficients both 1ms and 2ms ago. We can extend this to include channel coefficients from 3ms ago but the dependency on it is much smaller as suggested by Eq. (4.6) and Fig. 4.3. Another interesting aspect to note is that the dependence of channel quality on these two features is highly **non-linear**. There is no simple hyperplane that separates the good channels (green colored points) from the bad channels (red colored points). This suggests that a more nuanced,

non-linear predictor might be able to capture this phenomena better than a linear predictor. However, as linear predictors are generally simpler to implement and study, we first look at a few simple predictors that are extremely easy to implement in real hardware and then explore more complicated predictors and their performances.

6.5 Simulation Network Setup

We envision these channel quality predictors and relay selection schemes to be used in highly-variable environments with a lot of uncertainty. We need to simulate settings and apply these techniques on those settings that are more emblematic to real-world high-variable environments. Therefore, we have the following setup. There is one controller and 30 client nodes in the environment. Each node is traveling at a speed of 10m/s in random directions. These directions are not known to each other. Client nodes have knowledge of channels between themselves and other client nodes and the controller and the client nodes. The relative velocity is not known and is not estimated. The number of scatterers in the environment is also not known. It is under these uncertainties that we compare different schemes.

6.6 Static Buddy Nominator

The simplest relay nomination scheme is to always choose the same node. Thus, the users would ignore the channel state information between other users and the controller and itself with the other users. However, this could be an extremely simple one to implement. One would expect its performance to be bad as it completely ignores all side information and the performance is as expected. We simulated this scheme and for a network size of 30 users and one controller, message size 20bytes, cycle of time of 2ms and 20MHz bandwidth with nominal SNR of 5dB and decoding threshold of 0dB, nodes moving at a speed of 10m/s, the failure rate – the rate at which messages did not reach its destination through **single relay-only route** is a whopping 27%. If we nominate around 16 relays, the failure rate will go down to our desired error rate of 10^{-9}. However, this is too much!

6.7 Static Channel Model Based Nominator

The simplest scheme that considers the side information of channel state is the static channel model based selector. In this scheme, nodes assume that the channels between different nodes remain static throughout the duration of the cycle and decide on the relay based on the static channel model assumption. Each node s_i, considers the channel between itself and the potential relays h_{s_i,s_j}, where $j \neq i$ and the potential relays and the controller h_{c,s_j}, where $j \neq i$. Then it scores each relay j based on one of the two popular relay-scoring metrics

- **Min-energy**: The minimum of the two channel energies associated with a relay j i.e., the source-relay channel energy $|h_{s_i,s_j}|^2$ and the relay-destination channel energy $|h_{c,s_j}|^2$. This metric is the same as the metric used in the theoretical calculations in Sec. 6.2. The score of relay j is $\min(|h_{s_i,s_j}|^2, |h_{c,s_j}|^2)$.
- **Harmonic mean of energy**: The harmonic mean of the two channel energies associated with a relay j i.e., the source-relay channel energy $|h_{s_i,s_j}|^2$ and the relay-destination channel energy $|h_{c,s_j}|^2$. The score of relay j is $\frac{|h_{s_i,s_j}|^2 \cdot |h_{c,s_j}|^2}{|h_{s_i,s_j}|^2 + |h_{c,s_j}|^2}$. This metric also captures the quality of the metric based on the 'smallest' energy between the two channels as the smaller term dominates the harmonic mean.

We simulated this scheme (both metrics gave similar results) and for a network size of 30 users and one controller, message size 20bytes, cycle of time of 2ms and 20MHz bandwidth with nominal SNR of 5dB and decoding threshold of 0dB, nodes moving at a speed of 10m/s, the failure rate – the rate at which messages did not reach its destination through **single relay-only route** is 1%. This is a significant improvement over the 27% failure rate which suggests how useful side information can be.

6.8 Polynomial Channel Prediction Based Nominator

The simplest prediction scheme would be a simple polynomial-interpolator based prediction. We use channel knowledge from only a few recent samples to predict the channel quality and pick the best relay. We use a simple polynomial interpolator to fit a low-degree (at most 2) local polynomial on the past 4 channel coefficients to predict the future channel coefficient. We then use this predicted channel coefficient to get the channel quality. We use traditionally scoring metrics such as the relay with the maximum harmonic mean of source-relay and destination-relay channel energies or the relay with the maximum min-energy to select the best relay [182, 183, 180]. The results are equivalent. We simulated the simple polynomial channel coefficient predictor scheme and for a network size of 30 users and one controller, message size 20bytes, cycle of time of 2ms and 20MHz bandwidth with nominal SNR of 5dB and decoding threshold of 0dB, nodes moving at a speed of 10m/s, the failure rate – the rate at which messages did not reach its destination through **single relay-only route** is 0.1%.

6.9 Gaussian Prediction Based Nominator

As we model the channel fading process as a Gaussian process, it is natural to build a Gaussian predictor. The Gaussian predictor is actually really simple – its a linear combination of channel fades from the past. The weights for the linear combination can either be learned through training or can simply be obtained using Eq. 4.7.

6.9.1 Features from Gaussian model

Assume that we have m channel coefficients $\vec{h} = \begin{bmatrix} h_1 & h_2 & \dots & h_m \end{bmatrix}^T$ from the past at times $\vec{t} = \begin{bmatrix} t_1 & t_2 & \dots & t_m \end{bmatrix}^T$. We want to predict h_{m+1} at time t_{m+1} using the samples in $\vec{t}$. The linear predictor would be:

$$h_{m+1} = \vec{a}^T \vec{h} \tag{6.10}$$

where $\vec{a} = \begin{bmatrix} a_1 & a_2 & \dots & a_m \end{bmatrix}^T$. Let us model $\vec{a}$ assuming that we use the Gaussian process model as is. Let $\mathbf{K}$ be the covariance matrix of the channel coefficient corresponding to the times of observations so far given by

$$\mathbf{K} = \begin{bmatrix} k(v, t_1 - t_1) & k(v, t_2 - t_1) & \dots & k(v, t_m - t_1) \\ k(v, t_2 - t_1) & k(v, t_2 - t_2) & \dots & k(v, t_m - t_2) \\ \vdots & \vdots & \ddots & \vdots \\ k(v, t_m - t_1) & k(v, t_m - t_2) & \dots & k(v, t_m - t_m) \end{bmatrix}$$

where, $k(v,t) = J_0\left(\frac{2\pi}{\lambda_c} vt\right)/2$ (bessel function of the first kind as calculated in Eq. 4.6. Let $\mathbf{K}_*$ be the covariance matrix of channel coefficients corresponding to the future time of interest and the times of observations so far given by, $\mathbf{K}_* = \begin{bmatrix} k(v, t_{m+1} - t_1) & \dots & k(v, t_{m+1} - t_m) \end{bmatrix}$. Then, the linear coefficient vector $\vec{a}$ is given by

$$\vec{a} = \mathbf{K}_* \mathbf{K}^{-1} \tag{6.11}$$

The above equations assume that the relative velocity between different nodes is known and is some constant v.

6.9.2 Features from training

Let us consider the scenario where we learn $\vec{a}$ using training data. The problem then reduces to a simple linear regression problem. If we have training set $\mathbf{H}_I, \mathbf{H}_Q, \vec{h}^I_{m+1}, \vec{h}^Q_{m+1}$ where $\mathbf{H}_I$ and $\mathbf{H}_Q$ are the matrices where each row corresponds to one instance of channel coefficients (in-phase and quadrature component respectively) and the corresponding entry in $\vec{h}^I_{m+1}$ and $\vec{h}^Q_{m+1}$ are the in-phase and quadrature component of the realization of the same channel at time t_{m+1}. If we put the in-phase and quadrature component together such that $\vec{h}_{m+1} = \begin{bmatrix} (\vec{h}^I_{m+1})^T & (\vec{h}^Q_{m+1})^T \end{bmatrix}^T$ and $\mathbf{H} = \begin{bmatrix} \mathbf{H}_I^T & \mathbf{H}_Q^T \end{bmatrix}^T$ then the problem can be formulated as

$$\min_{\vec{a}} ||e||^2 = ||\vec{h}_{m+1} - \mathbf{H}\vec{a}||^2$$

which yields the standard least-squares estimator.

We simulated both the schemes (Gaussian process based parameters as well as learned parameters). We use the past 3 channel coefficients to predict the future channel coefficient. We then use this predicted channel coefficient to get the channel quality. We use traditionally

scoring metrics such as the relay with the maximum harmonic mean of source-relay and destination-relay channel energies or the relay with the maximum min-energy to select the best relay [180]. The results are equivalent. For a network size of 30 users and one controller, message size 20bytes, cycle of time of 2ms and 20MHz bandwidth with nominal SNR of 5dB and decoding threshold of 0dB, nodes moving at a speed of 10m/s, the failure rate – the rate at which messages did not reach its destination through **single relay-only route** is 0.1% (similar to the simple polynomial predictor).

We do not see an improvement in performance using Gaussian linear predictor over a polynomial based predictor. Both of them are similar in flavor – using a linear combination of the channel coefficients to predict the future channel coefficients and energy. The polynomial predictor does not assume any model and simply extrapolates while the Gaussian predictor assumes that the relative velocities is known. Theoretically, the Gaussian predictor should pick a relay with fidelity of 10^{-6} as Fig. 6.4 would suggest. However, there the relative velocity was known exactly and here it is not! This is the key reason why although we modeled the process as a Gaussian process, the Gaussian predictor does not fare well in the highly-variable setting that we consider. Additionally, the scatter plot from Fig. 6.5 essentially shows us that there is a non-linear relationship between the channel qualities. Let us try to leverage that insight to build a more sophisticated predictor in addition to training the predictor using a *training set that is drawn from a highly-variable environment with uncertainty that is emblematic of real-world settings.*

6.10 Neural Network Prediction Based Nominator

We consider a simple neural network where we feed the channel quality as the features. We use the channel qualities as opposed to the channel coefficients (in-phase and quadrature) as we wanted to couple the features since it is the combination of in-phase and quadrature features that determines the quality of the channel. We consider a simple architecture with just three layers (as shown in Fig. 6.6) – an input layer, a hidden layer and an output layer. The input nodes are essentially the channel quality of the past few milliseconds (2ms data the case in Fig. 6.6). The output node is the quality metric of the channel ranging from $[0, 1]$ – the higher the number, the more confident the neural network is that this channel is good and the closer to zero, the more confident that the channel is bad. The hidden layer nodes compute based on the input nodes and biases.

We train the neural network in the following way. We generate simulated data using an arbitrary room configuration and an arbitrary number of scatterers. We then optimize for different parameters as follows. Let $\vec{x}_k$ be the k^{th}input to the neural network consisting of the channel qualities of interest. Let y_k be the associated output for the given channel quality. The output is 0 if the channel quality if below a threshold and it is 1 if the output quality is above the threshold. Following the standard neural network terminology [184] let the weight of the edge joining j^{th} node in layer l to the i^{th} node in layer $l+1$ be $W_{i,j}^{(l)}$. The input to the i^{th} node of layer $l+1$ is then given by $W_{i,j}^{(l)} z_j + b_i^{(l)}$ where z_j is the output from

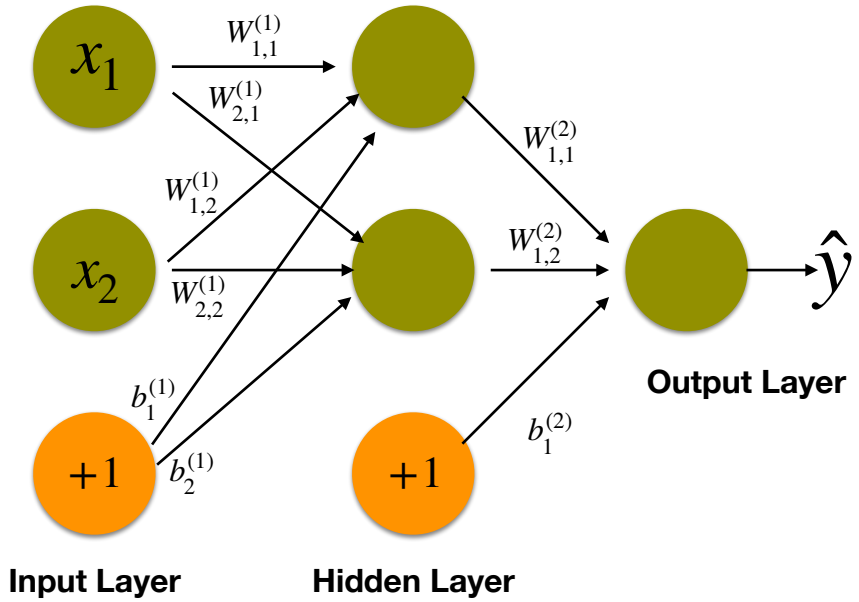


Figure 6.6: Architecture of the neural network for channel quality prediction.

node j of layer l and b_i is the bias. The hidden nodes and the output nodes all employ a sigmoid activation function over the appropriate sum of their inputs as mentioned earlier. Let there be m data points in the training set. Let the k^{th} output of the neural network be given by $\hat{y}_k$. Let the cost function to minimize be

$$\min \frac{1}{m} \sum_{k=1}^{m} (y_k - \hat{y}_k)^2 + \lambda \sum (W^{(l)}_{i,j})^2$$

where λ is a hyper-parameter over which we can search. We use the standard back-propagation based gradient-descent algorithm [185] with decaying learning rate along with hyper-parameter search to arrive at the parameters.

We tested the performance of the architecture on different rooms with different dimensions and number of scatterers. We predict the channel quality of various channels using this architecture follow similar process as earlier where we use the harmonic mean or the min of the two channel qualities to give each relay a score. For a network size of 30 users and one controller, message size 20bytes, cycle of time of 2ms and 20MHz bandwidth with nominal SNR of 5dB and decoding threshold of 0dB, nodes moving at a speed of 10m/s, the failure rate – the rate at which messages did not reach its destination through **single relay-only route** using the new neural network based architecture is 10^{-5}. This is a *massive* improvement over non-dynamics relay-selection schemes that have an error rate of 10^{-2}. We also used the architecture over noisy input (with measurement noise of 4dB) and the performance only degraded to 10^{-4}. This is still a significant improvement and essentially suggests that we need to employ 3 relays to get robust performance. Fig. 6.7 compares the performance of

the schemes considered so far along with the number of relays needed to provide a reliability of 10^{-9}.

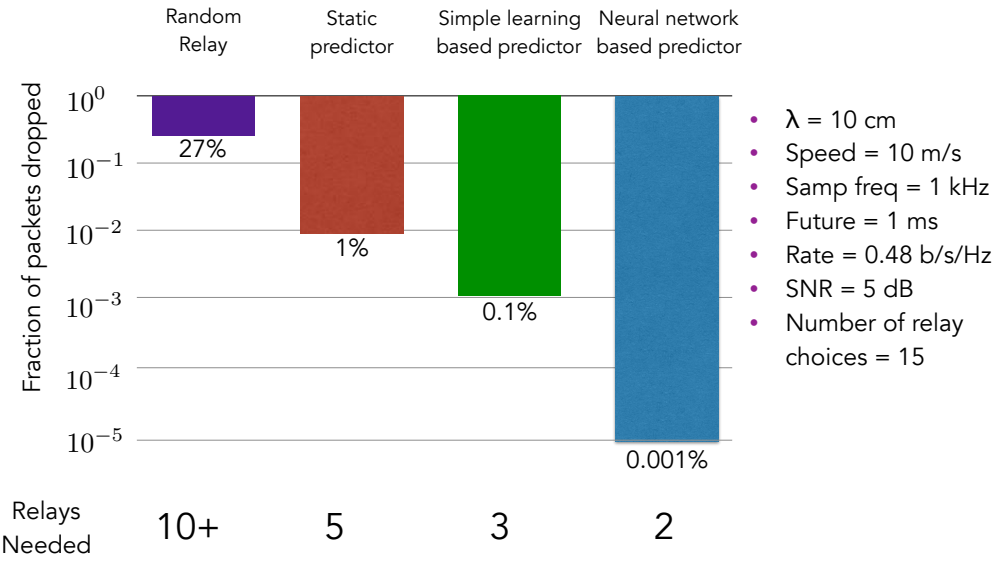


Figure 6.7: Comparison of various relay quality prediction schemes (assuming no measurement noise) and the number of relays needed to provide a reliability of 10^{-9}.

6.11 Selecting a relay versus selecting a set

In this chapter we explored the theoretical performance of selecting a single relay amongst a given set of relays under the Gaussian process model. We also looked at many channel quality predictors and their performance. We also assumed that selecting multiple relays should behave somewhat in an independent way and reduce the probability of set failure. Essentially, if selecting one relay out of k relays gives us error rate of p_k, does selecting d relays give us somewhere close to $p_k \cdot p_{k-1} \cdot p_{k-d+1}$ – essentially independent failure events but with progressively lower number of nodes to choose from? We explored this briefly and reserve detailed analysis for future work.

To answer this question, we can extend the framework from Eq. (6.9) in Sec. 6.2 where we were selecting only 1 relay. Consider the case where we select d relays from a set of k relays. Let the best relay (the relay with the smallest b_r) be i_1, the second best be i_2 and so on until i_d. Then, the probability that the *set is not good enough* is p_{set}, which is given by,

$$p_{set} = b_{i_1} \cdot b_{i_2} \cdot \ldots \cdot b_{i_d}. \tag{6.12}$$

Given we have already stated the importance of the parameters that determine the value of p_{set}, we will not explore some of those in much detail. In fact, we will restrict to a sampling frequency of 1000Hz (assuming nodes move at 10m/s) and the potential relay choices. We

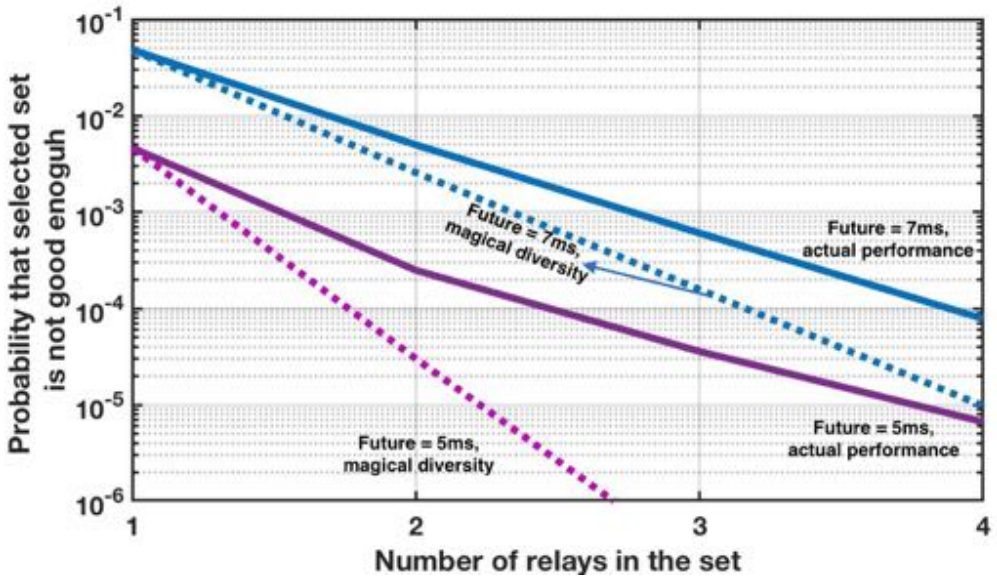


Figure 6.8: Probability of the selected relay not being good enough for varying future horizons. The model order is 3, the number of relays to choose from is $k = 9$, nodes are moving in a random direction at speed 10m/s, nominal SNR is 5dB and the center frequency is $f_c = 3$GHz. The dotted curves correspond to having a pure diversity like effect if each relay selection was from a new set of nodes every time but with a progressively smaller set of nodes.

will look at how the cardinality of the set of chosen relays i.e., d affects p_{set}. Fig. 6.8 shows how p_{set} actually scales (the solid curves) and how "magical diversity" would scale (the dotted curves). The solid lines diverge from the dotted lines to a non-trivial amount. This suggests that by selecting multiple relays the performance would not scale up by the same amount. But again, performance would not scale up to the same amount because there is an underlying outage that we simply cannot beat (at the given operating nominal SNR and rates of transmissions). The magical diversity line could go below 10^{-9} but given we are operating at outages of 10^{-9}, the magical diversity is actually non-attainable. Though this seems somewhat pessimistic on the first glance, it shows us that engineering the system to select the right number of relays needs to be thought out more carefully. Interestingly the performance of selecting 2 relays in our practical relay selection schemes scaled like the magical diversity. However, due to simulation limitations we were only able to see effects up to a certain degree as ultra-rare event simulation would require a more complex and lengthy simulation such as the ones studied in [186]. This suggests that it is very important to carefully consider selecting a set of relays both theoretically and through simulations.

6.12 Conclusions and Future Work

Predicting channel quality is important for selecting the right set of relays. Specifically in the context of URLLC, it is imperative to select a small set of relays with high confidence as it will reduce significant burden on implementation – especially synchronization. Since channel fading processes are not bandlimited, we concluded that looking at past few milliseconds of channel quality progress would provide a better model for channel quality in the future as opposed to the quasi-static channel mode. We leveraged these insights to build a robust neural-network based channel quality predictor that is able to pick the right relay upto an accuracy of 10^{-4} (even with measurement noise). This is a huge improvement over the quasi-static model which is incorrect at the rate of 10^{-2} despite having perfect measurement. Therefore, using these robust prediction schemes we need to nominate 3 relays to get to ultra-high reliability of 10^{-9} as opposed to needing greater than 10 relays.

In this thesis, we have only begun the exploration of using ideas from machine learning to predict channel quality. We assumed that nodes only had channel quality information but nothing about the relative velocity. However, in practical systems, it may be possible to estimate this relative velocity (say by tracking positions). In the presence of the knowledge of relative velocity, simple Gaussian process predictors may tremendously improve their performance. This is worth exploring. As mentioned earlier, the performance of selecting a set of relays as opposed to a single relay is not fully understood. An idea worth exploring is dividing the set of relays into disjoint subsets and nominating the best from each. How does the performance of such a nomination scheme compare to other ones that we have considered. Additionally, simulating more complex real-world like environment as well as testing these techniques on actual channel measurements is crucial. These are still open questions which will guide us to design robust wireless communication systems.

Chapter 7

Preliminary Experimental Results and Future Work

We have so far theoretically explored wireless channel models and their temporal and spatial behavior. In this chapter, we describe our experimental setup as well as our findings so far. The main purpose of the experiments is to understand the spatial and temporal characteristics of wireless channels as well as to understand the effects of line-of-sight (LOS) paths. The authors of [187] do a notable URLLC communication scheme evaluation which also primarily incorporates ideas of direct retransmissions and relay-based routing. However, their radios are primarily stationary.

7.1 Oscillator stability study

Our schemes, Occupy CoW and XOR-CoW depend on having scheduled transmissions. In fact, any scheme that aims to support URLLC must have scheduled transmissions as having random access will violate latency requirements. In order to have scheduled transmissions, nodes in the network need to keep time. Therefore, synchronization protocols (such as the ones described in [188, 189, 190]) become extremely important. Recent studies by the research as well as standards community [24, 25, 26] have identified the synchronization requirements as well as suggested techniques to achieve those requirements – for example, having the sources of synchronization signals be identified to then weight the signals accordingly.

However, most of these synchronization protocols have a node be the master clock and the rest of the network synchronize to that master clock. For this, the nodes need to listen to the synchronization signals from the master clock. However, due to fading, nodes may often not get the synchronization signal[1] (similar to how they may not get their packets).

[1]Even though they do not receive the synchronization signal from the master clock, they will still get secondary clock information from other nodes transmitting and keeping their clocks. Most synchronization protocols will take advantage of that but we ignore it in this study to purely focus on oscillator drift.

