Low-Latency, High-Reliability Wireless Networks for Control Applications

Matthew Weiner

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-114
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-114.html

May 14, 2015

Low-Latency, High-Reliability
Wireless Networks for Control Applications

by

Matthew Geoffrey Weiner

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Borivoje Nikolić, Chair
Professor Anant Sahai
Professor Paul Wright

Spring 2015

Low-