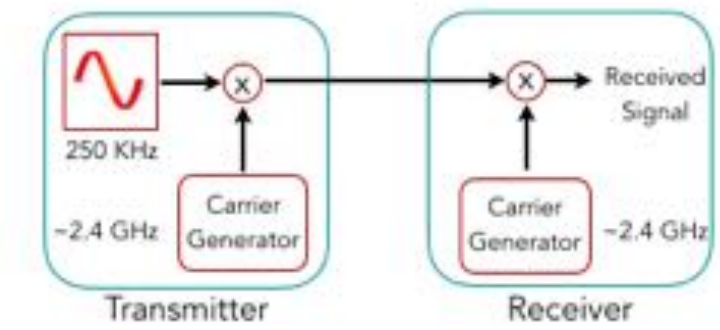


Figure 7.1: Schematic of the experiment for oscillator dynamics study

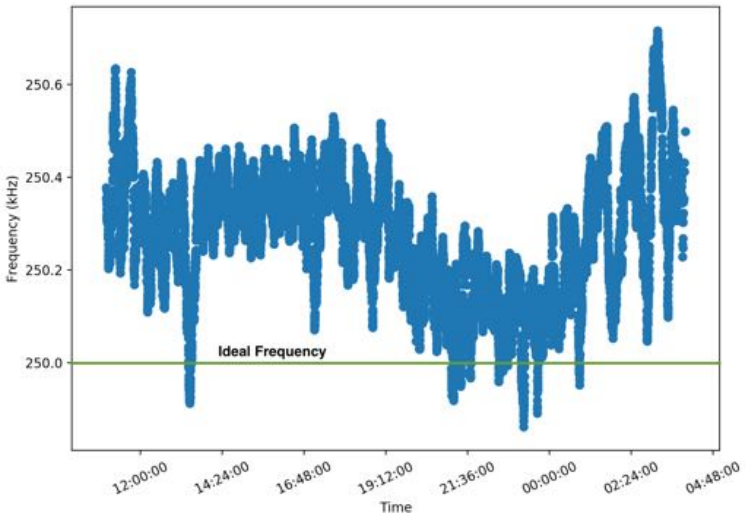


Figure 7.2: Instantaneous frequency at baseband as time progresses. The green line indicates the ideal frequency if oscillators were not off.

Their clocks may drift and if it drifts significantly, they might transmit at the wrong time and their transmission might collide with other packets. In order to study that, we look at how much an off-the-shelf clock's oscillator drifts.

Fig. 7.1 shows the schematic of the experiment. The transmitter and the receiver are connected by a wire (as we wanted to suppress any wireless channel induced artifacts). The transmitter transmits a tone of 250KHz modulated onto a carrier frequency of 2.4GHz and the receiver tracks the instantaneous frequency of the received signal at baseband.

Fig. 7.2 plots the instantaneous frequency at baseband over a period of 15 hours. As oscillators are imperfect (they may be oscillating at a frequency slightly off from what they ideally should be operating at), we expect the instantaneous frequency to be different from the ideal frequency of 250KHz and we see this behavior. Another finding is that the drift itself is at most 5Hz/s. This means that the drift itself is predictable and possibly corrected

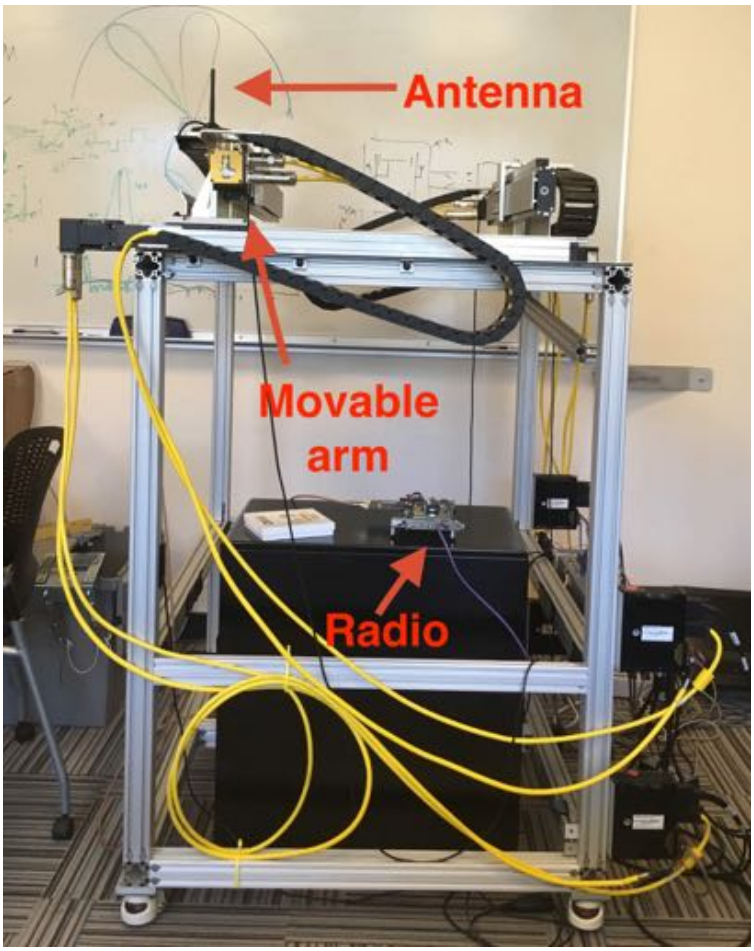


Figure 7.3: Receiver setup with antenna mounted on a movable arm.

quite easily through synchronization signal. However, the potentially troubling finding is that the maximum drift away from the expected frequency is 1KHz. This number is quite high and might potentially lead to a node transmitting at the incorrect time leading to a collision. This effect could be further exaggerated if nodes are moving due to Doppler. This indicates that studying drifts in real-world settings is essential to designing guard bands and other safety mechanisms to prevent packet collisions and other bad events.

7.2 Channel measurement study

In this thesis, we have emphasized the importance of studying the dynamics of wireless channels for designing systems that meet the demands of URLLC. We found that fading processes are not bandlimited and therefore the assumption that the channel quality remains predictably constant is not valid. However, these findings were through simulations. Although we have confidence that the model captures the essence of real-world physics well,

it is still not real-world data. To get actual channel measurements, we did the following experiment.

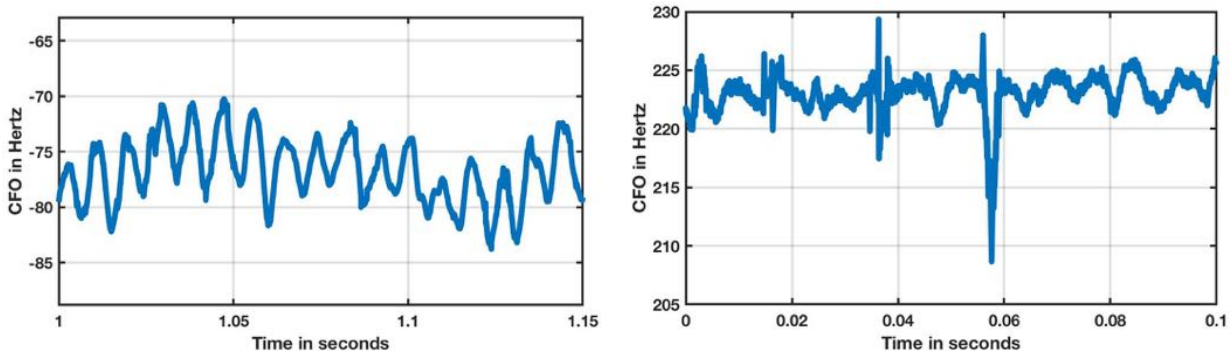
The setup is in a room of size about 15 × 10sq ft (shown in Fig. 7.3). In the setup, we have a static transmitter and a mobile receiver. The static transmitter is kept on a fixed platform. The receiver (more precisely the receive antenna) is on a slightly more complex setup. To capture the spatial variation of channels as accurately as possible, we use a linear actuator based table (henceforth referred to as XY table) with motion controlled setup using Parker 6K controllers [191]. The XY table has a movable arm which can be controlled (upto an accuracy of 100μm) through the 6K controllers. The receive antenna is affixed to the movable arm such that it moves as precisely as the movable arm. The antenna is connected to an FPGA which along with the antenna makes up the receiver.

The transmitter transmits a gold code of length 4095 in loop. The signal is then QPSK modulated onto the carrier – the 802.11g/n channel centered at 2.462 GHz (channel number 11). The signal rate at the transmitter is 2.5 mega samples per second. The receiver samples the signal at 2.5 mega samples per second. At the receiver we get the mixed down signal at baseband. In total, we collect about 16 million samples at the receiver – per collection. The post-processing pipeline is as follows:

- Autocorrelate the signal over N samples (depending on the kind of signal being sent) to find the center frequency offset (CFO) per sample.
- Correct for the CFO by rotating the received signal by the appropriate phasor.
- Partition the corrected signal into chunks of 4095 (the length of the gold code), take the FFT of this corrected signal and divide it by the FFT of the gold code to get the FFT of the channel response.
- Take the inverse FFT of the channel response to get an estimate for the time-domain response of the channel.

Although these steps seems quite straightforward, we learned several things about setting up the experiment correctly as well as the post-processing. First, having automatic gain control can make the CFO oscillate at the same frequency as the control loop. Fig. 7.4a shows how the CFO varies when the receiver is doing some sort of gain control. On the other hand, the CFO did not exhibit this consistent oscillatory behavior once gain control was turned off as seen in Fig. 7.4b.

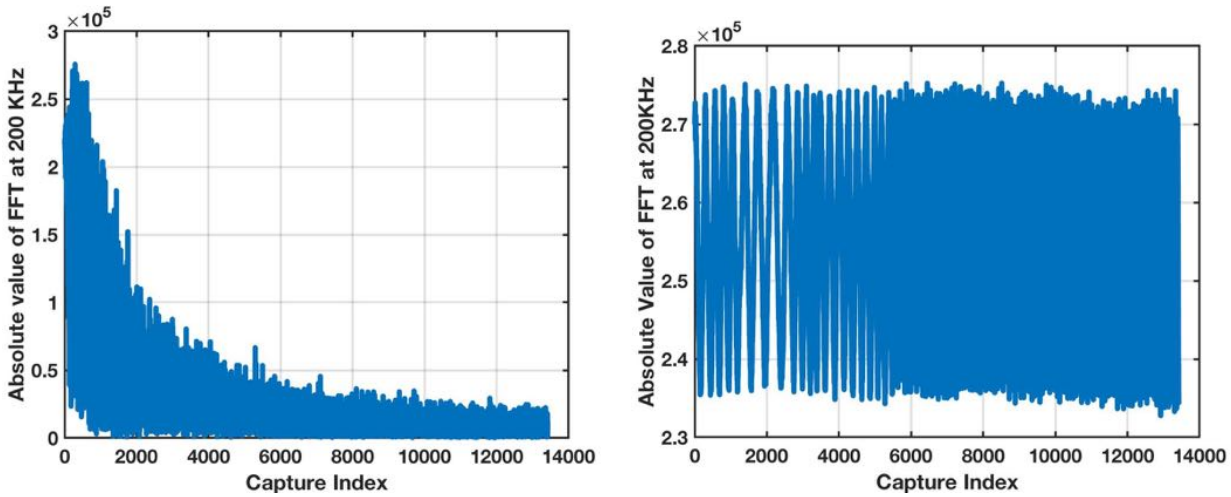
Second, while correcting for CFO, the effect of noise in the CFO as higher sample numbers should be taken into account. Why? The reason is quite simple. Lets say the sampling time is T_s, the CFO of m^{th} sample is f_m. This CFO value is noisy because it was estimated using noisy signal. The way we correct for the CFO is to multiply the sample $x[m]$ with the phasor $e^{-j2\pi T_s \cdot f_m \cdot m}$. For small values of m, the noise in the exponent $2\pi T_s \cdot f_m \cdot m$ due to noise in f_m is not amplified. However for large values of m, the noise in the exponent becomes too much and this manifests adversely. This effect is shown in Fig. 7.5a.



(a) CFO variation and oscillation due to gain control at the receiver

(b) CFO variation when gain control is turned off at the receiver

Figure 7.4: Effect of gain control on CFO



(a) Multiplicative noise in CFO correction term causing degradation at higher samples

(b) Batch processing for CFO correction negating the effect of multiplicative noise

Figure 7.5: Effect of multiplicative noise in CFO corrective

We transmitted a tone at 200KHz modulated onto a carrier of 2.462GHz (wirelessly). We followed the post-processing steps on the received signal and plot the absolute value of the fourier transform at 200KHz. Since the tone might actually appear at a slightly different frequency (as seen from Fig. 7.2), we first pick out the frequency which gives us the peak (which roughly corresponds to 200KHz) and plot the absolute value at that frequency throughout the capture. We see a decrease in the value because of the increase in noise due to the CFO correction at higher sample index.

How do we deal with this? A simple insight aids us here. We need to correct for CFO

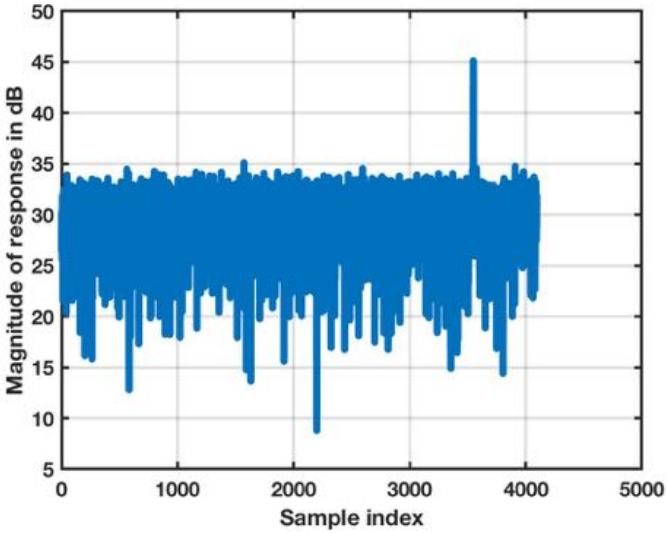


Figure 7.6: Channel response in the time-domain

in batches corresponding to the kind of signal we are looking at such that each new batch we can start the index at zero. For instance, when we transmit a gold-code signal, we need to consider batches of 4095 samples. And if we wanted to finely track the channel response (say sample by sample), this technique can readily be applied one a moving window basis as well. Using this technique, we can completely get rid of the effect of amplifying noise and the corresponding peak plot is shown in Fig. 7.5b.

Before exploring using the XY table, we look at how channel between a static transmitter and receiver varies. We transmit a gold code of length 4095 in a loop and look at the channel response in the time domain. In this setting, we did not do anything explicit to suppress line-of-sight path. When we looked at the time-domain channel response, we saw a very strong line-of-sight path. In fact, the values of the indexes neighboring the peak are very much at the 'noise' level. We saw this strong line-of-sight even when we tried blocking it using a simple barrier. This remains something that is yet unsolved and is currently an ongoing investigation. We have not yet been able to recreate the Bessel function or anything like that which could lead to a better knowledge of the channel dynamics. We will continue to pursue working on that first by suppressing the line-of-sight path and then looking at channel variation. It is imperative to explore what might lead to suppression of line-of-sight and how that effects channel dynamics.

In addition to the above mentioned explorations, to make any real-world URLLC application, it is crucial to test if we can get a reliability of 10^{-9}. In order to do that, we need to put together an experimental setup in which the radios nominate buddy nodes and they simultaneously transmit using a distributed-space-time-code. Testing to that accuracy is not trivial – the experiment run-time could be as long as a year. This calls for careful understanding of each component (say the error correction codes and the synchronization protocol) and

benchmarking each component and its performance in addition to intelligently designing an experiment that actually captures the events that may happen on a factory floor. All of these are open questions and addressing them will pave the way for making the grand vision of IoT a reality.

Appendix A

Detailed analysis of Occupy CoW protocol

In this appendix, we extend the union-bound for an arbitrary topology to include 3-hop success and then provide more nuanced calculations for the specialized star topology where we consider the downlink and uplink stages separately. A downlink failure occurs when at least one node fails to receive its message from the controller in the downlink stage and an uplink failure is vice-a-versa. The method of calculating the probability of error for uplink and downlink depends on the number of protocol hops. Finally, a union bound over the uplink and downlink phases is used to determine the overall 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. For the generic topology, we calculate the bound for a fixed schedule (and fixed transmission rate) while we consider the adaptive schedule protocol for the star topology.

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

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

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

$$F(n,p) = 1-(1-p)^n. \tag{A.2}$$

The probability of a good link has already been described in (B.3). 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. 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.

A.0.1 Union bound for 3-hop protocol:

The union bound analysis for a 2-hop protocol for a generic topology was provided in Sec. 3.3.3. In this section, we extend this to consider 3-hop successes. Consider a generic network with n nodes and s message streams. Let's say that each stream has one origin and on average d subscribers. For simplicity, the rates for all transmissions are kept constant at some rate R with a corresponding probability p of link failure as given by Eq. (B.3). Consider a single message-destination pair. Let each message get *three* shots at reaching its subscribers – directly from the source or through two or three hop relays. Then the probability of the message reaching any specific destination is

$$\begin{aligned} q_s &= \mathrm{P}(\text{success to a single destination}) \\ &= \mathrm{P}(\text{direct link}) \times \mathrm{P}(\text{success}|\text{direct link}) + \mathrm{P}(\text{no direct link}) \times \mathrm{P}(\text{success}|\text{no direct link}) \end{aligned} \tag{A.3}$$

The probability of a direct link $\mathrm{P}(\text{direct link}) = 1 - p$ (and $\mathrm{P}(\text{no direct link}) = p$) and the probability of success given there is a direct link $\mathrm{P}(\text{success}|\text{direct link}) = 1$. There are two ways for a node to succeed indirectly either by connecting to the set of nodes that heard the message directly from the source (let that set be $\mathcal{I}$) or if it did not connect to $\mathcal{I}$ then, by connecting to the set of nodes that heard the message from $\mathcal{I}$ (let this set be $\mathcal{J}$). If the node connects to $\mathcal{I}$, then the node succeeds in two hops (source $\rightarrow \mathcal{I} \rightarrow$ destination). If the node did not connect $\mathcal{I}$ but connects to $\mathcal{J}$, then the node succeeds in three hops (source $\rightarrow \mathcal{I} \rightarrow \mathcal{J} \rightarrow$ destination). The probability of success when there is no direct link is then given by

$\mathrm{P}(\text{success}|\text{no direct link})$

$$= \sum_{i=1}^{n-2} \left(B(n-2,i,p) \left\{ \left(1-p^i\right) + p^i \sum_{j=1}^{n-2-i} \left(B(n-2-i,j,p^i) \cdot \left(1-p^j\right) \right) \right\} \right)$$

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

$$\mathrm{P}(\text{failure}) = s \times d \times (1 - q_s). \tag{A.4}$$

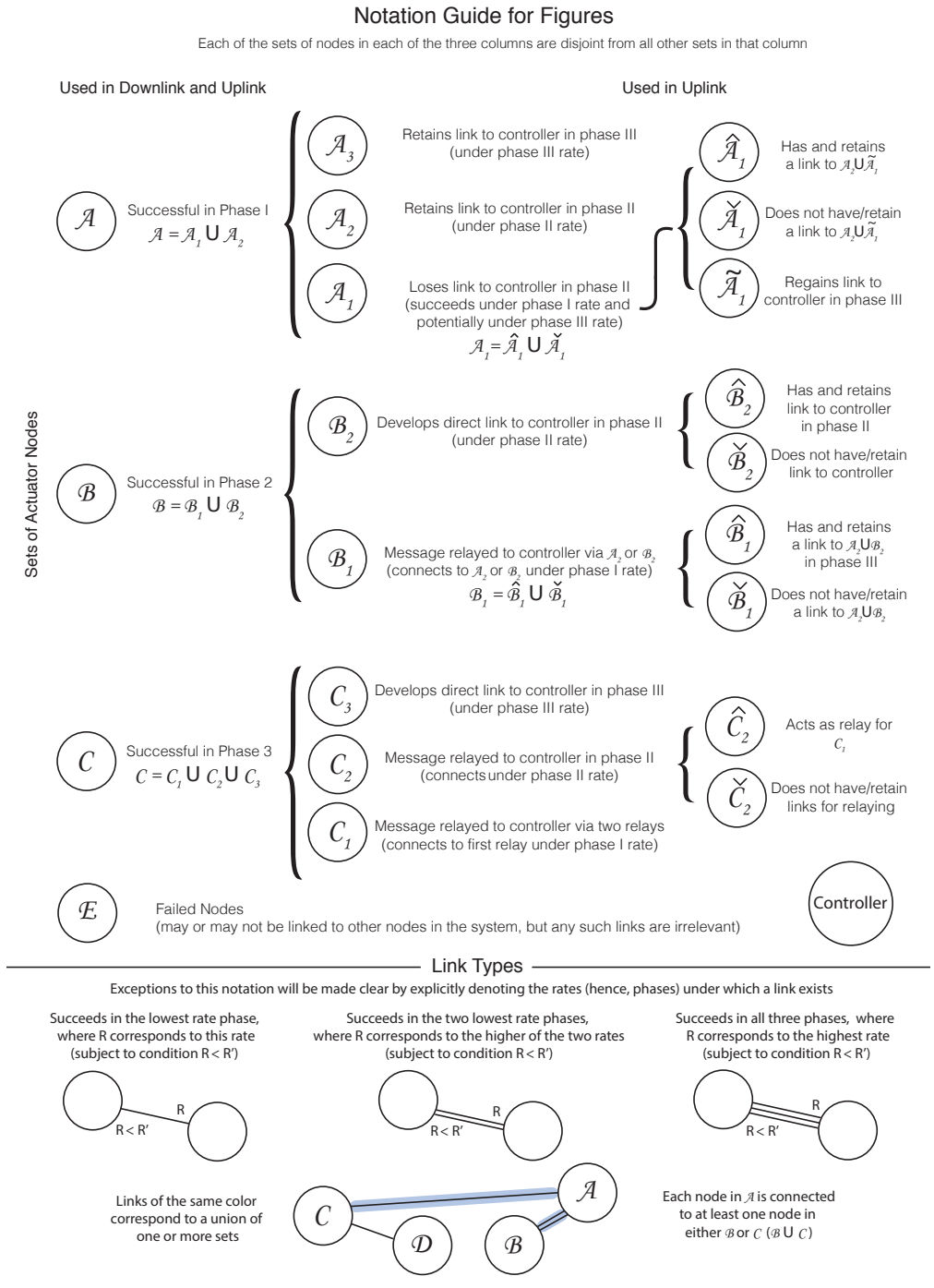
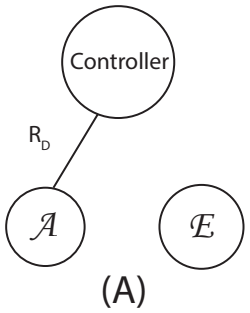
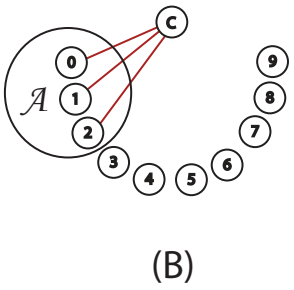
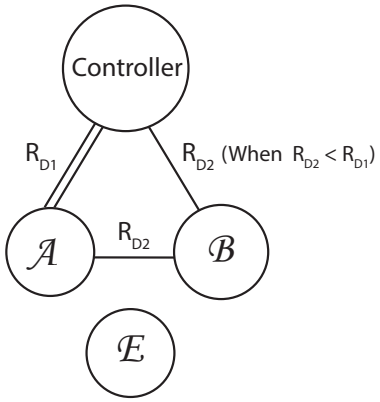


Figure A.1: This figure enumerates the various sets that we will be using throughout the analysis. In addition, how we represent various links in each of the protocol figures is also found here.





(a) We denote the set of nodes that have a direct link to the controller by $\mathcal{A}$. A node fails in one hop if it is not in Set $\mathcal{A}$. This case is the same for both downlink and uplink, but the rates of transmission 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 $\mathcal{E}$.



(b) 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.

A.0.2 Star Topology Analysis

The crux of analysis for star-topology relies on partitioning each stage of the protocol into a number of distinct states. As we saw when stepping through Fig. 3.15, 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 derive the probabilities of error for the downlink and uplink stages for each.

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. A.1 for both uplink and downlink.

A.0.3 One-Hop Protocol:

Recall that in this framework the entire protocol consists of stages 1 and 2 of Fig. 3.15. The controller broadcasts messages, each of length m bits for each node, to the n nodes, and the

nodes respond by transmitting their information as in Fig. 3.15. 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.

A.0.3.1 One-Hop Downlink

Theorem 3. *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. (B.3). The probability of cycle failure is then*

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

Proof. The rate of transmission is $R_D = \frac{m \cdot n}{T_D}$. Hence, following Eq. (B.3), 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. A.2a). Thus we get that the probability of failure for a one-hop downlink protocol is $P(\text{fail, 1D}) = F(n, p_D)$. □

A.0.3.2 One-Hop Uplink

Theorem 4. *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. (B.3). The probability of cycle failure is then*

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

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. A.2a). Thus we get $P(\text{fail, 1U}) = F(n, p_U)$. □

A.0.4 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. 3.15 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 conservative estimate of the total probability of cycle failure.

A.0.4.1 Two-Hop Downlink

Theorem 5. *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. (B.3). 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. (B.3) (the superscript* (a) *is to indicate the dependence on* a*). The probability of downlink failure is then*

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

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 set $\mathcal{A}$ 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. A.2b, 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 $R_{D_2}^{(a)} = \frac{m \cdot (n-a)}{T_{D_2}} + \frac{2n}{T_{D_2}}$, 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. (B.3)). 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[1] $p_{con}^{(a)} = P\left(R_{D_2}^{(a)} > C | R_{D_1} > C\right) = \min\left(\frac{p_{D_2}^{(a)}}{p_{D_1}}, 1\right)$.

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

[1] 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. (B.3) to get the final expression.

$\left(p_{D_2}^{(a)}\right)^a \cdot p_{con}^{(a)}$ 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 the probability that at least one of the remaining $n-a$ is unable to connect to the controller can be expressed with Eq. (B.2) 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:

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

□

A.0.4.2 Two-Hop Uplink

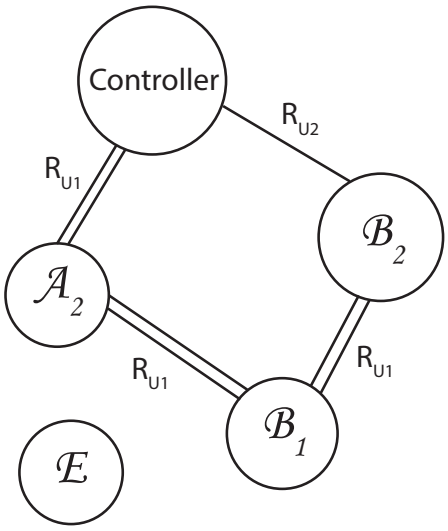
Theorem 6. *Let the Phase I uplink time be T_{U_1}, the Phase II uplink time be T_{U_2}, the number of non-controller nodes be n and the message size be m. The Phase I transmission rate is given by $R_{U_1} = \frac{(m+1)\cdot n}{T_{U_1}}$, and the corresponding probability of a single link failure, p_{U_1}, is given by Eq. (B.3). 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. (B.3). The probability of cycle failure is then*

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

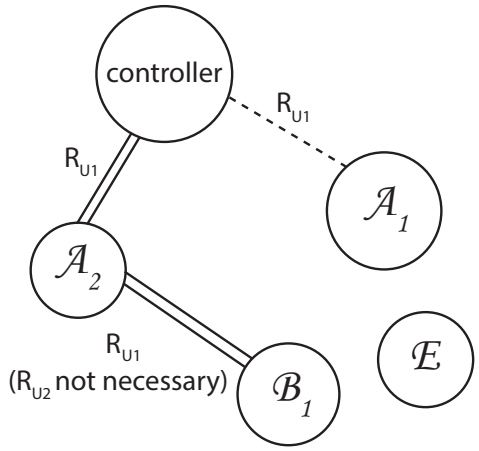
where,

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

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



(a) This figure depicts the only ways to succeed in two-hop uplink, given that $R_{U_2} < R_{U_1}$. They are: to have a direct connection to the controller under any of the two rates, or to have connected, in phase I (double lines), to a node that can succeed via a direct link to the controller.



(b) This figure depicts the possible means of success in a two-hop uplink protocol when $R_{U_2}^{(a)} \geq R_{U_1}$. The paths are: only having a direct link to the controller under R_{U_1} (dashed line), having a direct link under R_{U_1} & $R_{U_2}^{(a)}$ (double lines) to either the controller or one of the nodes who retained their link to the controller under $R_{U_2}^{(a)}$.

phase, T_{U_1} and is equal to $\frac{(m+1)\cdot n}{T_{U_1}}$. 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: $R_{U_2}^{(a)} \geq R_{U_1}$

If the second phase rate is higher, the means of success can be as depicted in Fig. A.3b. We will now derive the probability of error for this case. When $R_{U_2}^{(a)} \geq R_{U_1}$, 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 $R_{U_2}^{(a)} \geq R_{U_1}$, 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[2] $q^{(a)} = P(C < R_{U_2}^{(a)} | C > R_{U_1}) = \frac{p_{U_2}^{(a)} - p_{U_1}}{1 - p_{U_1}}$. 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. A.3b. This 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.

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

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. (B.2), 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. (B.1), 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 $M_U = n - a$.

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[3] $\tilde{q}^{(a)} = P\left(R_{U_2}^{(a)} < C | R_{U_1} > C\right) =$

[2] 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. (B.3) to get the final expression.

[3] 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. (B.3) to get the final expression.

$1 - \frac{p_{U_2}^{(a)}}{p_{U_1}}$. 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. A.3a 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 no node in $\mathcal{A}$ loses 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. A.3a).

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.

$$P(\text{fail 2U, case 2}) = P(\text{fail 2U } | \text{case 2}) \cdot P(\text{case 2}) = \sum_{a=a_0}^{n-1} P(\text{fail 2U} | A_2 = a) \cdot P(A_2 = a)$$

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

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 at least one node failed to reach the controller after two-hops can be expressed with Eq. (B.2) 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. (B.1) as $P(B_2 = b_2 | A_2 = a) = B\left(M_U, b_2, 1 - \widetilde{q}^{(a)}\right)$. 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.

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

where $M_U = n - a$. The probability of failure of the two-hop uplink protocol is then given by Eq. (A.10), where the first term comes from case 1, and the second is from case 2.

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

where $M_U = n - a$. □

A.0.5 Three-Hop Protocol

The failed protocol depicted in Fig. 3.15 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.

A.0.5.1 Three-Hop Downlink

Theorem 7. *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. (B.3). 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. (B.3). The probability 3-hop downlink failure is then*

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

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

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. A.4 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} (Set $\mathcal{A}$).
- A node can succeed in phase II of the protocol by either directly connecting to the controller under the new rate, $R_{D_2}^{(a)}$, or by connecting to 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 $R_{D_3}^{(a)} < \min(R_{D_2}^{(a)}, R_{D_1})$).

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 $\mathcal{B}$).

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

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. (B.2) 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. (B.3)). This gives us $B(n,a,p_{D_1})$. Therefore, the probability of failure of the 3-phase downlink protocol is given by

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

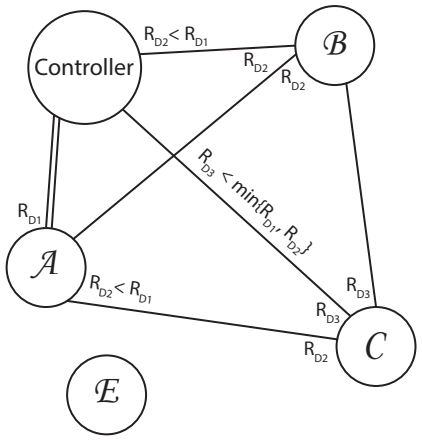


Figure A.4: The only ways to succeed in a three-hop downlink protocol are displayed. A node can succeed in the first phase directly from the controller, in Phase II from either the controller or someone who succeeded in Phase I, and in Phase III from someone who succeeded in Phase II. Please refer to Fig. A.1 to recall the exact meaning of each set name.

where $M_D = n - a$. □

A.0.5.2 Three-Hop Uplink

Theorem 8. *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 $R_{U_1} = \frac{(m+1)\cdot n}{T_{U_1}}$. The Phase II and Phase III transmission rate is given by $R_{U_2}^{(a)} = \frac{m\cdot(n-a)}{T_{U_2}}$, and $R_{U_3}^{(a)} = \frac{m\cdot(n-a)}{T_{U_3}}$ where a is the number of "successful nodes"*

in Phase I. The probability of cycle failure is then

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

where

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

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

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

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

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

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

$$
\begin{aligned}
P(\textit{fail}_4) = {} & F\left(n-a-b_1, p_1^{\widehat{a}_1+\widehat{b}_1}\right) \times B(a_1, \widehat{a}_1, p_2^{a_3}) \times B\left(b_1, \widehat{b}_1, s_{21}[a_3, a_2]\right) \times B\left(n-a, b_1, p_1^{a_2}\right) \times \\
& \times B\left(a_2, a_3, r_{32}\right) \times B\left(a, a_2, r_{21}\right) \times B(n, a, p_1)
\end{aligned}
$$

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

$$P(fail_5) = F\left(n-a-b_1, p_1^{\widehat{a}_1+\widehat{b}_1}\right) \times B\left(a-\widetilde{a}_1-a_2, \widehat{a}_1, p_2^{\widetilde{a}_1+a_2}\right) \times B\left(b_1, \widehat{b}_1, s_{21}[a_2, a_2]\right) \times \\ \times B\left(n-a, b_1, p_1^{a_2}\right) \times B\left(a-a_2, \widetilde{a}_1, m_{312}\right) \times B\left(a, a_2, r_{21}\right) \times B(n, a, p_1)$$

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

$$P(fail_6) = F\left(n-a-b-c_2-c_3, p_1^{\widehat{b}_1+\widehat{c}_2}\right) \times B\left(c_2, \widehat{c}_2, s_{21}[a+c_3, a+c_3]\right) \times B\left(b_1, \widehat{b}_1, s_{21}[a_2, a_2]\right) \times \\ \times B(n-a-b_1, c_3, q_{31}) \times B(n-a-b-c_3, c_2, p_1^{a_1+c_3}) \times \\ \times B\left(n-a, b_1, p_1^{a_2}\right) \times B\left(a, a_2, r_{21}\right) \times B(n, a, p_1)$$

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

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

Proof. We will now deal with each case one-by-one to understand all the subtle effects that occur in the uplink case.

Case 1: $R_{U_1} \geq R_{U_2} > R_{U_3}$

The rate of transmission in Phase I, R_{U_1}, is determined by the time allocated for this phase,

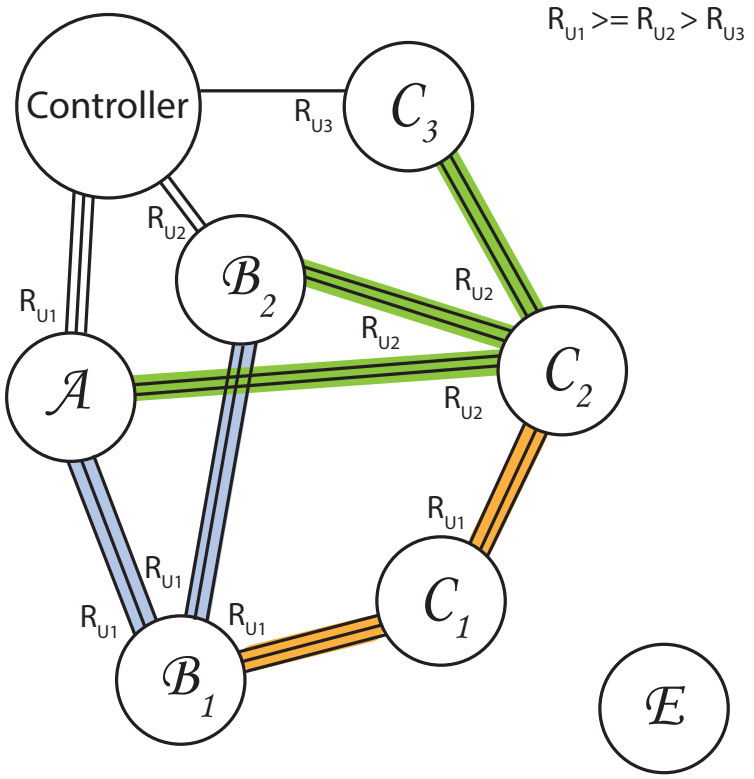


Figure A.5: Case 1: $R_{U_1} \geq R_{U_2} > R_{U_3}$. The only ways to succeed in the 1st case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to the controller or a node which can succeed in Phase II, and in Phase III by directly connecting to 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).

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, $R_{U_2}^{(a)} = \frac{m \cdot (n-a)}{T_{U_2}}$, $R_{U_3}^{(a)} = \frac{m \cdot (n-a)}{T_{U_3}}$. 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. A.5 displays an exhaustive list of ways to succeed in case 1 of the three-hop uplink protocol.

- A node can succeed directly to 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 $\mathcal{B}_2$).

- A node can succeed in the second phase of the protocol by connecting in the first phase (is 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.

- A node can succeed in the third phase of the protocol by connecting directly to the controller under the new rate, $R_{U_3}^{(a)}$ (is in set $\mathcal{C}_3$).

- A node can succeed in the third phase in a two-hop fashion by connecting to the set $\mathcal{A} \bigcup \mathcal{B}_2 \bigcup \mathcal{C}_3$ under the lower phase two rate $R_{U_2}^{(a)}$ (is in set $\mathcal{C}_2$). The set $\mathcal{A} \bigcup \mathcal{B}_2 \bigcup \mathcal{C}_3$ is the set of nodes which can connect to the controller in the third phase. Connecting to $\mathcal{A} \bigcup \mathcal{B}_2 \bigcup \mathcal{C}_3$ in phase II ensures that the message to be conveyed in phase III has been conveyed to the relays by phase II.

- A node can succeed in the third phase in a three-hop fashion by connecting to the set $\mathcal{C}_2 \bigcup \mathcal{B}_1$ in the first phase under rate R_{U_1} (is in set $\mathcal{C}_1$). The set $\mathcal{C}_2 \bigcup \mathcal{B}_1$ is the set of nodes which can connect to the set $\mathcal{A} \bigcup \mathcal{B}_2 \bigcup \mathcal{C}_3$ (they can connect to the controller in the third phase) in the second phase. Connecting to the set $\mathcal{C}_2 \bigcup \mathcal{B}_1$ 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 the three-hop uplink protocol, we will unroll the state space in a manner similar to the three-hop downlink derivations and 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 $\mathcal{A}$, $\mathcal{B} = \mathcal{B}_1 \bigcup \mathcal{B}_2$ and $\mathcal{C}_2 \bigcup \mathcal{C}_3$, is also not in $\mathcal{C}_1$ (fails given the instantiations of set $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$).

The probability of $A = a$ is exactly the same as we have seen before, as it relies on just point to point links to the controller, each of which fails independently with probability $p_1 = p_{U_1}$ (we use Eq. (B.3)). 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 $q_{21} = P(C < R_{U_2}^{(a)} | C < R_{U_1}^{(a)}) = (p_2)/(p_1)$. $\mathcal{B}_2$ is the set which can connect to the controller in the second phase. 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 $A = a$ and $B_2 = b_2$, 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 $A = a$, $B_2 = b_2$ and $B_1 = b_1$, we can calculate the probability of a node not being able to gain a connection to the controller in the third phase given there was no connection in the first two phases as $q_{32} = P(C < R_{U_3}^{(a)} | C < R_{U_2}^{(a)}) = (p_3)/(p_2)$. $\mathcal{C}_3$ is the set which can connect to the controller in the third phase. Hence we calculate the probability that $C_3 = c_3$ using a binomial distribution with parameter q_{32} as $B(n - a - b, c_3, q_{32})$.

Given $A = a$, $B_1 = b_1$, $B_2 = b_2$ and $C_3 = c_3$, we can calculate the probability of a node not succeeding in Phase III in two hops as $q_{21}^{a+b_2+c_3}$, as it must fail to connect to $\mathcal{A} \bigcup \mathcal{B}_2 \bigcup \mathcal{C}_3$ 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 $q_{21}^{a+b_2+c_3}$ as $B(n-a-b-c_3, c_2, q_{21}^{a+b_2+c_3})$. Given $A = a$, $B_1 = b_1$, $B_2 = b_2$, $C_3 = c_3$ and $C_2 = c_2$, the probability of a node (not in $\mathcal{A} \bigcup \mathcal{B}_1 \bigcup \mathcal{B}_2 \bigcup \mathcal{C}_2 \bigcup \mathcal{C}_3$) failing after three-hops is the probability that it cannot connect to $\mathcal{C}_2 \bigcup \mathcal{B}_1$ in the first phase. This is distributed Bernoulli $p_1^{b_1+c_2}$, and can be written with Eq. (B.2) as $F(n-a-b-c_2-c_3, p_1^{b_1+c_2})$.

Thus we have that given the realization $A = a$, the probability that the protocol fails under case 1: $R_{U_1} \geq R_{U_2} > R_{U_3}$ is given by

$$P(\text{fail}|\text{Case 1}, A = a) = \left(\sum_{b_2=0}^{n-a-1} \sum_{b_1=0}^{n-a-b_2-1} \sum_{c_3=0}^{n-a-b-1} \sum_{c_2=0}^{n-a-b-c_3-1} P(\text{fail}_1) \right)$$

where

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

Case 2: $R_{U_1} > R_{U_3} \geq R_{U_2}$

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, $R_{U_2}^{(a)} = \frac{m \cdot (n-a)}{T_{U_2}}$, $R_{U_3}^{(a)} = \frac{m \cdot (n-a)}{T_{U_3}}$. 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. A.6 displays an exhaustive list of ways to succeed in case 2 of the three-hop uplink protocol.

- A node can succeed directly to 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 (is 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 links to the controller in the third phase (set $\mathcal{A} \bigcup \widehat{\mathcal{B}}_2$) and $\check{\mathcal{B}}_1$ which does not have links

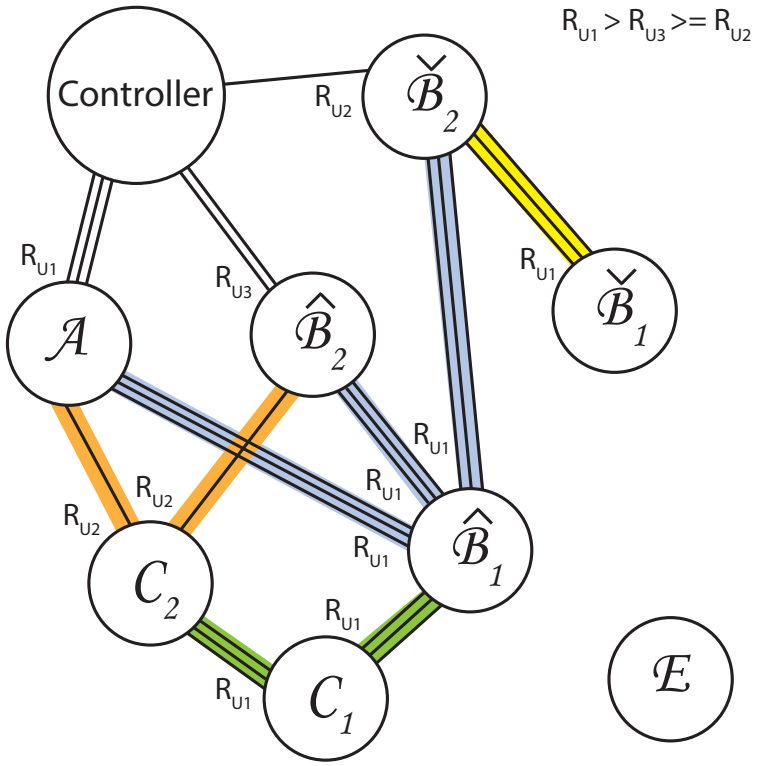


Figure A.6: Case 2: $R_{U_1} > R_{U_3} \geq R_{U_2}$. The only ways to succeed in the 2nd case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to the controller or a node which can succeed in Phase II, and in Phase III by connecting directly to 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).

to the set which has links to the controller in the third phase (set $\mathcal{A} \bigcup \widehat{\mathcal{B}}_2$). Thus, the 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 $\mathcal{A} \bigcup \widehat{\mathcal{B}}_2$ in phase II ensures that the message to be conveyed in phase III has been conveyed to the relays by phase II.

- 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 $\mathcal{A} \bigcup \widehat{\mathcal{B}}_2$ (they 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 the three-hop uplink protocol, we will again unroll the state space in a manner similar case 1 and 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 $\mathcal{A}$, $\mathcal{B} = \mathcal{B}_1 \bigcup \mathcal{B}_2$ and $\mathcal{C}_2$ is also not in $\mathcal{C}_1$.

The probability of $A = a$ is exactly the same as we have seen before, as it relies on just point to point links to the controller, each of which fails independently with probability $p_1 = p_{U_1}$ (we use Eq. (B.3)). 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 $q_{21} = P(C < R_{U_2}^{(a)} | C < R_{U_1}^{(a)}) = (p_2)/(p_1)$. $\mathcal{B}_2$ is the set which can connect to the controller in the second phase. 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 $\check{\mathcal{B}}_2$ 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 $A = a$, $B_2 = b_2$, $\widehat{B}_2 = \widehat{b}_2$, 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 $A = a$, $B_2 = b_2$, $\widehat{B}_2 = \widehat{b}_2$, and $B_1 = b_1$ we can calculate the probability of a node in $\mathcal{B}_1$ being only connected to $\check{\mathcal{B}}_2$ in the second phase given it connected to the set $\check{\mathcal{B}}_2 \bigcup \widehat{\mathcal{B}}_2 \bigcup \mathcal{A}$ as $s_{22}[a + \hat{b}_2, a + b_2] = (1 - p_2^{a+\widehat{b}_2})/(1 - p_2^{a+b_2})$. Hence we calculate the probability that $\check{B}_1 = \check{b}_1$ 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 $A = a$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$, $B_2 = b_2$, $\widehat{B}_2 = \widehat{b}_2$, we can calculate the probability of a node not succeeding in Phase III in two hops as $q_{21}^{a+\widehat{b}_2}$, 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 $q_{21}^{a+\widehat{b}_2}$ as $B(n - a - b, c_2, q_{21}^{a+\widehat{b}_2})$. 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 $p_1^{\widehat{b}_1 + c_2}$, and can be written with Eq. (B.2) as $F(n - a - b - c_2, p_1^{\widehat{b}_1 + c_2})$.

Thus we have that given the realization $A = a$, the probability that the protocol fails 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

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

Case 3: $R_{U_3} \geq R_{U_1} > R_{U_2}$
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, $R_{U_2}^{(a)} = \frac{m \cdot (n-a)}{T_{U_2}}$, $R_{U_3}^{(a)} = \frac{m \cdot (n-a)}{T_{U_3}}$. 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. A.7 displays an exhaustive list of ways to succeed in case 3 of the three-hop uplink protocol.

- A node can succeed directly to the controller in the first hop under rate R_{U_1} (is in set $\mathcal{A}$). This set is then divided into two disjoint sets $\mathcal{A}_1$ (nodes which lose their link to the controller in phase 3) and $\mathcal{A}_3$ (nodes which retain link to the controller in phase 3) such that $\mathcal{A} = \mathcal{A}_1 \bigcup \mathcal{A}_3$.
- 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 $\mathcal{B}_2$).
- A node can succeed in the second phase of the protocol by connecting in the first phase (is 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 links to the controller in the third phase (set $\mathcal{A}_3$) and $\check{\mathcal{B}}_1$ which does not have links to the set which has links to the controller in the third phase (set $\mathcal{A}_3$). 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 $\mathcal{A}_3$ under the lower phase two rate $R_{U_2}^{(a)}$ (is in set $\mathcal{C}_2$). The set $\mathcal{A}_3$ is the set of nodes which can connect to the controller in the third phase. Connecting to $\mathcal{A}_3$ in phase II ensures that the message to be conveyed in phase III has been conveyed to the relays by phase II.
- 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 $\mathcal{A}_3$ (they 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.

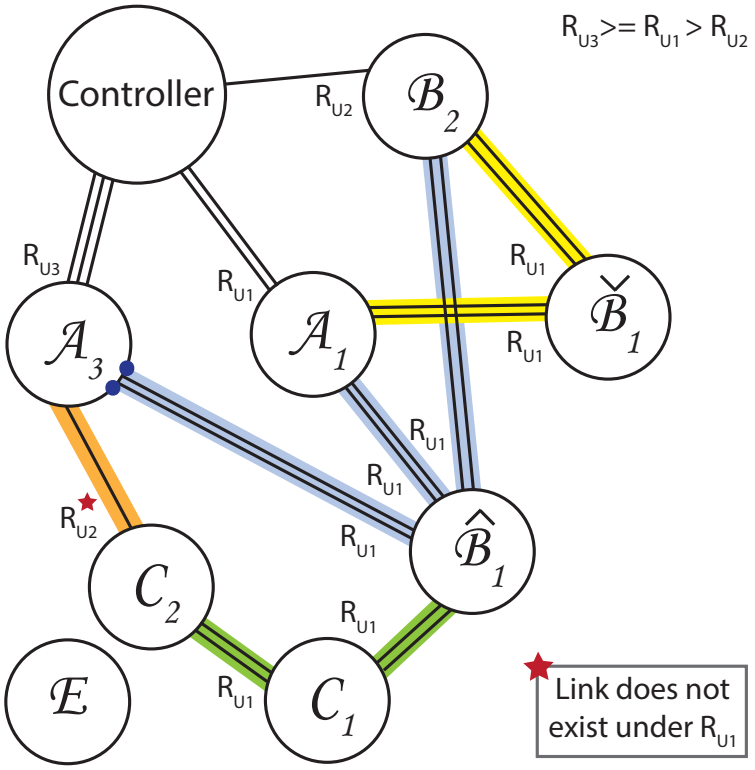


Figure A.7: Case 3: $R_{U_3} \geq R_{U_1} > R_{U_2}$. The only ways to succeed in the 3rd case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to the controller or a node which can succeed in Phase II, and in Phase III by connecting directly to 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).

To calculate the probability of error of the three-hop uplink protocol, we will unroll the state space as before and 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 $\mathcal{A}$, $\mathcal{B} = \mathcal{B}_1 \bigcup \mathcal{B}_2$ and $\mathcal{C}_2$ is also not in $\mathcal{C}_1$.

The probability of $A = a$ is exactly the same as we have seen before, as it relies on just point to point links to the controller, each of which fails independently with probability $p_1 = p_{U_1}$ (we use Eq. (B.3)). 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 $q_{21} = P(C < R_{U_2}^{(a)} | C < R_{U_1}^{(a)}) = (p_2)/(p_1)$. $\mathcal{B}_2$ is the set which can connect to the controller in the second phase. Hence we calculate the probability that $B_2 = b_2$ using a binomial distribution with parameter q_{21} as $B(n-a, b_2, q_{21})$. None of the nodes in the set $\mathcal{B}_2$ retain the link to the controller in phase 3. Given $A = a$ we can calculate the probability of a node in $\mathcal{A}$ losing connection to the controller in the third

phase as $r_{31} = P(C < R_{U_3}^{(a)} | C > R_{U_1}^{(a)}) = (p_3 - p_1)/(1 - p_1)$. This set is denoted as $\mathcal{A}_1$ and the set that retains the link is denoted as $\mathcal{A}_3$. Hence we calculate the probability that $A_3 = a_3$ using a binomial distribution with parameter r_{31} as $B(a, a_3, r_{31})$.

Given $A = a$, $A_3 = a_3$, $B_2 = b_2$, 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 $A = a$, $B_2 = b_2$, $A_3 = a_3$, and $B_1 = b_1$ we can calculate the probability of a node in $\mathcal{B}_1$ being only connected to $\mathcal{A}_3$ in the second phase given it connected to the set $\mathcal{A} \bigcup \mathcal{B}_2$ as $s_{22}[a_3, a + b_2] = (1 - p_2^{a_3})/(1 - p_2^{a+b_2})$. Hence we calculate the probability that $\check{B}_1 = \check{b}_1$ using a binomial distribution with parameter $s_{22}[a_3, a + b_2]$ as $B(b_1, \widehat{b}_1, s_{22}[a_3, a + b_2])$.

Given $A = a$, $A_3 = a_3$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$, $B_2 = b_2$, we can calculate the probability of a node not succeeding in Phase III in two hops as $q_{21}^{a_3}$, as it must fail to connect to $\mathcal{A}_3$ 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 $q_{21}^{a_3}$ as $B(n - a - b, c_2, q_{21}^{a_3})$. Given $C_2 = c_2$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$, $B_2 = b_2$, $A_3 = a_3$ 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 $p_1^{\widehat{b}_1 + c_2}$, and can be written with Eq. (B.2) as $F(n - a - b - c_2, p_1^{\widehat{b}_1 + c_2})$.

Thus we have that given the realization $A = a$, the probability that the protocol fails under case 3: $R_{U_3} \geq R_{U_1} > R_{U_2}$ is given by

$$P(\text{fail}|\text{Case 3}, A = a) = \left(\sum_{a_3=0}^{a} \sum_{b_2=0}^{n-a-1} \sum_{b_1=0}^{n-a-b_2-1} \sum_{\widehat{b}_1=0}^{b_1} \sum_{c_2=0}^{n-a-b-1} P(\text{fail}_3) \right)$$

where

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

Case 4: $R_{U_3} > R_{U_2} \geq R_{U_1}$

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, $R_{U_2}^{(a)} = \frac{m \cdot (n-a)}{T_{U_2}}$, $R_{U_3}^{(a)} = \frac{m \cdot (n-a)}{T_{U_3}}$. 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. A.8 displays an exhaustive list of ways to succeed in case 4 of the three-hop uplink protocol.

- A node can succeed directly to the controller in the first hop under rate R_{U_1} (is in set $\mathcal{A}$). This set is further divided into disjoint sets $\mathcal{A}_1$ (which lose connection to the

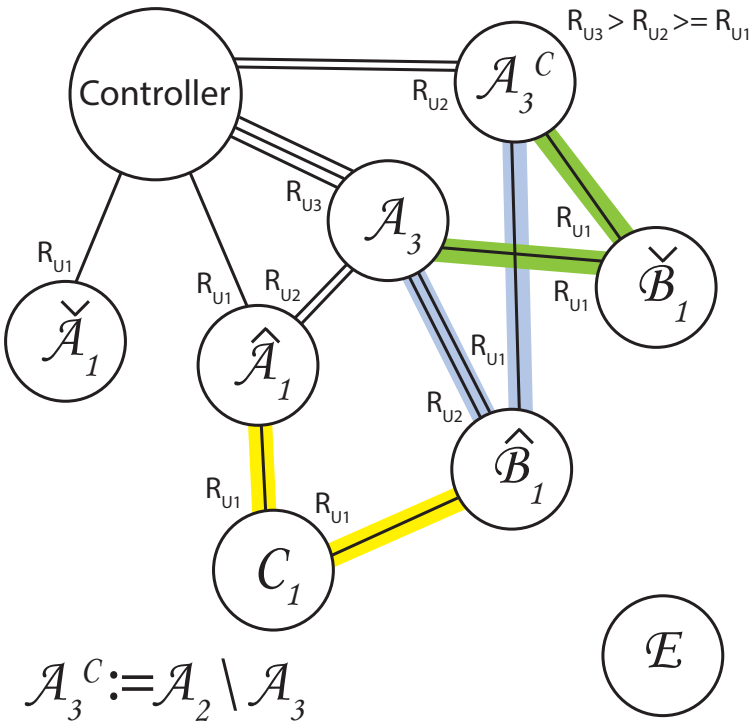


Figure A.8: Case 4: $R_{U_3} > R_{U_2} \geq R_{U_1}$: The only ways to succeed in the 4th case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to a node which can succeed in Phase II, and in Phase III by connecting via 2 hops to the nodes which have connections to the controller (thus succeeding in 3 hops).

controller after the first phase), $\mathcal{A}_3$ (the only set to retain connection to the controller in the third phase) and $\mathcal{A}_3^C$ (the set of nodes to retain connection to the controller in the second phase but not the third) such that $\mathcal{A} = \mathcal{A}_1 \bigcup \mathcal{A}_3 \bigcup \mathcal{A}_3^C$. Further, we divide $\mathcal{A}_1$ into disjoint sets $\widehat{\mathcal{A}}_1$ (the nodes which have a link to $\mathcal{A}_3$ in phase 2) and $\check{\mathcal{A}}_1$ (the nodes which do not have a link to $\mathcal{A}_3$ in phase 2) such that $\mathcal{A}_1 = \widehat{\mathcal{A}}_1 \bigcup \check{\mathcal{A}}_1$.

- A node can succeed in the second phase of the protocol by connecting in the first phase (is in set $\mathcal{B}_1$) to one of the nodes in the set $\mathcal{A}_2 = \mathcal{A}_3 \bigcup \mathcal{A}_3^C$ (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 links to the controller in the third phase (set $\mathcal{A}_3$) and $\check{\mathcal{B}}_1$ which does not have links to the set which has links to the controller in the third phase (set $\mathcal{A}_3$). Thus set $\check{\mathcal{B}}_1$ cannot act as relay for three-hop successes.

- A node can succeed in the third phase in a three-hop fashion by connecting to the set $\widehat{\mathcal{A}}_1 \bigcup \widehat{\mathcal{B}}_1$ in the first phase under rate R_{U_1} (is in set $\mathcal{C}_1$). The set $\widehat{\mathcal{A}}_1 \bigcup \widehat{\mathcal{B}}_1$ is the set of

nodes which can connect to the set $\mathcal{A}_3$ (they 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 space as before and 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 $\mathcal{B}_1$ is also not in $\mathcal{C}_1$.

The probability of $A = a$ is exactly the same as we have seen before, as it relies on just point to point links to the controller, each of which fails independently with probability $p_1 = p_{U_1}$ (we use Eq. (B.3)). This gives us $B(n, a, p_1)$. Given $A = a$ we can calculate the probability of a node in $\mathcal{A}$ losing connection to the controller in the second phase as $r_{21} = P(C < R_{U_2}^{(a)} | C > R_{U_1}^{(a)}) = (p_2 - p_1)/(1 - p_1)$. This losing link set is denoted as $\mathcal{A}_1$ and the set that retains the link is denoted as $\mathcal{A}_2$. Hence we calculate the probability that $A_2 = a_2$ using a binomial distribution with parameter r_{21} as $B(a, a_2, r_{21})$. Given $A = a$ and $A_2 = a_2$ we can calculate the probability of a node in $\mathcal{A}_2$ losing connection to the controller in the second 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 $\mathcal{A}_3^C$ and the set that retains the link is denoted as $\mathcal{A}_3$. Hence we calculate the probability that $A_3 = a_3$ using a binomial distribution with parameter r_{32} as $B(a_2, a_3, r_{32})$.

Given $A = a$, $A_2 = a_2$ and $A_3 = a_3$, we can calculate the probability of a node not succeeding in Phase II in two hops as $p_1^{a_2}$, as it must fail to connect to $\mathcal{A}_2$ in the first phase. Hence we calculate the probability that $B_1 = b_1$ using a binomial distribution with parameter $p_1^{a_2}$ as $B(n - a, b_1, p_1^{a_2})$. Given $A = a$, $A_2 = a_2$, $A_3 = a_3$, and $B_1 = b_1$ we can calculate the probability of a node in $\mathcal{B}_1$ being only connected to $\mathcal{A}_3^C$ in the second phase given it connected to the set $\mathcal{A}_2$ as $s_{21}[a_3, a_2] = (1 - p_2^{a_3})/(1 - p_1^{a_2})$. Hence we calculate the probability that $\check{B}_1 = \check{b}_1$ using a binomial distribution with parameter $s_{21}[a_3, a_2]$ as $B(b_1, \widehat{b}_1, s_{21}[a_3, a_2])$. Given $A = a$, $A_2 = a_2$ and $A_3 = a_3$, we can calculate the probability of a node in $\mathcal{A}_1$ being unable to connect to $\mathcal{A}_3$ in the second phase as $p_2^{a_3}$. The set of nodes being able to connect is denoted by $\widehat{A}_1$ and the probability that $\widehat{A}_1 = \widehat{a}_1$ is calculated using a binomial distribution with parameter $p_2^{a_3}$ as $B(a_1, -\widehat{a}_1, p_2^{a_3})$.

Given $A = a$, $A_2 = a_2$, $A_3 = a_3$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$, and $\widehat{A}_1 = \widehat{a}_1$, the probability of a node (not in $\mathcal{A} \bigcup \mathcal{B}_1$) failing after three-hops is the probability that it cannot connect to $\widehat{\mathcal{A}}_1 \bigcup \widehat{\mathcal{B}}_1$ in the first phase. This is distributed Bernoulli $p_1^{\widehat{a}_1 + \widehat{b}_1}$, and can be written with Eq. (B.2) as $F(n - a - b_1, p_1^{\widehat{a}_1 + \widehat{b}_1})$.

Thus we have that given the realization $A = a$, the probability that the protocol fails under case 4: $R_{U_3} > R_{U_2} > R_{U_1}$ is given by

$$P(\text{fail}|\text{Case 4}, A = a) = \left(\sum_{a_2=0}^{a} \sum_{a_3=0}^{a_2} \sum_{\widehat{a}_1=0}^{a-a_2} \sum_{b_1=0}^{n-a-1} \sum_{\widehat{b}_1=0}^{b_1} P(\text{fail}_4) \right)$$

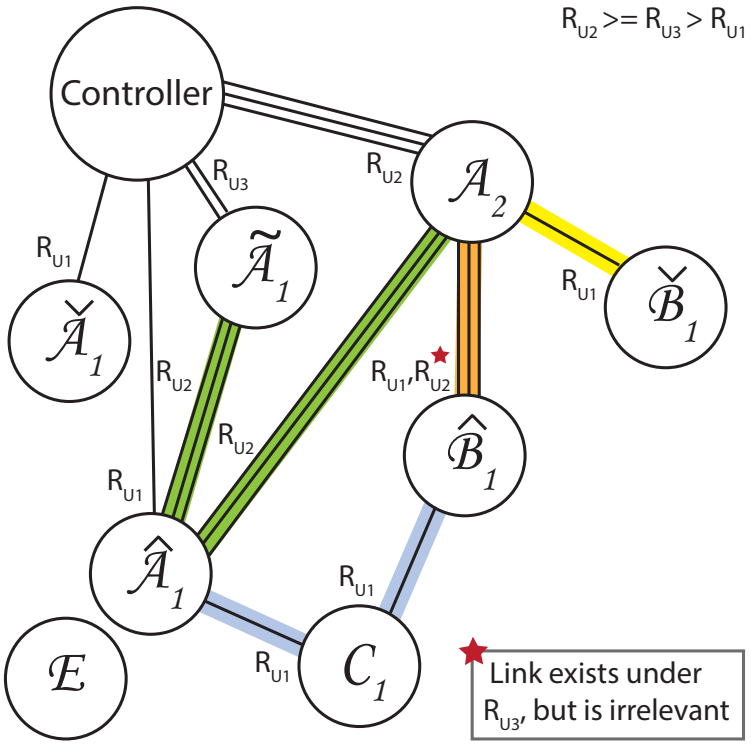


Figure A.9: Case 5: $R_{U_2} \geq R_{U_3} > R_{U_1}$: The only ways to succeed in the 5th case of three-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to a node which can succeed in Phase II, and in Phase III by connecting via 2 hops to the nodes which have connections to the controller (thus succeeding in 3 hops).

where

$$P(\text{fail}_4) = F\left(n-a-b_1, p_1^{\widehat{a}_1+\widehat{b}_1}\right) \times B(a_1, \widehat{a}_1, p_2^{a_3}) \times B\left(b_1, \widehat{b}_1, s_{21}[a_3, a_2]\right) \times B\left(n-a, b_1, p_1^{a_2}\right) \times \\ \times B\left(a_2, a_3, r_{32}\right) \times B\left(a, a_2, r_{21}\right) \times B(n, a, p_1)$$

Case 5: $R_{U_2} \geq R_{U_3} > R_{U_1}$
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, $R_{U_2}^{(a)} = \frac{m \cdot (n-a)}{T_{U_2}}$, $R_{U_3}^{(a)} = \frac{m \cdot (n-a)}{T_{U_3}}$. 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. A.9 displays an exhaustive list of ways to succeed in case 5 of the three-hop uplink protocol.

- A node can succeed directly to the controller in the first hop under rate R_{U_1} (is in set $\mathcal{A}$). This set is further divided into disjoint sets $\mathcal{A}_1$ (which lose connection to

the controller in the second phase) and $\mathcal{A}_2$ (the only set to retain connection to the controller in the second and third phase) such that $\mathcal{A} = \mathcal{A}_1 \bigcup \mathcal{A}_2$. Further, we divide $\mathcal{A}_1$ into disjoint sets $\widetilde{\mathcal{A}}_1$ (the nodes which gain back the link to the controller in the third phase), $\widehat{\mathcal{A}}_1$ (the nodes which have a link to $\mathcal{A}_2 \bigcup \widetilde{\mathcal{A}}_1$ in phase 2) and $\check{\mathcal{A}}_1$ (the nodes which do not have a link to $\mathcal{A}_2 \bigcup \widetilde{\mathcal{A}}_1$ in phase 2) such that $\mathcal{A}_1 = \check{\mathcal{A}}_1 \bigcup \widehat{\mathcal{A}}_1 \bigcup \check{\mathcal{A}}_1$.

- A node can succeed in the second phase of the protocol by connecting in the first phase (is in set $\mathcal{B}_1$) to one of the nodes in the set $\mathcal{A}_2$ (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 links to the controller in the third phase (set $\mathcal{A}_3 = \mathcal{A}_2 \bigcup \widetilde{\mathcal{A}}_1$) and $\check{\mathcal{B}}_1$ which does not have links to the set which has links to the controller in the third phase (set $\mathcal{A}_3$). Thus set $\check{\mathcal{B}}_1$ cannot act as relay for three-hop successes.

- A node can succeed in the third phase in a three-hop fashion by connecting to the set $\widehat{\mathcal{A}}_1 \bigcup \widehat{\mathcal{B}}_1$ in the first phase under rate R_{U_1} (is in set $\mathcal{C}_1$). The set $\widehat{\mathcal{A}}_1 \bigcup \widehat{\mathcal{B}}_1$ is the set of nodes which can connect to the set $\mathcal{A}_3$ (they 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 space as before and 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 $\mathcal{B}_1$ is also not in $\mathcal{C}_1$.

The probability of $A = a$ is exactly the same as we have seen before, as it relies on just point to point links to the controller, each of which fails independently with probability $p_1 = p_{U_1}$ (we use Eq. (B.3)). This gives us $B(n, a, p_1)$. Given $A = a$ we can calculate the probability of a node in $\mathcal{A}$ losing connection to the controller in the second phase as $r_{21} = P(C < R_{U_2}^{(a)} | C > R_{U_1}^{(a)}) = (p_2 - p_1)/(1 - p_1)$. This set is denoted as $\mathcal{A}_1$ and the set that retains the link is denoted as $\mathcal{A}_2$. Hence we calculate the probability that $A_2 = a_2$ using a binomial distribution with parameter r_{21} as $B(a, a_2, r_{21})$. Given $A = a$ and $A_2 = a_2$ we can calculate the probability of a node in $\mathcal{A}_1$ gaining back its connection to the controller in the third phase as $m_{312} = P(C < R_{U_3}^{(a)} | R_{U_1} C < R_{U_2}^{(a)}) = (p_3 - p_1)/(p_2 - p_1)$. This set is denoted by $\widetilde{\mathcal{A}}_1$ and the probability that $\widetilde{A}_1 = \widetilde{a}_1$ is calculated using a binomial distribution with parameter m_{312} as $B(a - a_2, \widetilde{a}_1, m_{312})$. The set $\mathcal{A}_3 = \mathcal{A}_2 \bigcup \widetilde{\mathcal{A}}_1$. Given $A = a$, $A_2 = a_2$ and $\widetilde{A}_1 = a_1$, we can calculate the probability of a node in $A_1 \setminus \widetilde{A}_1$ being unable to connect to A_3 in the second phase as $p_2^{a_2 + \widetilde{a}_1}$, as it must fail to connect to $\mathcal{A}_3$ in the second phase. The set that can connect is denoted by $\widehat{A}_1$ and the probability that $\widehat{A}_1 = \widehat{a}_1$ is calculated using a binomial distribution with parameter $p_2^{a_3}$ as $B(a - a_2 - \widetilde{a}_1, \widehat{a}_1, p_2^{a_3})$.

Given $A = a$, $A_2 = a_2$ and $A_3 = a_3$, we can calculate the probability of a node not succeeding in Phase II in two hops as $p_1^{a_2}$, as it must fail to connect to $\mathcal{A}_2$ in the first phase. Hence we calculate the probability that $B_1 = b_1$ using a binomial distribution with parameter $p_1^{a_2}$ as $B(n-a, b_1, p_1^{a_2})$. Given $A = a$, $A_2 = a_2$, $A_3 = a_3$, and $B_1 = b_1$ we can calculate the probability of a node in $\mathcal{B}_1$ being connected to $\mathcal{A}_2$ in the second phase given it connected to the set $\mathcal{A}_2$ in the first phase as $s_{21}[a_2, a_2] = (1 - p_2^{a_2})/(1 - p_1^{a_2})$. Hence we calculate the probability that $\check{B}_1 = \check{b}_1$ using a binomial distribution with parameter $s_{21}[a_2, a_2]$ as $B(b_1, \widehat{b}_1, s_{21}[a_2, a_2])$.

Given $A = a$, $A_2 = a_2$, $A_3 = a_3$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$, and $\widehat{A}_1 = \widehat{a}_1$, the probability of a node (not in $\mathcal{A} \bigcup \mathcal{B}_1$) failing after three-hops is the probability that it cannot connect to $\widehat{\mathcal{A}}_1 \bigcup \widehat{\mathcal{B}}_1$ in the first phase. This is distributed Bernoulli $p_1^{\widehat{a}_1 + \widehat{b}_1}$, and can be written with Eq. (B.2) as $F(n - a - b_1, p_1^{\widehat{a}_1 + \widehat{b}_1})$.

Thus we have that given the realization $A = a$, the probability that the protocol fails under case 5: $R_{U_2} \geq R_{U_3} > R_{U_1}$ is given by

$$P(\text{fail}|\text{Case 5}, A = a) = \left(\sum_{a_2=0}^{a} \sum_{\widetilde{a}=0}^{a-a_2} \sum_{\widehat{a}_1=0}^{a-a_2-\widetilde{a}_1} \sum_{b_1=0}^{n-a-1} \sum_{\widehat{b}_1=0}^{b_1} P(\text{fail}_5) \right)$$

where

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

Case 6: $R_{U_2} > R_{U_1} \geq R_{U_3}$

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, $R_{U_2}^{(a)} = \frac{m \cdot (n-a)}{T_{U_2}}$, $R_{U_3}^{(a)} = \frac{m \cdot (n-a)}{T_{U_3}}$. 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. A.10 displays an exhaustive list of ways to succeed in case 6 of the three-hop uplink protocol.

- A node can succeed directly to the controller in the first hop under rate R_{U_1} (is in set $\mathcal{A}$). This set is further divided into disjoint sets $\mathcal{A}_1$ (which lose connection to the controller in the second phase) and $\mathcal{A}_2$ (retains link to controller in the second phase) such that $\mathcal{A} = \mathcal{A}_1 \bigcup \mathcal{A}_2$.
- A node can succeed in the second phase of the protocol by connecting in the first phase (is in set $\mathcal{B}_1$) to one of the nodes in the set $\mathcal{A}_2$ (the set of nodes which can communicate to the controller in phase II). This ensures that the nodes which can

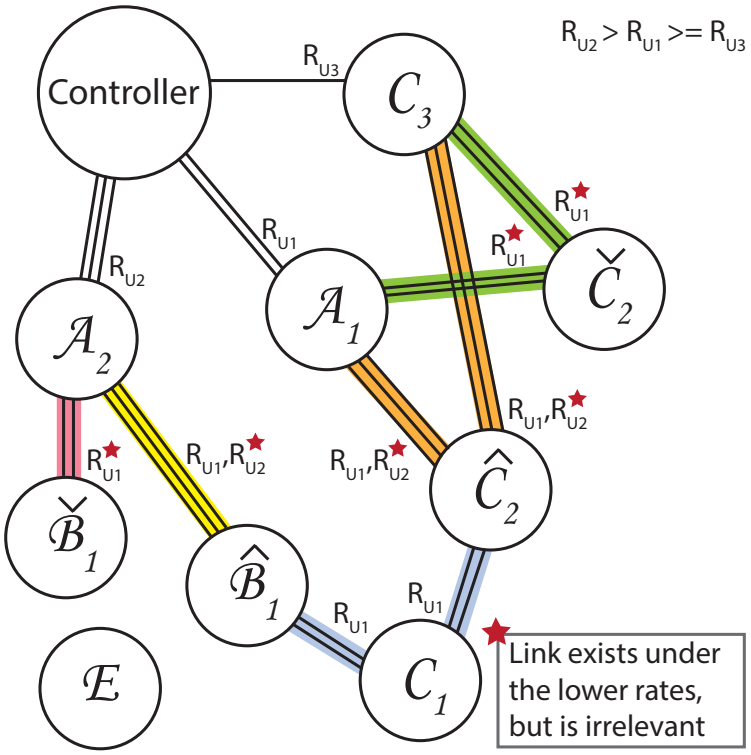


Figure A.10: Case 6: $R_{U_2} > R_{U_1} \geq R_{U_3}$: The only ways to succeed in the 6th case of 3-hop uplink protocol are displayed. A node can succeed in Phase I directly, in Phase II by connecting to a node which can succeed in Phase II, and in Phase III by directly connecting to the controller 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).

connect to the controller in the second phase already have the message. This set is then segregated into two disjoint sets: $\hat{\mathcal{B}}_1$ which has good links to the set which has links to the controller in the third phase (set $\mathcal{A}_2$) and $\check{\mathcal{B}}_1$ which does not have links to the set which has links to the controller in the third phase (set $\mathcal{A}_2$). Thus set $\check{\mathcal{B}}_1$ cannot act as relay for three-hop successes.

- A node can succeed in the third phase of the protocol by connecting directly to the controller under the new rate, $R_{U_3}^{(a)}$ (is in set $\mathcal{C}_3$).
- A node can succeed in the third phase in a two-hop fashion by connecting to the set $\mathcal{A}_1 \bigcup \mathcal{C}_3$ under the phase one rate of R_{U_1} (is in set $\mathcal{C}_2$). As the set they connect to is the set that doesn't have a connection to the controller in the second phase but does have connection in the third phase, these nodes, succeed in phase 3. These are further divided into disjoint sets $\hat{\mathcal{C}}_2$ (nodes that retain this link at the higher rate of $R_{U_2}^{(a)}$) and $\check{\mathcal{C}}_2$ (nodes that lose this link at the higher rate of $R_{U_2}^{(a)}$) such that $\mathcal{C}_2 = \check{\mathcal{C}}_2 \bigcup \hat{\mathcal{C}}_2$.

- A node can succeed in the third phase in a three-hop fashion by connecting to the set $\widehat{\mathcal{B}}_1 \bigcup \widehat{\mathcal{C}}_2$ in the first phase under rate R_{U_1} (is in set $\mathcal{C}_1$). The set $\widehat{\mathcal{C}}_2 \bigcup \widehat{\mathcal{B}}_1$ is the set of nodes which can connect to the set $\mathcal{A} \bigcup \mathcal{C}_3$ (they 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 the three-hop uplink protocol, we will unroll the state space as before and 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 $\mathcal{A}$, $\mathcal{B}_1$ and $\mathcal{C}_3 \bigcup \mathcal{C}_2$ is also not in $\mathcal{C}_1$.

The probability of $A = a$ is exactly the same as we have seen before, as it relies on just point to point links to the controller, each of which fails independently with probability $p_1 = p_{U_1}$ (we use Eq. (B.3)). This gives us $B(n, a, p_1)$. Given $A = a$ we can calculate the probability of a node in $\mathcal{A}$ losing connection to the controller in the second phase as $r_{21} = P(C < R_{U_2}^{(a)} | C > R_{U_1}^{(a)}) = (p_2 - p_1)/(1 - p_1)$. This set is denoted as $\mathcal{A}_1$ and the set that retains the link is denoted as $\mathcal{A}_2$. Hence we calculate the probability that $A_2 = a_2$ using a binomial distribution with parameter r_{21} as $B(a, a_2, r_{21})$.

Given $A = a$ and $A_2 = a_2$, we can calculate the probability of a node not succeeding in Phase II in two hops as $p_1^{a_2}$, as it must fail to connect to $\mathcal{A}_2$ in the first phase. Hence we calculate the probability that $B_1 = b_1$ using a binomial distribution with parameter $p_1^{a_2}$ as $B(n - a, b_1, p_1^{a_2})$. Given $A = a$, $A_2 = a_2$ and $B_1 = b_1$ we can calculate the probability of a node in $\mathcal{B}_1$ being only connected to $\mathcal{A}_2$ in the second phase given it connected to the set $\mathcal{A}_2$ as $s_{21}[a_2, a_2] = (1 - p_2^{a_2})/(1 - p_1^{a_2})$. Hence we calculate the probability that $\check{B}_1 = \check{b}_1$ using a binomial distribution with parameter $s_{21}[a_2, a_2]$ as $B(b_1, \widehat{b}_1, s_{21}[a_2, a_2])$.

Given $A = a$ and $B_1 = b_1$, we can calculate the probability of a node not being able to gain a connection to the controller in the third phase given there was no connection in the first two phases as $q_{31} = P(C < R_{U_3}^{(a)} | C < R_{U_1}^{(a)}) = (p_3)/(p_1)$. $\mathcal{C}_3$ is the set which can connect to the controller in the third phase. Hence we calculate the probability that $C_3 = c_3$ using a binomial distribution with parameter q_{31} as $B(n - a - b_1, c_3, q_{31})$.

Given $A = a$, $A_1 = a_1$, $C_3 = c_3$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$, we can calculate the probability of a node not succeeding in Phase III in two hops as $p_1^{a_1 + c_3}$, as it must fail to connect to $\mathcal{A}_1 \bigcup \mathcal{B}_3$ in the first phase. Hence we calculate the probability that $C_2 = c_2$ using a binomial distribution with parameter $p_1^{a_1 + c_3}$ as $B(n - a - b_1 - c_3, c_2, p_1^{a_1 + c_3})$. Given $C_2 = c_2$, $A_1 = a_1$ and $C_3 = c_3$, the probability of a node in $\mathcal{C}_2$ losing connection in the second phase is given by $s_{21}[a_1 + c_3, a_1 + c_3] = (1 - p_2^{a_1 + c_3})/(1 - p_1^{a_1 + c_3})$. Hence we calculate the probability that $\check{C}_2 = \check{c}_2$ using a binomial distribution with parameter $s_{21}[a_2, a_2]$ as $B(c_2, \widehat{c}_2, s_{21}[a_1 + c_3, a_1 + c_3])$.

Given $C_2 = c_2$, $\widehat{C}_2 = \widehat{c}_2$, $B_1 = b_1$, $\widehat{B}_1 = \widehat{b}_1$ and $A = a$, the probability of a node (not in $\mathcal{A} \bigcup \mathcal{B}_1 \bigcup \mathcal{C}_3 \bigcup \mathcal{C}_3$) failing after three-hops is the probability that it cannot connect to $\widehat{\mathcal{C}}_2 \bigcup \widehat{\mathcal{B}}_1$ in the first phase. This is distributed Bernoulli $p_1^{\widehat{b}_1 + \widehat{c}_2}$, and can be written with Eq. (B.2) as $F(n - a - b - c_2 - c_3, p_1^{\widehat{b}_1 + \widehat{c}_2})$.

Thus we have that given the realization $A = a$, the probability that the protocol fails under case 6: $R_{U_2} > R_{U_1} \geq R_{U_3}$ is given by

$$P(\text{fail}|\text{Case 6}, A = a) = \left(\sum_{a_2=0}^{a} \sum_{b_1=0}^{n-a-1} \sum_{\widehat{b}_1=0}^{b_1} \sum_{c_3=0}^{n-a-b_1-1} \sum_{c_2=0}^{n-a-b_1-c_3-1} \sum_{\widehat{c}_2=0}^{c_2} P(\text{fail}_6) \right)$$

where

$$\begin{aligned} P(\text{fail}_6) = F\left(n-a-b-c_2-c_3, p_1^{\widehat{b}_1+\widehat{c}_2}\right) \times B\left(c_2, \widehat{c}_2, s_{21}[a+c_3, a+c_3]\right) \times B\left(b_1, \widehat{b}_1, s_{21}[a_2, a_2]\right) \times \\ \times B(n-a-b_1, c_3, q_{31}) \times B(n-a-b-c_3, c_2, p_1^{a_1+c_3}) \times \\ \times B\left(n-a, b_1, p_1^{a_2}\right) \times B\left(a, a_2, r_{21}\right) \times B(n, a, p_1). \end{aligned}$$

□

Appendix B

Detailed analysis of XOR-CoW protocol

We analyze the XOR-CoW protocol by looking at all the ways at least one of the messages did not reach the destination within the cycle. We achieve this by partitioning the nodes into various sets which depend on various aspects like downlink/uplink success and the state of node-node as well as node-controller links in different phases. Before continuing with the analysis itself, we define some notation.

Notation

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

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

Note that the probability p is the probability of failure, not the probability of success. The probability of at least one out of n independent experiments failing will be denoted as

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

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, the assumption of Rayleigh fading tells us that the probability of an unsuccessful transmission is defined as

$$p = P(R > C) = 1 - \exp\left(-\frac{2^{R/W}-1}{\text{SNR}}\right). \tag{B.3}$$

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

Set Notation

We describe the various sets used in the analysis. Following general convention, 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. The sets being considered are:

- $\mathcal{A}$: the set of nodes successful in the downlink phase. Further divided into disjoint sets $\widetilde{\mathcal{A}}$ and $\check{\mathcal{A}}$ such that $\mathcal{A} = \widetilde{\mathcal{A}} \bigcup \check{\mathcal{A}}$.
  - $\widetilde{\mathcal{A}}$: the set of nodes which succeed in downlink as well as uplink phases. This is further partitioned into $\widetilde{\mathcal{A}}_X$ (the set which connects to the controller in the XOR phase) and $\widetilde{\mathcal{A}}_U$ (the set which cannot connect to the controller in the XOR phase).
  - $\check{\mathcal{A}}$: the set of nodes which do not succeed in uplink. This set is further partitioned into $\check{\mathcal{A}}_X$ (which can connect to the controller in the XOR phase) and $\check{\mathcal{A}}_D$ (which cannot connect to the controller in the XOR phase).
- $\mathcal{B}$: the set of nodes that weren't successful in downlink phase but were successful in uplink phase. Further partitioned into disjoint sets $\widetilde{\mathcal{B}}$ (has link to the controller in the XOR phase) and $\check{\mathcal{B}}$ (doesn't have link to controller in the XOR phase) such that $\mathcal{B} = \widetilde{\mathcal{B}} \bigcup \check{\mathcal{B}}$.
- $\mathcal{C}$: the set of nodes that succeed only in the XOR phase – both uplink and downlink successes happen in this phase. They can only succeed through relays.

Analysis of XOR-CoW:

Let the time allocated for the downlink phase be T_D, the uplink phase be T_U and the XOR phase be T_X such that $T_D + T_U + T_X = T$ where T is the given cycle time. If we chose to do fixed scheduling then the transmission rates for downlink, uplink and XOR phases are fixed at $R_D = \frac{m \cdot n}{T_D}$, $R_U = \frac{m \cdot n}{T_U}$ and $R_X = \frac{m \cdot n}{T_X}$ respectively. If adaptive scheduling scheme is employed, then the transmission rates for downlink, uplink and XOR phases are given by $R_D = \frac{m \cdot n}{T_D}$, $R_U = \frac{(m+1) \cdot n}{T_U}$ and $R_X = \frac{m \cdot (n - \widetilde{a})}{T_X}$ where $\widetilde{a}$ is the number of nodes that succeeded in both uplink and downlink phases. These $\tilde{\mathcal{A}}$ are called "strong nodes" and all the others need help. Without loss of generality we consider the flexible schedule scheme and proceed with the analysis. Depending on the time allocations for different phases and the number of strong nodes $\widetilde{a}$, we get the following theorem.

Theorem 9. *Let the time allocated for downlink, uplink and XOR phases be T_D, T_U and T_X respectively, the number of non-controller nodes be n, and message size be m bits. The downlink and uplink transmission rates are given by $R_D = \frac{m \cdot n}{T_D}$ and $R_U = \frac{(m+1) \cdot n}{T_U}$ respectively. The corresponding probability of a single link failure, p_D & p_U, is given by Eq. (B.3). The XOR phase transmission rate is given by $R_X^{\widetilde{a}} = \frac{m \cdot (n - \widetilde{a})}{T_X}$ where $\widetilde{a}$ is the number of "strong nodes" in both downlink and uplink phases and the corresponding probability of a single failure p_X, is given by Eq. (B.3). The probability XOR-CoW failure is then*

$$
\begin{aligned}
P(fail) = \sum_{a=0}^{n} \Bigg[& \sum_{b=0}^{n-a} P\left(fail_1\right) \mathbb{1}\left(R_D \geq R_U > R_X\right) + \sum_{b=0}^{n-a} \sum_{\widetilde{b}=0}^{b} P\left(fail_2\right) \mathbb{1}\left(R_D > R_X \geq R_U\right) \\
& + \sum_{\widetilde{a}=0}^{a} P\left(fail_3\right) \mathbb{1}\left(R_U \geq R_D > R_X\right) + \sum_{\widetilde{a}=0}^{a} \sum_{\check{a}_X=0}^{a-\widetilde{a}} P\left(fail_4\right) \mathbb{1}\left(R_U > R_X \geq R_D\right) \\
& + \sum_{\widetilde{a}=0}^{a} \sum_{\widetilde{a}_X=0}^{\widetilde{a}} P\left(fail_5\right) \mathbb{1}\left(R_X \geq R_U > R_D\right) + \sum_{\widetilde{a}_X=0}^{a} \sum_{b=0}^{n-a} P\left(fail_6\right) \mathbb{1}\left(R_X > R_D \geq R_U\right) \Bigg]
\end{aligned}
$$

where, $P\left(fail_1\right) = B(n, a, p_D) \times B(n-a, b, q_{UD}) \times F(n-a-b, p_U^a)$
is the probability of failure if the relationship between the rates is $R_D \geq R_U > R_X$,

$$P\left(fail_2\right) = B(n, a, p_D) \times B(n-a, b, q_{UD}) \times B(b, \widetilde{b}, r_{UX,UD}) \times F(n-a-\widetilde{b}, p_X^a)$$

is the probability of failure if the relationship between the rates is $R_D > R_X \geq R_U$,

$$P\left(fail_3\right) = B(n, a, p_D) \times B(a, \widetilde{a}, s_{UD}) \times F(n-a, p_U^a)$$

is the probability of failure if the relationship between the rates is $R_U \geq R_D > R_X$,

$$P\left(fail_4\right) = B(n, a, p_D) \times B(a, \widetilde{a}, s_{UD}) \times B(\check{a}, \check{a}_X, r_{DX,DU}) \times F(n-\widetilde{a}-\check{a}_X, p_U^{\widetilde{a}+\check{a}_X})$$

is the probability of failure if the relationship between the rates is $R_U > R_X \geq R_D$,

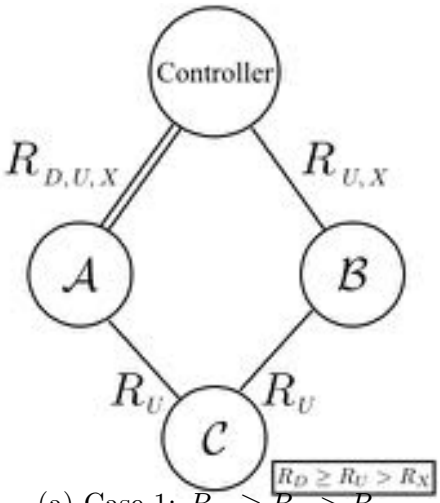
$$P\left(fail_5\right) = B(n, a, p_D) \times B(a, \widetilde{a}, s_{UD}) \times B(\widetilde{a}, \widetilde{a}_X, s_{XU}) \times (1 - P(success_5))$$

$$P(success_5) = (1 - p_X^{\widetilde{a}_X})^{\check{a}} \times \left(\sum_{k=1}^{\widetilde{a}_X} B(\widetilde{a}_X, k, p_U) \left(1 - s_{XU}^k + s_{XU}^k (1 - p_X^{\widetilde{a}_U})\right) \right)^{n-a}$$

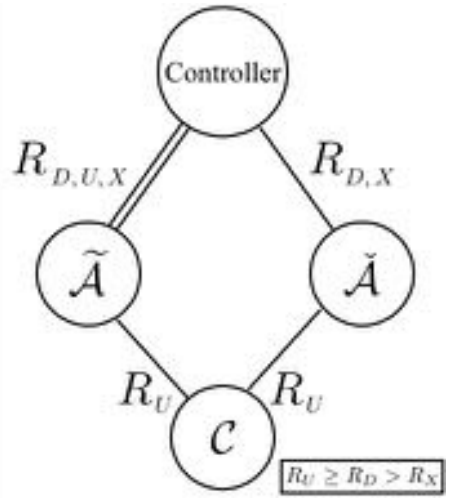
are the probabilities of failure and success if the relationship between the rates is $R_X \geq R_U > R_D$, $P\left(fail_6\right) = B(n, a, p_D) \times B(\widetilde{a}, \widetilde{a}_X, s_{XD}) \times B(n-a, b, q_{UD}) \times (1 - P(success_6))$

$$P(success_6) = (1 - p_X^a)^b \times \left(\sum_{k=1}^{\widetilde{a}_X} B(\widetilde{a}_X, k, p_U) \left(1 - s_{XU}^k + s_{XU}^k (1 - p_X^{\widetilde{a}_U})\right) \right)^{n-a-b}$$

are the probabilities of failure and success if the relationship between the rates is $R_X > R_D \geq R_U$, where:



(a) Case 1: $R_D \geq R_U > R_X$.



(b) Case 2: $R_U \geq R_D > R_X$.

Figure B.1: Different ways to succeed in XOR-CoW protocol. The links between the controller and nodes are annotated with the rates in which they are present. The links to $\mathcal{C}$ are only denoted for the rate at which the links are important.

- $q_{UD} = P(C < R_U | C < R_D) = \frac{p_U}{p_D}$
- $s_{UD} = P(C < R_U | C > R_D) = \frac{p_U - p_D}{1 - p_D}$
- $s_{XU} = P(C < R_X | C > R_U) = \frac{p_X - p_U}{1 - p_U}$
- $s_{XD} = P(C < R_X | C > R_D) = \frac{p_X - p_D}{1 - p_D}$
- $r_{UX,UD} = P(R_U < C < R_X | R_U < C < R_D) = \frac{p_X - p_U}{p_D - p_U}$
- $r_{DX,DU} = P(R_D < C < R_X | R_D < C < R_U) = \frac{p_X - p_D}{p_U - p_D}$

Proof. All potential relays get the schedules in the scheduling phase where the rate of transmission is the same as downlink rate as stated earlier in Sec. 3.2. This ensures that all potential relays (those that have the downlink information) know when to transmit. Additionally, all nodes that need help also know which packet is intended for them as their identity is built into the packet. We look at each case to understand the subtle effects that may arise.

Case 1: $R_D \geq R_U > R_X$

The rates of transmission are as described earlier and the probabilities of a link succeeding in downlink, uplink and XOR phases are given by p_D, p_U and p_X respectively. Fig. B.1a shows the exhaustive list of ways to succeed in the first case of the XOR-CoW protocol.

- A node can succeed directly to the controller in downlink – these nodes are in set $\mathcal{A}$. As the rate in downlink phase R_D is greater than the rate in uplink phase R_U, these nodes also succeed in uplink directly to the controller (so they are an overall success). In this case, $\widetilde{\mathcal{A}} = \mathcal{A}$ as all of $\mathcal{A}$ retain links in the uplink phase and they are potential relays.

- A node can gain a link to the controller at the lower uplink rate of R_U – these nodes are in set $\mathcal{B}$. They get their downlink message directly from the controller in the XOR phase as all of them retain the link to the controller in the XOR phase.

- A node can have both downlink and uplink successes during the XOR phase, if they connected to $\mathcal{A}$ in the uplink phase and as the rate R_X in the XOR phase is less than R_U, the links do not disappear.

To calculate the probability of error of the XOR-CoW protocol, we will unroll the state space and sum over all possible instantiations of the sets of interest that result in failure.

The probability of $A = a$ depends on the point to point link to the controller which has a failure probability of p_D (we use Eq. (B.3)). Thus we have $P(A = a) = B(n, a, p_D)$. The probability that a node does not gain a link to the controller in the uplink phase given it did not have a link in the downlink phase is given by $q_{UD} = P(C < R_U | C < R_D) = p_U/p_D$. Conditioned on the realization that $A = a$, the probability that $B = b$ nodes gain links to the controller is given by $P(B = b|A = a) = B(n - a, b, q_{UD})$.

Given $A = a$ and $B = b$, the probability of a node in $\mathcal{S} \setminus (\mathcal{A} \bigcup \mathcal{B})$, failing in the XOR phase is the probability that it doesn't connect to $\mathcal{A}$ in the uplink phase. The probability of a single node failing is given by p_U^a. Thus the overall probability of failure given $A = a$ and $B = b$ is $F(n - a - b, p_U^a)$. Thus we get that the probability of failure of the XOR-CoW protocol when the relationship between the rates is $R_D \geq R_U > R_X$ is given by

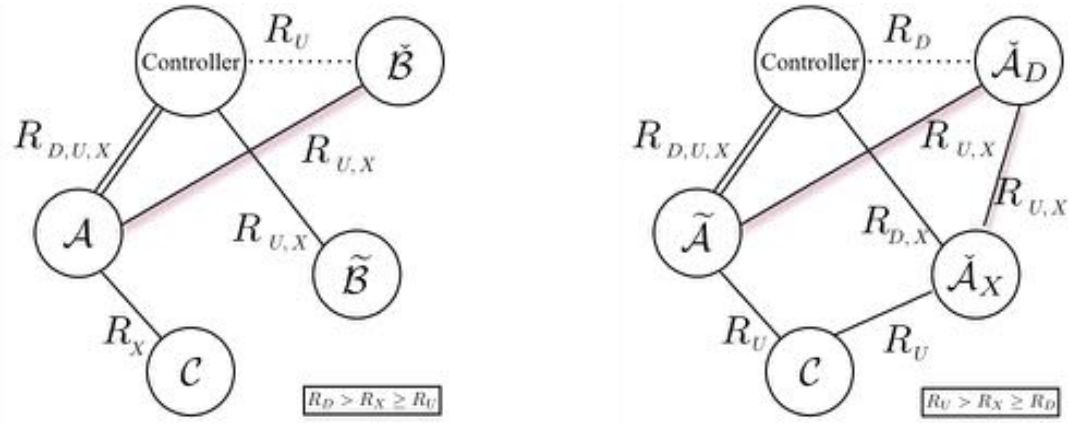
$$\sum_{a=0}^{n} \sum_{b=0}^{n-a} P\left(\text{fail}_1\right) \mathbb{1}\left(R_D \geq R_U > R_X\right)$$

where, $P\left(\text{fail}_1\right) = B(n, a, p_D) \times B(n - a, b, q_{UD}) \times F(n - a - b, p_U^a)$.

Case 2: $R_U \geq R_D > R_X$

The rates of transmission are as described earlier and the probabilities of a link succeeding in downlink, uplink and XOR phases are given by p_D, p_U and p_X respectively. Fig. B.1b shows the exhaustive list of ways to succeed in the third case of the XOR-CoW protocol.

- A node can succeed directly to the controller in downlink – these nodes are in set $\mathcal{A}$. As the rate R_D in the downlink phase is lower than the rate R_U in the uplink phase, this set is further divided into two disjoint sets $\widetilde{\mathcal{A}}$ (which retains the connection to the controller in the uplink phase) and $\check{\mathcal{A}}$ (which loses the connection to the controller in the uplink phase). The nodes in $\widetilde{\mathcal{A}}$ are the potential uplink message helpers in the XOR phase.



(a) Case 3: $R_D > R_X \geq R_U$.

(b) Case 4: $R_U > R_X \geq R_D$.

Figure B.2: Different ways to succeed in XOR-CoW protocol. The links between the controller and nodes are annotated with the rates in which they are present. The links to $\mathcal{C}$ are only denoted for the rate at which the links are important.

- The nodes in $\check{\mathcal{A}}$ succeed directly to the controller in the XOR phase as they have the downlink as well as uplink packets to XOR.
- A node can have both downlink and uplink successes during the XOR phase, if they connected to $\mathcal{A}$ in the uplink phase and as the rate in XOR phase R_X is less than R_U, the links do not disappear.

To calculate the probability of error of the XOR-CoW protocol, we will unroll the state space and sum over all possible instantiations of the sets of interest that result in failure. The probability of $A = a$ depends on the point to point link to the controller which has a failure probability of p_D (we use Eq. (B.3)). Thus we have $P(A = a) = B(n, a, p_D)$. Given $A = a$, the probability that a node in $\mathcal{A}$ loses its link to the controller in the uplink phase is given by $s_{UD} = P(C < R_U | C > R_D) = (p_U - p_D)/(1 - p_D)$. Thus we get the probability that $\widetilde{A} = \widetilde{a}$ (these do not lose the links) given $A = a$ is $B(a, \widetilde{a}, s_{UD})$. Given $A = a$ and $\widetilde{A} = \widetilde{a}$, the probability of a node in $\mathcal{S} \setminus \mathcal{A}$, failing in the XOR phase is the probability that it doesn't connect to $\mathcal{A}$ in the uplink phase. The probability of a single node failing is given by p_U^a. Thus, the overall probability of failure given $A = a$ and $\widetilde{A} = \widetilde{a}$ is $F(n - a, p_U^a)$. Thus, we get that the probability of failure of the XOR-CoW protocol when the relationship between the rates is $R_U \geq R_D > R_X$ is given by

$$\sum_{a=0}^{n} \sum_{\widetilde{a}=0}^{a} P\left(\text{fail}_3\right) \mathbb{1}\left(R_U \geq R_D > R_X\right)$$

where, $P\left(\text{fail}_3\right) = B(n, a, p_D) \times B(a, \widetilde{a}, s_{UD}) \times F(n - a, p_U^a)$.

Case 3: $R_D > R_X \geq R_U$
The rates of transmission are as described earlier and the probabilities of a link succeeding in downlink, uplink and XOR phases are given by p_D, p_U and p_X respectively. Fig. B.2a shows the exhaustive list of ways to succeed in the second case of the XOR-CoW protocol.

- A node can succeed directly to the controller in downlink – these nodes are in set $\mathcal{A}$. As the rate in downlink phase R_D is greater than the rate in uplink phase R_U, these nodes also succeed in uplink directly to the controller (they are an overall success). In this case $\widetilde{\mathcal{A}} = \mathcal{A}$ as all nodes in $\mathcal{A}$ retain links in uplink phase. All of these will be potential relays.

- A node can gain a link to the controller at the lower uplink rate of R_U – these nodes are in the set $\mathcal{B}$. Some of these nodes lose the link during the XOR phase as (since $R_X \geq R_U$). The nodes that retain the links constitute the set $\widetilde{\mathcal{B}}$ and the ones which lose the link constitute the set $\check{\mathcal{B}}$. The set $\widetilde{\mathcal{B}}$ get their downlink message directly from the controller in the XOR phase but the set $\check{\mathcal{B}}$ doesn't. They need to connect to at least one node in $\mathcal{A}$ in the uplink as well as XOR phase to successfully receive their downlink message.

- A node can have both downlink and uplink successes during the XOR phase, if they connected to $\mathcal{A}$ in the uplink phase as well as in the XOR phase (similar to $\check{\mathcal{B}}$).

To calculate the probability of error of the XOR-CoW protocol, we will unroll the state space and sum over all possible instantiations of the sets of interest that result in failure. The probability of $A = a$ depends on the point to point link to the controller which has a failure probability of p_D (we use Eq. (B.3)). Thus we have $P(A = a) = B(n, a, p_D)$. The probability that a node does not gain a link to the controller in the uplink phase given it did not have a link in the downlink phase is given by $q_{UD} = P(C < R_U | C < R_D) = p_U/p_D$. Conditioned on the realization that $A = a$, the probability that $B = b$ nodes gain link to the controller is given by $P(B = b|A = a) = B(n - a, b, q_{UD})$. Given $B = b$, the probability that a node in $\mathcal{B}$, loses the connection to the controller in the XOR phase is given by $r_{UX,UD} = p(R_U < C < R_X | R_C < C < R_D) = (p_X - p_U)/(p_D - p_U)$. Thus the probability that $\widetilde{B} = \widetilde{b}$ given $B = b$ is given by $B(b, \widetilde{b}, r_{UX,UD})$. Given $A = a$, $B = b$ and $\widetilde{B} = \widetilde{b}$ the probability of a node in $\mathcal{S} \setminus \left(\mathcal{A} \bigcup \widetilde{\mathcal{B}}\right)$, failing in the XOR phase is the probability that it doesn't connect to $\mathcal{A}$ in the uplink and XOR phases. The probability of a single node failing is given by p_X^a. Thus, the overall probability of failure given $A = a$, $B = b$ and $\widetilde{B} = \widetilde{b}$ is $F(n - a - \widetilde{b}, p_X^a)$. Thus, we get that the probability of failure of the XOR-CoW protocol when the relationship between the rates is $R_D > R_X \geq R_U$ is given by

$$\sum_{a=0}^{n} \sum_{b=0}^{n-a} \sum_{\widetilde{b}=0}^{b} P\left(\text{fail}_2\right) \mathbb{1}\left(R_D > R_X \geq R_U\right)$$

where, $P\left(\text{fail}_2\right) = B(n, a, p_D) \times B(n - a, b, q_{UD}) \times B(b, \widetilde{b}, r_{UX,UD}) \times F(n - a - \widetilde{b}, p_X^a)$.

Case 4: $R_U > R_X \geq R_D$
The rates of transmission are as described earlier and the probabilities of a link succeeding in downlink, uplink and XOR phases are given by p_D, p_U and p_X respectively. Fig. B.2b shows the exhaustive list of ways to succeed in the fourth case of the XOR-CoW protocol.

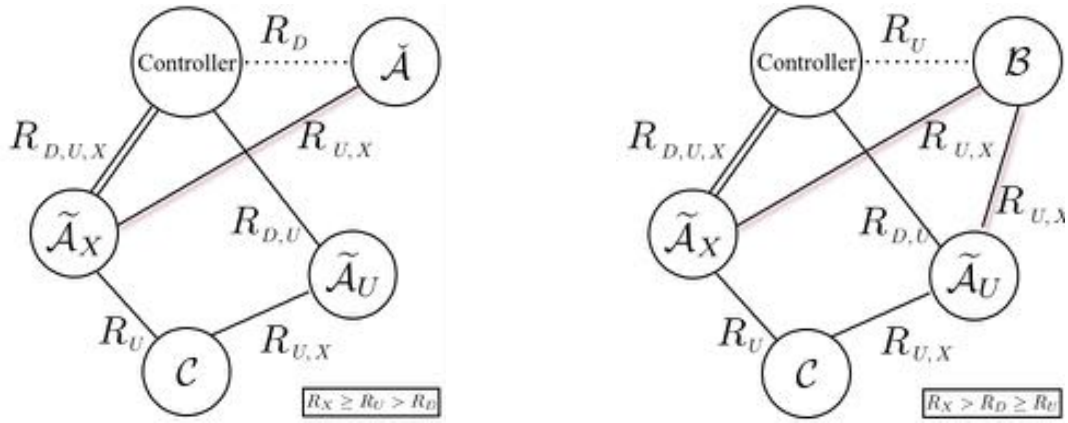
- A node can succeed directly to the controller in downlink – these nodes are in set $\mathcal{A}$. As the rate in downlink phase R_D is lower than the rate in uplink phase R_U, this set is further divided into two disjoint sets $\widetilde{\mathcal{A}}$ (which retains the connection to the controller in the uplink phase) and $\check{\mathcal{A}}$ (which loses the connection to the controller in the uplink phase).
- The nodes in $\check{\mathcal{A}}$ are further divided to $\check{\mathcal{A}}_X$ (those that regain the link to the controller in the XOR phase) and $\check{\mathcal{A}}_D$ (those that do not regain the link to the controller). The nodes in $\check{\mathcal{A}}_X$ successfully transmit their own uplink message to the controller in the XOR phase as they have the downlink messages to XOR and the link to transmit.
- The nodes in $\check{\mathcal{A}}_D$ succeed only by connecting to $\widetilde{\mathcal{A}} \bigcup \check{\mathcal{A}}$ in the uplink phase (the link back to them will automatically exist in the XOR phase since $R_X < R_U$).
- Any other node can have both downlink and uplink successes during the XOR phase, if they connected to $\widetilde{\mathcal{A}} \bigcup \check{\mathcal{A}}_X$ in the uplink phase and as the rate in XOR phase R_X is less than R_U, the links do not disappear.

To calculate the probability of error of the XOR-CoW protocol, we will unroll the state space and sum over all possible instantiations of the sets of interest that result in failure. The probability of $A = a$ depends on the point to point link to the controller which has a failure probability of p_D (we use Eq. (B.3)). Thus we have $P(A = a) = B(n, a, p_D)$. Given $A = a$, the probability that a node in $\mathcal{A}$ loses link to the controller in the uplink phase is given by $s_{UD} = P(C < R_U | C > R_D) = (p_U - p_D)/(1 - p_D)$. Thus we get the probability that $\widetilde{A} = \widetilde{a}$ (these do not lose the links) given $A = a$ is $B(a, \widetilde{a}, s_{UD})$. Given $A = a$ and $\widetilde{A} = \widetilde{a}$, the probability of a node in $\check{\mathcal{A}}$ gaining a link to the controller in the XOR phase is given by $1 - P(R_D < C < R_X | R_D < C < R_U) = 1 - r_{DX,DU}$. Thus, we get that $\check{A}_X = \check{a}_X$ nodes gain links to the controller in the XOR phase with probability $B(\check{a}, \check{a}_X, r_{DX,DU})$.

Given $A = a$, $\widetilde{A} = \widetilde{a}$ and $\widetilde{A}_X = \widetilde{a}_X$, the probability of a node in $\mathcal{S} \setminus \left(\widetilde{\mathcal{A}} \bigcup \check{\mathcal{A}}_X\right)$ failing in the XOR phase is the probability that it doesn't connect to $\widetilde{\mathcal{A}} \bigcup \check{\mathcal{A}}_X$ in the uplink phase. The probability of a single node failing is given by $p_U^{\widetilde{a}+\check{a}_X}$. Thus the overall probability of failure given $A = a$ and $\widetilde{A} = \widetilde{a}$ is $F(n - \widetilde{a} - \check{a}_X, p_U^{\widetilde{a}+\check{a}_X})$. Thus we get that the probability of failure of the XOR-CoW protocol when the relationship between the rates is $R_U > R_X \geq R_D$ is given by

$$\sum_{a=0}^{n} \sum_{\widetilde{a}=0}^{a} \sum_{\check{a}_X=0}^{a-\widetilde{a}} P\left(\text{fail}_4\right) \mathbb{1}\left(R_U > R_X \geq R_D\right)$$

where, $P\left(\text{fail}_4\right) = B(n, a, p_D) \times B(a, \widetilde{a}, s_{UD}) \times B(\check{a}, \check{a}_X, r_{DX,DU}) \times F(n - \widetilde{a} - \check{a}_X, p_U^{\widetilde{a}+\check{a}_X})$.



(a) Case 5: $R_X \geq R_U > R_D$.

(b) Case 6: $R_X > R_D \geq R_U$.

Figure B.3: Different ways to succeed in XOR-CoW protocol. The links between the controller and nodes are annotated with the rates in which they are present. The links to $\mathcal{C}$ are only denoted for the rate at which the links are important.

Case 5: $R_X \geq R_U > R_D$

The rates of transmission are as described earlier and the probabilities of a link succeeding in downlink, uplink and XOR phases are given by p_D, p_U and p_X respectively. Fig. B.3a shows the exhaustive list of ways to succeed in the fifth case of the XOR-CoW protocol.

- A node can succeed directly to the controller in downlink – these nodes are in set $\mathcal{A}$. As the rate in downlink phase R_D is lower than the rate in uplink phase R_U, this set is further divided into two disjoint sets $\widetilde{\mathcal{A}}$ (which retains the connection to the controller in the uplink phase) and $\check{\mathcal{A}}$ (which loses the connection to the controller in the uplink phase).
- The nodes in $\widetilde{\mathcal{A}}$ are further divided to $\widetilde{\mathcal{A}}_X$ (those that retain the link to the controller in the XOR phase – thus can act as uplink message relays) and $\widetilde{\mathcal{A}}_U$ (those that lose the link to the controller). The set $\widetilde{\mathcal{A}}_U$ can still act a relays for downlink messages.
- The nodes in $\check{\mathcal{A}}$ succeed only if they connect to $\widetilde{\mathcal{A}}_X$ in the uplink phase.
- The nodes in $\mathcal{S} \setminus \mathcal{A}$ succeed only in the following way: they must connect to $\widetilde{\mathcal{A}}_X$ in the uplink phase (to convey their uplink message). They can receive their downlink message either by connecting to $\widetilde{\mathcal{A}}_X$ in the XOR phase (this is not guaranteed as the rate in the XOR phase is higher) or by connecting to $\widetilde{\mathcal{A}}_U$ in the uplink and XOR phase.

To calculate the probability of error of the XOR-CoW protocol, we will unroll the state space and sum over all possible instantiations of the sets of interest that result in failure. The probability of $A = a$ depends on the point to point link to the controller which has a

failure probability of p_D (we use Eq. (B.3)). Thus we have $P(A = a) = B(n, a, p_D)$. Given $A = a$, the probability that a node in $\mathcal{A}$ loses link to the controller in the uplink phase is given by $s_{UD} = P(C < R_U | C > R_D) = (p_U - p_D)/(1 - p_D)$. Thus we get the probability that $\widetilde{A} = \widetilde{a}$ (these do not lose the links) given $A = a$ is $B(a, \widetilde{a}, s_{UD})$. Given $A = a$ and $\widetilde{A} = \widetilde{a}$, the probability that a node in $\widetilde{\mathcal{A}}$ loses link to the controller in the XOR phase is given by $s_{XU} = P(C < R_X | C > R_U) = (p_X - p_U)/(1 - p_U)$. Thus, the probability that $\widetilde{\mathcal{A}}_X = \widetilde{\mathcal{A}}_X$ is given by $B(\widetilde{a}, \widetilde{a}_X, s_{XU})$.

The probability that nodes in $\check{\mathcal{A}}$ succeed is the probability that they connect to $\widetilde{\mathcal{A}}_X$ in the uplink phase which is given by $1 - p_U^{\widetilde{a}_X}$. Thus the probability that all nodes in $\check{\mathcal{A}}$ succeed is $(1 - p_U^{\widetilde{a}_X})^{\check{a}}$. For the rest of the nodes, let us calculate the probability of success. To succeed, a node *must* connect to $\widetilde{\mathcal{A}}_X$ in the uplink phase. Let us consider that the node is connected to k nodes in $\widetilde{\mathcal{A}}_X$. The probability of this event is $B(\widetilde{a}_x, k, p_U)$. This is essential for uplink success. Downlink can succeed either by connecting to one of these k nodes in $\widetilde{\mathcal{A}}_X$ in the XOR phase or by having a connection to $\widetilde{\mathcal{A}}_U$ in the uplink as well as XOR phases. Thus we have the probability of downlink success is $\left(\left(1 - s_{XU}^k\right) + s_{XU}^k(1 - p_X^{\widetilde{a} - \widetilde{a}_X})\right)$. Combining the uplink and downlink success we get that a node in $\mathcal{S} \setminus \mathcal{A}$ succeeds with a probability $B(\widetilde{a}_x, k, p_U) \times \left(\left(1 - s_{XU}^k\right) + s_{XU}^k(1 - p_X^{\widetilde{a} - \widetilde{a}_X})\right)$. Thus, probability of success in Case 5 is given by

$$P(\text{success}_5) = (1 - p_U^{\widetilde{a}_X})^{\check{a}} \times \left(\sum_{k=1}^{\widetilde{a}_X} B(\widetilde{a}_X, k, p_U) \left(\left(1 - s_{XU}^k\right) + s_{XU}^k(1 - p_X^{\widetilde{a} - \widetilde{a}_X})\right) \right)^{n-a}. \quad \text{(B.4)}$$

Thus we get that the probability of failure of the XOR-CoW protocol when the relationship between the rates is $R_X \geq R_U > R_D$ is given by

$$\sum_{a=0}^{n} \sum_{\widetilde{a}=0}^{a} \sum_{\widetilde{a}_X=0}^{\widetilde{a}} P\left(\text{fail}_5\right) \mathbb{1}\left(R_X \geq R_U > R_D\right)$$

where, $P\left(\text{fail}_5\right) = B(n, a, p_D) \times B(a, \widetilde{a}, s_{UD}) \times B(\widetilde{a}, \widetilde{a}_X, s_{XU}) \times (1 - P(\text{success}_5))$.

Case 6: $R_X > R_D \geq R_U$

The rates of transmission are as described earlier and the probabilities of a link succeeding in downlink, uplink and XOR phases are given by p_D, p_U and p_X respectively. Fig. B.3b shows the exhaustive list of ways to succeed in the second case of the XOR-CoW protocol.

- A node can succeed directly to the controller in downlink – these nodes are in set $\mathcal{A}$. All the nodes in set $\mathcal{A}$ succeed in uplink as the rate R_U is less than R_D. Thus, $\mathcal{A} = \widetilde{\mathcal{A}}$.
- A node can gain a link to the controller at the lower uplink rate of R_U – these nodes are in set $\mathcal{B}$. Note that these succeeded only at R_U and not at R_D and hence these nodes cannot help to get to the controller in the higher XOR phase rate of R_X.

- The nodes in $\widetilde{\mathcal{A}}$ are further divided to $\widetilde{\mathcal{A}}_X$ (those that retain the link to the controller in the XOR phase) and $\widetilde{\mathcal{A}}_U$ (those that lose the link to the controller in the XOR phase). Only $\widetilde{\mathcal{A}}_X$ can effectively relay the uplink messages of the nodes in need.
- The nodes in $\mathcal{S} \setminus \mathcal{A}$ succeed only in the following way: they must connect to $\widetilde{\mathcal{A}}_X$ in the uplink phase (to convey their uplink message). They can receive their downlink message either by connecting to $\widetilde{\mathcal{A}}_X$ in the XOR phase as well (this is not guaranteed as the rate in the XOR phase is higher) or by connecting to $\widetilde{\mathcal{A}}_U$ in the uplink as well as XOR phase.

To calculate the probability of error of the XOR-CoW protocol, we will unroll the state space and sum over all possible instantiations of the sets of interest that result in failure. The probability of $A = a$ depends on the point to point link to the controller which has a failure probability of p_D (we use Eq. (B.3)). Thus we have $P(A = a) = B(n, a, p_D)$. The probability that a node does not gain a link to the controller in the uplink phase given it did not have a link in the downlink phase is given by $q_{UD} = P(C < R_U | C < R_D) = p_U/p_D$. Conditioned on the realization that $A = a$, the probability that $B = b$ nodes gain link to the controller is given by $P(B = b|A = a) = B(n - a, b, q_{UD})$. The probability that nodes in $\mathcal{B}$ succeed is the probability that they connect to $\widetilde{\mathcal{A}}_X$ in the uplink phase which is given by $1 - p_U^{\widetilde{a}_X}$. Thus the probability that all nodes in $\mathcal{A}$ succeed is $(1 - p_U^{\widetilde{a}_X})^b$. Given $A = a$, $\widetilde{A} = \widetilde{a}$ and $B = b$, the probability that a node in $\widetilde{\mathcal{A}}$ loses its link to the controller in the XOR phase is given by $s_{XD} = P(C < R_X | C > R_D) = (p_X - p_D)/(1 - p_D)$. Thus, the probability that $\widetilde{A}_X = \widetilde{a}_X$ is given by $B(\widetilde{a}, \widetilde{a}_X, s_{XD})$.

For the rest of the nodes, let us calculate the probability of success. In order to succeed, a node *must* connect to $\widetilde{\mathcal{A}}_X$ in the uplink phase. Let us consider that the node is connected to k nodes in $\widetilde{\mathcal{A}}_X$. The probability of this event is $B(\widetilde{a}_x, k, p_U)$. This is essential for uplink success. Downlink can succeed either by connecting to one of these k nodes in $\widetilde{\mathcal{A}}_X$ in the XOR phase or by having a connection to $\widetilde{\mathcal{A}}_D$ in the XOR phase. Thus we have the probability of downlink success is $\left(\left(1 - s_{XU}^k\right) + s_{XU}^k(1 - p_X^{\widetilde{a}-\widetilde{a}_X})\right)$. Combining the uplink and downlink success we get that a node in $\mathcal{S} \setminus \mathcal{A}$ succeeds with a probability $B(\widetilde{a}_x, k, p_U) \times \left(\left(1 - s_{XU}^k\right) + s_{XU}^k(1 - p_X^{\widetilde{a}-\widetilde{a}_X})\right)$. Thus, probability of success in case 6 is given by

$$P(\text{success}_6) = (1 - p_X^a)^b \times \left(\sum_{k=1}^{\widetilde{a}_X} B(\widetilde{a}_X, k, p_U) \left(\left(1 - s_{XU}^k\right) + s_{XU}^k(1 - p_X^{\widetilde{a}-\widetilde{a}_X})\right)\right)^{n-a-b}. \quad \text{(B.5)}$$

Thus we get that the probability of failure of the XOR-CoW protocol when the relationship between the rates is $R_X > R_D \geq R_U$ is given by

$$\sum_{a=0}^{n} \sum_{\widetilde{a}_X=0}^{a} \sum_{b=0}^{n-a} P\left(\text{fail}_6\right) \mathbb{1}\left(R_X > R_D \geq R_U\right)$$

where, $P\left(\text{fail}_6\right) = B(n, a, p_D) \times B(\widetilde{a}, \widetilde{a}_X, s_{XD}) \times B(n-a, b, q_{UD}) \times (1 - P(\text{success}_6))$.

□

Appendix C

Detailed analysis of the effects of uncertainties

The total time available is T. In the case of 2-hop Occupy CoW protocol where there are four phases, equal time is given to each phase ($T/4$) and each phase allocates equal time for each node ($T/4n$). In the case of XOR-CoW protocol where there are three phases, equal time is given to each phase ($T/3$) and each phase allocates equal time for each node ($T/3n$).

All the connections are wireless and each node can speak to any other node if the capacity of the link is sufficient to sustain the rate. Let the rate at which all the nodes are transmitting be R and the operating power be denoted by SNR. If the links are Rayleigh faded we have,

$$P(C < R) = p = 1 - \exp(-\frac{2^R - 1}{SNR}) \tag{C.1}$$

which is the probability that a link does not exist at the given rate and SNR.

In section C.1 we look at the combination of unmodeled errors when considering the nominal model. In section C.2 we analyze the performance of both protocols when the channel fade realizations change during the cycle. In section C.3 we look at the effect of capping the number of transmitters that can simultaneously transmit.

C.1 Combined effect of unmodeled errors

In this section, we calculate the probability of a failed cycle considering all the events that can lead to a failure. The main events that lead to failure are a) deep fade causing links to be bad modeled by p_w, b) uncertainty in fading model modeled by p_{off}, c) global per-slot badness such as interference that does not cumulate with the number of transmitters p_g, and d) per-transmitter induced badness due to say mis-synchronized packets p_c which cumulates with the number of transmitters. The unmodeled error events are listed in Table. C.1.

The network comprises of a single controller and a set $\mathcal{S}$ of n client nodes. The controller has m bits of information for each node and vice-versa. The total time available is T which

Parameter	Range	Unmodeled events captured
p_{off}	$0 - 0.1$	Imperfections in channel fade modeling. Spatial-correlation-based degradation can also be captured through this term.
p_c	$0 - 10^{-2}$	Errors due to clock mis-synchronizations or channels changing rapidly during a single packet. These errors compound with increasing number of simultaneous transmitters.
p_g	$0 - 10^{-2}$	Global errors that are due to burst-interference like events. These errors do not compound with increasing number of simultaneous transmitters. They fundamentally exist at the receivers.

Table C.1: Uncertainties captured and the parameters associated with them. Except for p_{off}, all of these are essentially independent from time-slot to time-slot if we assume that the communication scheme interleaves repetitions. Some of these might implicitly depend on the length of the time-slots (growing with time-slot length), but this dependence is suppressed here.

will be evenly divided among initial downlink and uplink phases and relaying downlink and uplink phases. Let the number of retransmissions in the initial phase and relaying phase be k_1 and k_2 respectively. Therefore the rate of transmission in initial downlink and uplink phases is $R_{w_1} = \frac{m \cdot n \cdot k_1}{T/4}$ and the rate of transmission in the relaying downlink and uplink phases is $R_{w_2} = \frac{m \cdot n \cdot k_2}{T/4}$. Let the perfect Rayleigh fading model based link failure probability for the initial and relaying phases be given by p_{w_1} and p_{w_2} respectively. They depend on the operating SNR as $p_{w_i} = 1 - \exp\left(-\frac{2^R_{w_i} - 1}{SNR}\right)$ for $i = 1, 2$.

Let the set of nodes that have a good link to the controller in the initial phase be $\mathcal{A}$. The random variable associated with this set be A and a particular instance of it be a. A node can succeed if it either is in $\mathcal{A}$ or if it connects successfully to a node in $\mathcal{A}$ and no fade-unrelated error event occurred (in both cases). The probability that set $\mathcal{A}$ has a nodes is

$$P(A = a) = \binom{n}{a}(1 - p_1)^a (p_1)^{(n-a)}(1 - p_c^{k_1})^a (1 - p_g^{k_1}) \tag{C.2}$$

where $p_1 = p_{w_1} + p_{off}$. The probability that a node in $\mathcal{S} \backslash \mathcal{A}$ succeeds can be calculated as follows. Let us say there is a set of potential nodes $\mathcal{R}$ out of the nodes in $\mathcal{A}$ that actually have a good link to our node of interest. The probability that $R = r$ given $A = a$ is given by

$$P(R = r | A = a) = \binom{a}{r}(1 - p_{w_2} - p_{off})^r (p_{w_2} + p_{off})^{(a-r)}.$$

We assume that each of the r nodes know that they are good relays and the $a - r$ nodes know that they are bad relays. They can acquire that knowledge by looking at the channel energy of the links measured during the initial phases. These r relays simultaneously transmit k_2 times. Then the probability of successfully receiving the message during the relaying phase

is

$$P(\text{relaying success}|A=a,R=r) = 1-(1-(1-p_c)^r(1-p_g))^{k_2}. \tag{C.3}$$

This gives us that the overall probability of success of a node in $\mathcal{S}\backslash\mathcal{A}$ is given by

$$P(\text{relaying success}|A=a) = \sum_{r=1}^{a} P(R=r|A=a)P(\text{relaying success}|A=a,R=r) \tag{C.4}$$

Therefore the overall probability of failure of a cycle is

$$P(fail) = p_g^{k_1} + (p_{w_1}+p_{off})^n + \sum_{a=1}^{n-1}(1-P(\text{relaying success}|A=a)^{n-a})P(A=a). \tag{C.5}$$

C.2 Effect of non-quasi-static channels

Channels can change anytime during a cycle in this case and then they remain constant for rest of the cycle time. The worst case modeling of channel changing is when they change at the boundary between phases. The worst case modeling of channels changing during a particular phase is when channel fades were drawn independent of their previous fades at the beginning of that particular phase. To this effect, we will assume that channels change in the boundary between two phases to account for the worst case scenario. We model the change as a new independent draw of fades which is not correlated with the previous realizations. We now consider the effect of channel change on Occupy CoW and XOR-CoW.

C.2.1 Occupy CoW

As there are four phases in 2-hop Occupy CoW, the channel realizations can change at 4 instances (with equal probability): between uplink phase 2 and downlink phase 1 (new cycle realization), between downlink phase 1 and uplink phase 1, between uplink phase 1 and downlink phase 2 and between downlink phase 2 and uplink phase 2. We consider each of them in turn and finally model the worst case probability of failure as the maximum over all 4 possible probabilities of cycle failure.

Case 1: Channels change between uplink phase 2 and downlink phase 1

In this case, the channels change when a new cycle is about the begin. The probability of failure in this case is

$$P(\text{fail, case 1}) = \sum_{a=0}^{n-1}\left(\binom{n}{a}(1-p_w)^a p_w^{n-a}\right)\left(1-(1-p_w^a)^{n-a}\right) \tag{C.6}$$

Case 2: Channels change between downlink phase 1 and uplink phase 1
In this case, the channels change after the cycle has begun and after this change, the channel realizations remain constant. Let the nodes which succeeded in downlink phase 1 be the set $\mathcal{A}$. When the channels change before uplink phase 1, some of the nodes in the set $\mathcal{A}$ lose their connection to the controller (set $\check{\mathcal{A}}$) and some of them retain it (set $\widehat{\mathcal{A}}$). Some nodes in the set $\mathcal{S}\backslash\mathcal{A}$ gain link to the controller (set $\widehat{\mathcal{B}}$). The ways to succeed are listed below:

- In downlink phase 1, nodes in set $\mathcal{A}$ succeed in getting downlink information. Let the set be divided into two disjoint sets: $\mathcal{A} = \widehat{\mathcal{A}} \bigcup \check{\mathcal{A}}$.
- In uplink phase 1, nodes in set $\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}$ succeed in conveying their uplink information to the controller. So far only the nodes in set $\widehat{\mathcal{A}}$ have succeeded in both uplink and downlink.
- In downlink phase 2, the nodes in set $\widehat{\mathcal{B}}$ succeed directly through the controller and all other nodes in $\mathcal{S}\backslash\{\mathcal{A} \bigcup \widehat{\mathcal{B}}\}$ can succeed if they connect to set $\mathcal{A}$.
- In uplink phase 2, the nodes in set $\check{\mathcal{A}}$ can succeed if they connected to $\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}$ in uplink phase 1. And the nodes in $\mathcal{S}\backslash\{\mathcal{A} \bigcup \widehat{\mathcal{B}}\}$ succeed if they connected to the set $\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}$ in uplink phase 1.

Thus, the probability that $A = a$ is given by

$$P(A = a) = \binom{n}{a} (1 - p_w)^a p_w^{n-a} \tag{C.7}$$

Conditioned on $A = a$, the probability of $\widehat{A} = \widehat{a}$ is a binomial distribution with a number of experiments, $\widehat{a}$ number of heads and $(1 - p_w)$ being the probability of heads. Thus the probability is given by

$$P(\widehat{A} = \widehat{a} | A = a) = \binom{a}{\widehat{a}} (1 - p_w)^{\widehat{a}} p_w^{a-\widehat{a}} \tag{C.8}$$

Conditioned on $A = a$, the probability of $\widehat{B} = \widehat{b}$ is derived similar to Eq. (C.8) where the number of experiments is $n - a$ and the probability is given by

$$P(\widehat{B} = \widehat{b} | A = a) = \binom{n-a}{\widehat{b}} (1 - p_w)^{\widehat{b}} p_w^{n-a-\widehat{b}} \tag{C.9}$$

The probability of success of the nodes in $\check{\mathcal{A}}$ conditioned on the states is the probability that they connect to $\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}$ in uplink phase 1. The probability that a node is connected to $\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}$ is the probability that it is connected to at least one of the nodes in $\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}$ which is $\left(1 - p_w^{\widehat{a}+\widehat{b}}\right)$ (it is one minus the probability that the node is disconnected from every node in $\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}$). Thus we have,

$$P(\text{success of nodes in } \check{\mathcal{A}} | \widehat{B} = \widehat{b}, \widehat{A} = \widehat{a}) = \left(1 - p_w^{\widehat{a}+\widehat{b}}\right)^{\check{a}} \tag{C.10}$$

The probability of success of the nodes in $\mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}$ conditioned on the states is the probability that they connect to $\widehat{\mathcal{A}}\bigcup\widehat{\mathcal{B}}$ in uplink phase 1 and $\mathcal{A}$ in downlink phase 2 (they can either connect to the intersection $\widehat{A}$ or to both $\check{A}$ and $\widehat{B}$). Thus we have,

$$P(\text{success of nodes in } \mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a}) = \left(\left(1-p_w^{\widehat{a}}\right)+p_w^{\widehat{a}}\left(1-p_w^{\check{a}}\right)\left(1-p_w^{\widehat{b}}\right)\right)^{n-a-\widehat{b}} \tag{C.11}$$

Combining the equations (C.10) and (C.11) we have the probability of cycle failure conditioned on the states,

$$P(\text{fail}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a},A=a) = 1-P(\text{success}|\text{states}) \tag{C.12}$$

where,

$$\begin{aligned}&P(\text{success}|\text{states})\\ &= P(\text{success of nodes in } \check{\mathcal{A}}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a})\cdot P(\text{success of nodes in } \mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a})\end{aligned} \tag{C.13}$$

Thus from equations (C.7), (C.8), (C.9) and (C.12) we have the probability of cycle failure if the channel changes between downlink phase 1 and uplink phase 1 to be:

$$\begin{aligned}&P(\text{fail, case 2})\\ &= \sum_{a=0}^{n-1}\sum_{\widehat{a}=0}^{a}\sum_{\widehat{b}=0}^{n-a-1} P(A=a)\cdot P(\widehat{A}=\widehat{a}|A=a)\cdot P(\widehat{B}=\widehat{b}|A=a)\cdot P(\text{fail}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a},A=a)\end{aligned} \tag{C.14}$$

Case 3: Channels change between uplink phase 1 and downlink phase 2
In this case, the channels change after the cycle has begun and after this change, the channel realizations remain constant. Let the nodes which succeeded in downlink phase 1 be the set $\mathcal{A}$. The same set of nodes succeed in uplink phase 1. When the channels change before downlink phase 1, some of the nodes in the set $\mathcal{A}$ lose their connection to the controller (set $\check{\mathcal{A}}$) and some of them retain it (set $\widehat{\mathcal{A}}$). Some nodes in the set $\mathcal{S}\backslash\mathcal{A}$ gain link to the controller (set $\widehat{\mathcal{B}}$). The ways to succeed are listed below:

- In downlink phase 1, nodes in set $\mathcal{A}$ succeed in getting downlink information.
- In uplink phase 1, nodes in set $\mathcal{A}$ succeed in conveying their uplink information to the controller. $\mathcal{A}=\widehat{\mathcal{A}}\bigcup\check{\mathcal{A}}$.
- In downlink phase 2, the nodes in set $\widehat{\mathcal{B}}$ succeed directly through the controller and all other nodes in $\mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}$ can succeed if they connect to set $\mathcal{A}$ in downlink phase 2 (after channel change).

- In uplink phase 2, the nodes in set $\widehat{\mathcal{B}}$ succeed directly through the controller. And the nodes in $\mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}$ succeed if they connected to the set $\widehat{\mathcal{A}}\bigcup\widehat{\mathcal{B}}$ in uplink phase 1 (before channel change). Thus, combining downlink phase 2 and uplink phase 2 success paths for the nodes in $\mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}$ we get that, they can succeed only if they have a link to the set $\widehat{\mathcal{A}}\bigcup\widehat{\mathcal{B}}$ in uplink phase 1 (before channel change) and to the set $\mathcal{A}$ in downlink phase 2 (after channel change).

The probabilities $P(A=a)$, $P(\widehat{A}=\widehat{a}|A=a)$, $P(\widehat{B}=\widehat{b}|A=a)$ are the same as Eq. (C.7), (C.8), (C.9).

The probability of failure of the nodes in $\mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}$ conditioned on the states is the failure to connect to the set $\widehat{\mathcal{A}}\bigcup\widehat{\mathcal{B}}$ in uplink phase 1. Thus the probability of failure conditioned on the states is given by

$$\begin{aligned}&P(\text{fail}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a})\\&=(1-p_w^a)^{n-a-\widehat{b}}\left(1-p_w^{\widehat{a}+\widehat{b}}\right)^{n-a-\widehat{b}}\end{aligned}\tag{C.15}$$

Thus from equations (C.7), (C.8), (C.9), (C.15) we have the probability of cycle failure if the channels changes between uplink phase 1 and downlink phase 2 to be:

$$\begin{aligned}&P(\text{fail, case 3})\\&=\sum_{a=0}^{n-1}\sum_{\widehat{a}=0}^{a}\sum_{\widehat{b}=0}^{n-a-1}P(A=a)\cdot P(\widehat{A}=\widehat{a}|A=a)\cdot P(\widehat{B}=\widehat{b}|A=a)\cdot P(\text{fail}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a})\end{aligned}\tag{C.16}$$

Case 4: Channels change between downlink phase 2 and uplink phase 2
In this case, the channels change after the cycle has begun and after this change, the channel realizations remain constant. Let the nodes which succeeded in downlink phase 1 be the set $\mathcal{A}$. The same set of nodes succeed in uplink phase 1. When the channels change before uplink phase 2, some of the nodes in the set $\mathcal{A}$ lose their connection to the controller (set $\check{\mathcal{A}}$) and some of them retain it (set $\widehat{\mathcal{A}}$). Some nodes in the set $\mathcal{S}\backslash\mathcal{A}$ gain link to the controller (set $\widehat{\mathcal{B}}$). The ways to succeed are listed below:

- In downlink phase 1, nodes in set $\mathcal{A}$ succeed in getting downlink information.
- In uplink phase 1, nodes in set $\mathcal{A}$ succeed in conveying their uplink information to the controller. $\mathcal{A}=\widehat{\mathcal{A}}\bigcup\check{\mathcal{A}}$.
- In downlink phase 2, the nodes in $\mathcal{S}\backslash\mathcal{A}$ can succeed if they connect to set $\mathcal{A}$.
- In uplink phase 2, the nodes in $\widehat{\mathcal{B}}$ succeed directly to the controller. The other nodes in $\mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}$ succeed if they succeeded to the set $\widehat{\mathcal{A}}$ in uplink phase 1.

The probabilities $P(A=a)$, $P(\widehat{A}=\widehat{a}|A=a)$, $P(\widehat{B}=\widehat{b}|A=a)$ are the same as Eq. (C.7), (C.8), (C.9).

The probability of success of the nodes in $\widehat{\mathcal{B}}$ conditioned on the states is the probability that they connect to $\mathcal{A}$ in downlink phase 2. Thus we have,

$$P(\text{success of nodes in } \widehat{\mathcal{B}}|\widehat{B}=\widehat{b}, A=a) = (1-p_w^a)^{\widehat{b}} \tag{C.17}$$

The probability of success of the nodes in $\mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}$ conditioned on the states is the probability that they connect to $\widehat{\mathcal{A}}$ in uplink phase 1 (more discussion on this is in the discussions of the previous cases). Thus we have,

$$P(\text{success of nodes in } \mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}|\widehat{B}=\widehat{b}, \widehat{A}=\widehat{a}) = \left(\left(1-p_w^{\widehat{a}}\right)+p_w^{\widehat{a}}\left(1-p_w^{\check{a}}\right)\left(1-p_w^{\widehat{b}}\right)\right)^{n-a-\widehat{b}} \tag{C.18}$$

Thus from equations (C.17) and (C.18) we have the probability of cycle failure conditioned on the states,

$$P(\text{fail}|\widehat{B}=\widehat{b}, \widehat{A}=\widehat{a}, A=a) = 1-P(\text{success}|\text{states}) \tag{C.19}$$

where,

$$\begin{aligned}&P(\text{success}|\text{states})\\ &= P(\text{success of nodes in } \widehat{\mathcal{B}}|\widehat{B}=\widehat{b}, A=a)\cdot P(\text{success of nodes in } \mathcal{S}\backslash\{\mathcal{A}\bigcup\widehat{\mathcal{B}}\}|\widehat{B}=\widehat{b}, \widehat{A}=\widehat{a})\end{aligned}$$

Thus from equations (C.7), (C.8), (C.9) and (C.19), we have the probability of cycle failure if the channel changes between downlink phase 1 and uplink phase 1 to be:

$$\begin{aligned}&P(\text{fail, case 4})\\ &= \sum_{a=0}^{n-1}\sum_{\widehat{a}=0}^{a}\sum_{\widehat{b}=0}^{n-a-1} P(A=a)\cdot P(\widehat{A}=\widehat{a}|A=a)\cdot P(\widehat{B}=\widehat{b}|A=a)\cdot P(\text{fail}|\widehat{B}=\widehat{b}, \widehat{A}=\widehat{a}, A=a)\end{aligned} \tag{C.20}$$

The worst case behavior is contributed by the case which has the maximum probability of failure due to change in channel. Thus by combining equations (C.6), (C.14), (C.16), (C.20) we get,

$$\begin{aligned}&P(\text{fail due to non-quasi-static})\\ &= \max\{P(\text{fail, case 1}), P(\text{fail, case 2}), P(\text{fail, case 3}), P(\text{fail, case 4})\}\end{aligned}$$

C.2.2 XOR-CoW

We discuss what happens when channels change during a cycle in XOR-CoW protocol. Similar to the assumption we made in the case of Occupy CoW, we assume that changes

happen during the boundary between two phases. There are three possibilities: between XOR phase and downlink phase, between downlink phase and uplink phase, between uplink phase and XOR phase.

Case 1: Channels change between XOR phase and downlink phase

In this case, the channels change when a new cycle is about the begin. The probability of failure in this case is

$$P(\text{fail, case 1}) = \sum_{a=0}^{n-1} \left(\binom{n}{a} (1-p_x)^a p_x^{n-a} \right) \left(1 - (1-p_x^a)^{n-a}\right) \tag{C.21}$$

Case 2: Channels change between downlink phase and uplink phase

In this case, the channels change after downlink phase ends and then remain constant through uplink and XOR phases. Let the nodes which succeeded in downlink phase be the set $\mathcal{A}$. When the channels change before uplink phase, some of the nodes in the set $\mathcal{A}$ lose their connection to the controller (set $\check{\mathcal{A}}$) and some of them retain it (set $\widehat{\mathcal{A}}$). Some nodes in the set $\mathcal{S}\backslash\mathcal{A}$ gain link to the controller (set $\widehat{\mathcal{B}}$). The ways to succeed are listed below:

- In downlink phase, nodes in set $\mathcal{A}$ succeed in getting downlink information. $\mathcal{A} = \widehat{\mathcal{A}} \bigcup \check{\mathcal{A}}$.
- In uplink phase, nodes in set $\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}$ succeed in conveying their uplink information to the controller.
- In XOR phase, the following are the ways to succeed:
  1. Nodes in $\widehat{\mathcal{B}}$ directly through the controller.
  2. Nodes in $\mathcal{S}\backslash\{\widehat{\mathcal{A}} \bigcup \widehat{\mathcal{B}}\}$ succeed if they have a link to $\widehat{\mathcal{A}}$ in the uplink phase (thus retaining the link in XOR phase).

Thus, the probability that $A = a$ is given by

$$P(A = a) = \binom{n}{a} (1-p_x)^a p_x^{n-a} \tag{C.22}$$

Conditioned on $A = a$, the probability of $\widehat{A} = \widehat{a}$ is given by

$$P(\widehat{A} = \widehat{a} | A = a) = \binom{a}{\widehat{a}} (1-p_x)^{\widehat{a}} p_x^{a-\widehat{a}} \tag{C.23}$$

Conditioned on $A = a$, the probability of $\widehat{B} = \widehat{b}$ is given by

$$P(\widehat{B} = \widehat{b} | A = a) = \binom{n-a}{\widehat{b}} (1-p_x)^{\widehat{b}} p_x^{n-a-\widehat{b}} \tag{C.24}$$

The probability of success of the nodes in $\mathcal{S}\backslash\{\widehat{\mathcal{A}}\bigcup\widehat{\mathcal{B}}\}$ conditioned on the states is the probability that they connect to $\widehat{\mathcal{A}}$ in uplink phase. Thus we have,

$$P(\text{success}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a})=\left(1-p_x^{\widehat{a}}\right)^{n-\widehat{a}-\widehat{b}} \tag{C.25}$$

By equations (C.22), (C.23), (C.24) and (C.25) we have that the probability of cycle failure when the channel changes between downlink and uplink phase is given by

$$P(\text{fail, case 2})$$
$$=\sum_{a=0}^{n-1}\sum_{\widehat{a}=0}^{a}\sum_{\widehat{b}=0}^{n-a-1}P(A=a)\cdot P(\widehat{A}=\widehat{a}|A=a)\cdot P(\widehat{B}=\widehat{b}|A=a)\cdot(1-P(\text{success}|\widehat{B}=\widehat{b},\widehat{A}=\widehat{a})) \tag{C.26}$$

Case 3: Channels change between uplink phase and XOR phase
In this case, the channels change after uplink phase ends and then remain constant XOR phase. Let the nodes which succeeded in downlink phase be the set $\mathcal{A}$. The same nodes succeed in uplink phase. When the channels change before XOR phase, some of the nodes in the set $\mathcal{A}$ lose their connection to the controller (set $\check{\mathcal{A}}$) and some of them retain it (set $\widehat{\mathcal{A}}$).

- In the downlink phase, nodes in the set $\mathcal{A}$ succeed in getting downlink information.
- In the uplink phase, nodes in the set $\mathcal{A}$ succeed in getting their uplink information to the controller. $\mathcal{A}=\widehat{\mathcal{A}}\bigcup\check{\mathcal{A}}$.
- In the XOR phase, the only way to succeed is to have a bi-directional link to $\widehat{\mathcal{A}}$ in uplink and XOR phase.

Thus, the probability that $A=a$ is the same as Eq. (C.22). Conditioned on $A=a$, the probability of $\widehat{A}=\widehat{a}$ is the same as Eq. (C.23). The probability of success of the nodes in $\mathcal{S}\backslash\mathcal{A}$ conditioned on the states is the probability that they have a bidirectional link to $\widehat{\mathcal{A}}$. The probability that a bi-directional link does not exist (denoted by p_{bi}) between two nodes is the probability that either the link does not exist in one direction or both is given by

$$p_{bi}=1-(1-p_x)(1-p_x) \tag{C.27}$$

Thus we have,

$$P(\text{success}|\widehat{A}=\widehat{a})=\left(1-p_{bi}^{\widehat{a}}\right)^{n-a} \tag{C.28}$$

By combining equations (C.22), (C.23) and (C.28) we get that the probability of cycle failure when the channels change between uplink and XOR phase is given by

$$P(\text{fail, case 3})=\sum_{a=0}^{n-1}\sum_{\widehat{a}=0}^{a}P(A=a)\cdot P(\widehat{A}=\widehat{a}|A=a)\cdot(1-P(\text{success}|\widehat{A}=\widehat{a})) \tag{C.29}$$

The worst case behavior is contributed by the case which has the maximum probability of failure due to change in channel. Thus by combining equations (C.21), (C.26), (C.29) we get,

$$P(\text{fail due to non-quasi-static}) = \max\{P(\text{fail, case 1}), P(\text{fail, case 2}), P(\text{fail, case 3})\}$$

C.3 Effect of synchronization impediments

If we cap the number of transmitters can that simultaneously transmit at k_{max} due to synchronization requirements, then the probability of cycle failure in Occupy CoW protocol is given by

$$P(\text{fail}) = \sum_{a=0}^{n-1} \left(\binom{n}{a} (1-p_w)^a p_w^{n-a} \right) \left(1 - (1-p_w^{t_m})^{n-a}\right)$$

where $t_m = \min(a, k_{max})$. In the case of XOR-CoW protocol, the probability of cycle failure is given by

$$P(\text{fail}) = \sum_{a=0}^{n-1} \left(\binom{n}{a} (1-p_x)^a p_x^{n-a} \right) \left(1 - (1-p_x^{t_m})^{n-a}\right)$$

where $t_m = \min(a, k_{max})$.

C.4 Effect of pessimistic spatial correlation model

Every new channel has a probability q of coming from an independent fading distribution. However, with probability $1-q$ the channel is identical to a channel that has already been realized. Consider the Occupy CoW protocol for the rest of the discussion. After downlink phase 1, let set A be the set of nodes which successfully decoded the controller's message. For a node in $S\backslash A$, the probability that it can harvest diversity k is given by $\binom{a}{k}(q^k)((1-q)^{a-k})$. And given that it harvests diversity of k, the probability of success for that node is $1-p^k$. Thus,

$$\begin{aligned} P(\text{success for a node in } S\backslash A) &= \sum_{k=0}^{a} P(\text{success for a node in } S\backslash A \text{ harvesting diversity } k) \\ &= \sum_{k=0}^{a} P(\text{success for a node in } S\backslash A| \text{ diversity } = k) P(\text{diversity } = k) \\ &= \sum_{k=0}^{a} (1-p^k)(\binom{a}{k}(q^k)((1-q)^{a-k})) \end{aligned}$$

Then probability of success of the cycle conditioned on cardinality of set A is $P(\text{success for a node in } S\backslash A)^N$
Total failure probability is given by

$$P(fail) = \sum_{a=0}^{N-1} \binom{N}{a} ((1-q)^a (q^{N-a})) \times (\sum_{k=0}^{a} (1-p^k)(\binom{a}{k} (q^k)((1-q)^{a-k})))^{N-a}.$$

This can be easily extended for XOR-CoW by replacing the appropriate probabilities of error.

Bibliography

[1] M. Weiner, *Low-Latency, High-Reliability Wireless Networks for Control Applications*. PhD thesis, EECS Department, University of California, Berkeley, May 2015.

[2] G. Fettweis, "The Tactile Internet: Applications and Challenges," *Vehicular Technology Magazine, IEEE*, vol. 9, pp. 64–70, March 2014.

[3] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "Farmbeats: An iot platform for data-driven agriculture," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, (Boston, MA), pp. 515–529, 2017.

[4] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What Will 5G Be?," *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 1065–1082, June 2014.

[5] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5g-enabled tactile internet," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, 2016.

[6] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolic, "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*, pp. 3829–3835, 2014.

[7] "SERCOS news, the automation bus magazine," January 2014.

[8] "3G." https://en.wikipedia.org/wiki/3G.

[9] "LTE Advanced." https://en.wikipedia.org/wiki/LTE_Advanced.

[10] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge University Press, 2005.

[11] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[12] "Carrier-sense multiple access with collision avoidance." https://en.wikipedia.org/wiki/Carrier-sense_multiple_access_with_collision_avoidance.

[13] M.-S. Alouini and A. J. Goldsmith, "Area spectral efficiency of cellular mobile radio systems," *IEEE Transactions on vehicular technology*, vol. 48, no. 4, pp. 1047–1066, 1999.

[14] P. Grover, K. Woyach, and A. Sahai, "Towards a communication-theoretic understanding of system-level power consumption," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1744–1755, 2011.

[15] Technical Specification Group Radio Access Network, "Service requirements for the 5G system," *3GPP Technical Report 22.261, Release 16*, 2018.

[16] B. Farhang-Boroujeny, "OFDM versus filter bank multicarrier," *IEEE signal processing magazine*, vol. 28, no. 3, pp. 92–112, 2011.

[17] V. Vakilian, T. Wild, F. Schaich, S. ten Brink, and J.-F. Frigon, "Universal-filtered multi-carrier technique for wireless systems beyond LTE," in *Globecom Workshops (GC Wkshps), 2013 IEEE*, pp. 223–228, IEEE, 2013.

[18] N. Michailow, M. Matthé, I. S. Gaspar, A. N. Caldevilla, L. L. Mendes, A. Festag, and G. Fettweis, "Generalized frequency division multiplexing for 5th generation cellular networks," *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3045–3061, 2014.

[19] S. Eldessoki, D. Wieruch, and B. Holfeld, "Impact of Waveforms on Coexistence of Mixed Numerologies in 5G URLLC Networks," in *WSA 2017; 21th International ITG Workshop on Smart Antennas; Proceedings of*, pp. 1–6, VDE, 2017.

[20] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5G: physical and MAC-layer solutions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, 2016.

[21] T. Levanen, J. Pirskanen, T. Koskela, J. Talvitie, and M. Valkama, "Low latency radio interface for 5G flexible TDD local area communications," in *Communications Workshops (ICC), 2014 IEEE International Conference on*, pp. 7–13, June 2014.

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

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

[24] A. Mahmood, M. I. Ashraf, M. Gidlund, and J. Torsner, "Over-the-air time synchronization for urllc: Requirements, challenges and possible enablers," *arXiv preprint arXiv:1807.00078*, 2018.

[25] Technical Specification Group Radio Access Network, "Study on Scenarios and Requirements for Next Generation Access Technologies," *3GPP Technical Report 38.913, Release 14*, 2017.

[26] K. Manolakis, W. Xu, and G. Caire, "Synchronization signal design and hierarchical detection for the d2d sidelink," in *Signals, Systems, and Computers, 2017 51st Asilomar Conference on*, pp. 1650–1654, IEEE, 2017.

[27] G. Durisi, T. Koch, and P. Popovski, "Toward Massive, Ultrareliable, and Low-Latency Wireless Communication With Short Packets," *Proceedings of the IEEE*, vol. 104, pp. 1711–1726, August 2016.

[28] P. Popovski, "Ultra-reliable communication in 5G wireless systems," in *5G for Ubiquitous Connectivity (5GU), 2014 1st International Conference on*, pp. 146–151, IEEE, February 2014.

[29] A. Anand, G. de Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *arXiv preprint arXiv:1712.05344*, 2017.

[30] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View," *arXiv preprint arXiv:1804.05057*, 2018.

[31] A. E. Kalør, R. Guillaume, J. J. Nielsen, A. Mueller, and P. Popovski, "Network slicing for ultra-reliable low latency communication in industry 4.0 scenarios," *arXiv preprint arXiv:1708.09132*, 2017.

[32] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. Leung, "Network slicing based 5G and future mobile networks: mobility, resource management, and challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[33] K. Katsalis, N. Nikaein, E. Schiller, A. Ksentini, and T. Braun, "Network slices toward 5G communications: Slicing the LTE network," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 146–154, 2017.

[34] Alliance, NGMN, "5G white paper," *Next generation mobile networks, white paper*, pp. 1–125, 2015.

[35] Huawei, "5g security architecture white paper," 2017.

[36] Huawei, "5g scenarios and security design," 2016.

[37] K. Leppänen and W. Weigel, "5G trends and potential–an industry view,"

[38] A. Papathanassiou and A. Khoryaev, "Cellular V2X as the Essential Enabler of Superior Global Connected Transportation Services," vol. 1, June 2017.

[39] "Introduction to SERCOS III with industrial ethernet."

[40] V. Narasimha Swamy, S. Suri, P. Rigge, M. Weiner, G. Ranade, A. Sahai, and B. Nikolic, "Cooperative communication for high-reliability low-latency wireless control," in *Communications (ICC), 2015 IEEE International Conference on*, pp. 4380–4386, June 2015.

[41] V. Narasimha Swamy, P. Rigge, G. Ranade, A. Sahai, and B. Nikoli, "Network coding for high-reliability low-latency wireless control," in *2016 IEEE Wireless Communications and Networking Conference Workshop on The Tactile Internet: Enabling Technologies and Applications (IEEEWCNC2016-TACNET)*, (Doha, Qatar), Apr. 2016.

[42] V. Narasimha Swamy, S. Suri, P. Rigge, M. Weiner, G. Ranade, A. Sahai, and B. Nikolic, "Real-time cooperative communication for automation over wireless," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7168–7183, 2017.

[43] V. Narasimha Swamy, P. Rigge, G. Ranade, B. Nikolić, and A. Sahai, "Predicting wireless channels for ultra-reliable low-latency communications," in *2018 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2018.

[44] V. Narasimha Swamy, P. Rigge, G. Ranade, B. Nikolic, and A. Sahai, "Wireless channel dynamics and robustness for ultra-reliable low-latency communications," *arXiv preprint arXiv:1806.08777*, 2018.

[45] V. Narasimha Swamy, G. Ranade, and A. Sahai, "Robustness of cooperative communication schemes to channel models," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 2194–2198, IEEE, 2016.

[46] L. Dickstein, V. Narasimha Swamy, G. Ranade, and A. Sahai, "Finite block length coding for low-latency high-reliability wireless communication," in *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, pp. 908–915, IEEE, 2016.

[47] "Real-time Communication Systems For Automation Over Wireless: Enabling Future Interactive Tech." http://www.eecs.berkeley.edu/ sahai.

[48] C. Hoymann, D. Astely, M. Stattin, G. Wikstrom, J.-F. Cheng, A. Hoglund, M. Frenne, R. Blasco, J. Huschke, and F. Gunnarsson, "LTE release 14 outlook," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 44–49, 2016.

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

[50] Technical Specification Group Radio Access Network, "Physical Channels and Modulation," *3GPP Technical Report 38.211, Release 15*, 2018.

[51] Technical Specification Group Radio Access Network, "Multiplexing and channel coding," *3GPP Technical Report 38.212, Release 15*, 2018.

[52] Technical Specification Group Radio Access Network, "Physical layer procedures for control," *3GPP Technical Report 38.213, Release 15*, 2018.

[53] Technical Specification Group Radio Access Network, "Physical layer procedures for data," *3GPP Technical Report 38.214, Release 15*, 2018.

[54] 5G Automotive Association, "The Case for Cellular V2X for Safety and Cooperative Driving," *5GAA White Paper*, 2016.

[55] Dario Sabella and Hassnaa Moustafa and Pekka Kuure and Sami Kekki and Zheng Zhou and Alice Li and Christoph Thein and Edwin Fischer and Ivan Vukovic and John Cardillo and Valerie Young and Soo Jin Tan and Vince Park and Michaela Vanderveen and Stefan Runeson and Stefano Sorrentino, "Toward fully connected vehicles: Edge computing for advanced automotive communications," *5GAA White Paper*, 2017.

[56] J. J. Nielsen, R. Liu, and P. Popovski, "Ultra-reliable low latency communication (URLLC) using interface diversity," *IEEE Transactions on Communications*, 2017.

[57] J. J. Nielsen, R. Liu, and P. Popovski, "Optimized Interface Diversity for Ultra-Reliable Low Latency Communication (URLLC)," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

[58] J. Rao and S. Vrzic, "Packet duplication for urllc in 5g dual connectivity architecture," in *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*, pp. 1–6, IEEE, 2018.

[59] R. Jurdi, S. R. Khosravirad, and H. Viswanathan, "Variable-rate ultra-reliable and low-latency communication for industrial automation," in *Information Sciences and Systems (CISS), 2018 52nd Annual Conference on*, pp. 1–6, IEEE, 2018.

[60] L. Liu and W. Yu, "A D2D-based Protocol for Ultra-Reliable Wireless Communications for Industrial Automation," *IEEE Transactions on Wireless Communications*, 2018.

[61] Z. Hou, C. She, Y. Li, T. Q. Quek, and B. Vucetic, "Burstiness Aware Bandwidth Reservation for Uplink Transmission in Tactile Internet," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, 2018.

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

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

[64] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks," in *Proceedings of the IEEE*, vol. 93, pp. 1130–1151, June 2005.

[65] P. Zand, S. Chatterjea, K. Das, and P. J. M. Havinga, "Wireless Industrial Monitoring and Control Networks: The Journey So Far and the Road Ahead," *J. Sensor and Actuator Networks*, vol. 1, no. 2, pp. 123–152, 2012.

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

[67] P. Morel, A. Croisier, and J.-D. Decotignie, "Requirements for wireless extensions of a FIP fieldbus," in *1996 IEEE Conference on Emerging Technologies and Factory Automation*, vol. 1, pp. 116–122 vol.1, Nov 1996.

[68] D. Miorandi and S. Vitturi, "A wireless extension of PROFIBUS DP based on the Bluetooth radio system," *Ad Hoc Networks*, vol. 3, pp. 479–494, Jul 2005.

[69] G. Cena, A. Valenzano, and S. Vitturi, "Hybrid wired/wireless networks for real-time communications," *Industrial Electronics Magazine, IEEE*, vol. 2, pp. 8–20, Mar 2008.

[70] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, Mar 2002.

[71] A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, and A. Sangiovanni-Vincentelli, "System Level Design for Clustered Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 3, pp. 202–214, Aug 2007.

[72] M. A. Yigitel, O. D. Incel, and C. Ersoy, "QoS-aware MAC Protocols for Wireless Sensor Networks: A Survey," *Comput. Netw.*, vol. 55, pp. 1982–2004, June 2011.

[73] A. Willig, "Recent and Emerging Topics in Wireless Industrial Communications: A Selection," *IEEE Transactions on Industrial Informatics*, vol. 4, pp. 102–124, May 2007.

[74] G. Scheible, D. Dzung, J. Endresen, and J.-E. Frey, "Unplugged but connected [Design and implementation of a truly wireless real-time sensor/actuator interface]," *Industrial Electronics Magazine, IEEE*, vol. 1, pp. 25–34, July 2007.

[75] ZigBee Alliance Standard, *ZigBee PRO Specfication*, October 2007.

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

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

[78] V. Gungor and G. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 4258–4265, Oct 2009.

[79] A. Kim, F. Hekland, S. Petersen, and P. Doyle, "When HART goes wireless: Understanding and implementing the WirelessHART standard," in *IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 899–907, 2008.

[80] T. Watteyne, J. Weiss, L. Doherty, and J. Simon, "Industrial IEEE802. 15.4 e networks: Performance and trade-offs," in *Communications (ICC), 2015 IEEE International Conference on*, pp. 604–609, IEEE, 2015.

[81] Siemens AG, *Basics of Setting up an Industrial Wireless LAN*, 5 2013. Version 3.

[82] H. Trsek, *Isochronous Wireless Network for Real-time Communication in Industrial Automation*. Springer, 2016.

[83] A. Frotzscher, U. Wetzker, M. Bauer, M. Rentschler, M. Beyer, S. Elspass, and H. Klessig, "Requirements and current solutions of wireless communication in industrial automation," in *2014 IEEE International Conference on Communications Workshops (ICC)*, pp. 67–72, IEEE, 2014.

[84] D. K. Lam, H. Urabe, Y. Shinozaki, K. Yamaguchi, S. Morita, Y. Nagao, M. Kurosaki, and H. Ochi, "A novel wireless LAN protocol for factory automation control," in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pp. 1248–1253, IEEE, 2014.

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

[86] J. Laneman, D. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, pp. 3062–3080, Dec 2004.

[87] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. Part I. System description," *IEEE Transactions on Communications*, vol. 51, pp. 1927–1938, Nov 2003.

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

[89] F. Oggier, G. Rekaya, J. claude Belfiore, and E. Viterbo, "Perfect spacetime block codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 9, pp. 3885–3902, 2006.

[90] P. Elia, B. A. Sethuraman, and P. V. Kumar, "Perfect Space-Time Codes for Any Number of Antennas," *IEEE Trans. Inform. Theory*, vol. 53, no. 11, pp. 3853–3868, 2007.

[91] G. Wu, Z. Li, H. Wang, and W. Zou, "Selective Random Cyclic Delay Diversity for HARQ in Cooperative Relay," in *IEEE Wireless Communications and Networking Conference (WCNC), 2010*, pp. 1–6, April 2010.

[92] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: A Distributed Wireless Architecture for Exploiting Sender Diversity," in *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, (New York, NY, USA), pp. 171–182, ACM, 2010.

[93] W. S. Wong and R. W. Brockett, "Systems with finite communication bandwidth constraints I: State estimation problems," *Automatic Control, IEEE Transactions on.*, vol. 42, no. 9, pp. 1294–1299, 1997.

[94] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1056–1068, 2004.

[95] A. Sahai and S. Mitter, "The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link part I: scalar systems," *IEEE Transactions on Information Theory*, vol. 52, no. 8, 2006.

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

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

[98] S. Park and A. Sahai, "Intermittent Kalman filtering: eigenvalue cycles and nonuniform sampling," in *American Control Conference (ACC).*, pp. 3692–3697, 2011.

[99] G. Ramnarayan, G. Ranade, and A. Sahai, "Side-information in control and estimation," in *2014 IEEE International Symposium on Information Theory*, pp. 171–175, IEEE, 2014.

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

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

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

[103] P. Park, J. Araújo, and K. H. Johansson, "Wireless networked control system co-design," in *Networking, Sensing and Control (ICNSC), 2011 IEEE International Conference on*, pp. 486–491, IEEE, 2011.

[104] C. L. Robinson and P. Kumar, "Optimizing controller location in networked control systems with packet drops," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 661–671, 2008.

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

[106] N. Elia, "Remote stabilization over fading channels," *Systems & Control Letters*, vol. 54, no. 3, pp. 237–249, 2005.

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

[108] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.

[109] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking (ToN)*, vol. 16, no. 3, pp. 497–510, 2008.

[110] S. Bagheri, F. Verde, D. Darsena, and A. Scaglione, "Randomized decode-and-forward strategies for two-way relay networks," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 12, pp. 4214–4225, 2011.

[111] G. Wu, Z. Li, H. Wang, and W. Zou, "Selective Random Cyclic Delay Diversity for HARQ in Cooperative Relay," in *IEEE Wireless Communications and Networking Conference (WCNC), 2010*, pp. 1–6, April 2010.

[112] M. Sybis, K. Wesolowski, K. Jayasinghe, V. Venkatasubramanian, and V. Vukadinovic, "Channel coding for ultra-reliable low-latency communication in 5G systems," in *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th*, pp. 1–5, IEEE, 2016.

[113] M. Shirvanimoghaddam, M. S. Mohamadi, R. Abbas, A. Minja, B. Matuz, G. Han, Z. Lin, Y. Li, S. Johnson, and B. Vucetic, "Short Block-length Codes for Ultra-Reliable Low-Latency Communications," *arXiv preprint arXiv:1802.09166*, 2018.

[114] A. Sharma and M. Salim, "Polar code: The channel code contender for 5g scenarios," in *Computer, Communications and Electronics (Comptelix), 2017 International Conference on*, pp. 676–682, IEEE, 2017.

[115] P. Wu and N. Jindal, "Coding versus ARQ in fading channels: how reliable should the PHY be?," *IEEE Transactions on Communications*, vol. 59, no. 12, pp. 3363–3374, 2011.

[116] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[117] Y.-W. Huang and P. Moulin, "Finite blocklength coding for multiple access channels," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pp. 831–835, IEEE, 2012.

[118] C. P. Lee and H. J. Su, "Dynamic decode and forward for the multi-access relay channel with finite block length," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1825–1829, Sept 2011.

[119] C. Lee and H. Su, "Diversity-multiplexing tradeoff in the multiaccess relay channel with finite block length," *CoRR*, vol. abs/1103.3719, 2011.

[120] Y. Hu, J. Gross, and A. Schmeink, "The impact of outdated CSI on the finite blocklength performance of relaying," *CoRR*, vol. abs/1606.06490, 2016.

[121] Y. Li, M. C. Gursoy, and S. Velipasalar, "Throughput of two-hop wireless channels with queueing constraints and finite blocklength codes," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 2599–2603, July 2016.

[122] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Diversity versus channel knowledge at finite block-length," in *Information Theory Workshop (ITW), 2012 IEEE*, pp. 572–576, Sept 2012.

[123] Y. Hu, A. Schmeink, and J. Gross, "Blocklength-limited performance of relaying under quasi-static rayleigh channels," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 4548–4558, July 2016.

[124] Y. Hu, J. Gross, and A. Schmeink, "On the capacity of relaying with finite blocklength," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1790–1794, 2016.

[125] V. Kostina and S. Verdu, "Fixed-length lossy compression in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 58, pp. 3309–3338, June 2012.

[126] G. J. Bradford and J. N. Laneman, "Rate, reliability, and delay tradeoffs for decode-and-forward relaying,"

[127] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Transactions on Information Theory*, vol. 60, pp. 4232–4265, July 2014.

[128] E. MolavianJazi and J. N. Laneman, "A finite-blocklength perspective on gaussian multi-access channels," *CoRR*, vol. abs/1309.2343, 2013.

[129] Y. Hu, M. Serror, K. Wehrle, and J. Gross, "Finite blocklength performance of multi-terminal wireless industrial networks," *CoRR*, vol. abs/1606.08646, 2016.

[130] K. R. Schaubach, N. Davis, and T. S. Rappaport, "A ray tracing method for predicting path loss and delay spread in microcellular environments," in *Vehicular Technology Conference, 1992, IEEE 42nd*, pp. 932–935, IEEE, 1992.

[131] C. R. Anderson, T. S. Rappaport, K. Bae, A. Verstak, N. Ramakrishnan, W. H. Tranter, C. A. Shaffer, and L. T. Watson, "In-building wideband multipath characteristics at 2.5 and 60 ghz," in *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, vol. 1, pp. 97–101, IEEE, 2002.

[132] J.-E. Berg, R. Bownds, and F. Lotse, "Path loss and fading models for microcells at 900 mhz," in *Vehicular Technology Conference, 1992, IEEE 42nd*, pp. 666–671, IEEE, 1992.

[133] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*, vol. 2. Prentice Hall PTR New Jersey, 1996.

[134] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient backscatter: wireless communication out of thin air," in *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 39–50, ACM, 2013.

[135] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Cutting the cord in virtual reality," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pp. 162–168, ACM, 2016.

[136] R. C. Daniels and R. W. Heath Jr, "60 ghz wireless communications: Emerging requirements and design recommendations," *IEEE Vehicular technology magazine*, vol. 2, no. 3, 2007.

[137] V. Va, T. Shimizu, G. Bansal, R. W. Heath Jr, *et al.*, "Millimeter wave vehicular communications: A survey," *Foundations and Trends® in Networking*, vol. 10, no. 1, pp. 1–113, 2016.

[138] T. S. Rappaport, G. R. MacCartney, M. K. Samimi, and S. Sun, "Wideband millimeter-wave propagation measurements and channel models for future wireless communication system design," *IEEE Transactions on Communications*, vol. 63, no. 9, pp. 3029–3056, 2015.

[139] W. C. Jakes and D. C. Cox, *Microwave mobile communications*. Wiley-IEEE Press, 1994.

[140] E. Baştuğ, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *arXiv preprint arXiv:1405.5974*, 2014.

[141] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[142] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *Communications (ICC), 2014 IEEE International Conference on*, pp. 1897–1903, IEEE, 2014.

[143] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, 2017.

[144] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Communications*, vol. 25, no. 3, 2018.

[145] P. Sandhir and K. Mitchell, "A neural network demand prediction scheme for resource allocation in cellular wireless systems," in *Region 5 Conference, 2008 IEEE*, pp. 1–6, IEEE, 2008.

[146] J. Jiang, R. Das, G. Ananthanarayanan, P. A. Chou, V. Padmanabhan, V. Sekar, E. Dominique, M. Goliszewski, D. Kukoleca, R. Vafin, *et al.*, "Via: Improving internet telephony call quality using predictive relay selection," in *Proceedings of the 2016 ACM SIGCOMM Conference*, pp. 286–299, ACM, 2016.

[147] Y. Wang, M. Martonosi, and L.-S. Peh, "Predicting link quality using supervised learning in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 3, pp. 71–83, 2007.

[148] S. Haykin *et al.*, "Cognitive radio: brain-empowered wireless communications," *IEEE journal on selected areas in communications*, vol. 23, no. 2, pp. 201–220, 2005.

[149] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of machine learning to cognitive radio networks," *IEEE Wireless Communications*, vol. 14, no. 4, 2007.

[150] A. Forster, "Machine learning techniques applied to wireless ad-hoc networks: Guide and survey," in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pp. 365–370, IEEE, 2007.

[151] P. Valente Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self organizing cellular networks," *IEEE Communications Surveys and Tutorials*, 2017.

[152] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.

[153] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[154] S.-Y. Lien, S.-C. Hung, D.-J. Deng, and Y. J. Wang, "Efficient ultra-reliable and low latency communications and massive machine-type communications in 5g new radio," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–7, IEEE, 2017.

[155] T. Chen, Q. Ling, Y. Shen, and G. B. Giannakis, "Heterogeneous online learning for thing-adaptive fog computing in iot," *IEEE Internet of Things Journal*, 2018.

[156] N. Strodthoff, B. Göktepe, T. Schierl, C. Hellge, and W. Samek, "Enhanced machine learning techniques for early harq feedback prediction in 5g," *arXiv preprint arXiv:1807.10495*, 2018.

[157] C. de Vrieze, S. Barratt, D. Tsai, and A. Sahai, "Cooperative multi-agent reinforcement learning for low-level wireless communication," *arXiv preprint arXiv:1801.04541*, 2018.

[158] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[159] Z. Shen, J. G. Andrews, and B. L. Evans, "Short range wireless channel prediction using local information," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, vol. 1, pp. 1147–1151, IEEE, 2003.

[160] L. S. Muppirisetty, T. Svensson, and H. Wymeersch, "Spatial wireless channel prediction under location uncertainty," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1031–1044, 2016.

[161] Q. Huang, M. Ghogho, J. Wei, and P. Ciblat, "Practical Timing and Frequency Synchronization for OFDM-Based Cooperative Systems," *IEEE Transactions on Signal Processing*, vol. 58, pp. 3706–3716, July 2010.

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

[163] S. Hanly and D. Tse, "Multiaccess fading channels. II. Delay-limited capacities," *IEEE Transactions on Information Theory*, vol. 44, pp. 2816–2831, Nov 1998.

[164] L. Ozarow, S. Shamai, and A. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 359–378, May 1994.

[165] A. Lozano and D. Porrat, "Non-peaky signals in wideband fading channels: Achievable bit rates and optimal bandwidth," *Wireless Communications, IEEE Transactions on*, vol. 11, pp. 246–257, January 2012.

[166] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-Static Multiple-Antenna Fading Channels at Finite Blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, 2014.

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

[168] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 2349–2360, Sept 2005.

[169] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[170] Z. Li and B. Li, "Network coding in undirected networks," CISS, 2004.

[171] T. S. Han and S. Verdu, "New results in the theory of identification via channels," *IEEE transactions on information theory*, vol. 38, no. 1, pp. 14–25, 1992.

[172] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2467–2485, 2006.

[173] V. Narasimha Swamy, S. Bhashyam, R. Sundaresan, and P. Viswanath, "An asymptotically optimal push–pull method for multicasting over a random network," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 5075–5087, 2013.

[174] M. F. Pop and N. C. Beaulieu, "Limitations of sum-of-sinusoids fading channel simulators," *IEEE Transactions on communications*, vol. 49, no. 4, pp. 699–708, 2001.

[175] M. J. Gans, "A power-spectral theory of propagation in the mobile-radio environment," *IEEE Transactions on Vehicular Technology*, vol. 21, no. 1, pp. 27–38, 1972.

[176] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1650–1662, 2005.

[177] X. Li, Y.-C. Wu, and E. Serpedin, "Timing synchronization in decode-and-forward cooperative communication systems," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1444–1455, 2009.

[178] S. Verdú, "Spectral efficiency in the wideband regime," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1319–1343, 2002.

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

[180] Y. Jing and H. Jafarkhani, "Single and multiple relay selection schemes and their achievable diversity orders," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1414–1423, 2009.

[181] "Marcum Q-function." http://mathworld.wolfram.com/MarcumQ-Function.html.

[182] A. Bletsas, A. Lippnian, and D. P. Reed, "A simple distributed method for relay selection in cooperative diversity wireless networks, based on reciprocity and channel measurements," in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 3, pp. 1484–1488, IEEE, 2005.

[183] A. S. Ibrahim, A. K. Sadek, W. Su, and K. R. Liu, "Cooperative communications with relay-selection: when to cooperate and whom to cooperate with?," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, 2008.

[184] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[185] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[186] S. Shah, A. Kapoor, D. Dey, and C. Lovett, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," *Field and Service Robotics*, pp. 621–635, November 2017.

[187] M. Serror, S. Vaaßen, K. Wehrle, and J. Gross, "Practical evaluation of cooperative communication for ultra-reliability and low-latency,"

[188] S. Graham and P. Kumar, "Time in general-purpose control systems: The control time protocol and an experimental evaluation," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 4, pp. 4004–4009, IEEE, 2004.

[189] R. Solis, V. Borkar, and P. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 2734–2739, IEEE San Diego, USA, 2006.

[190] N. M. Freris, S. R. Graham, and P. Kumar, "Fundamental limits on synchronizing clocks over networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1352–1364, 2011.

[191] Parker, "6K Controllers."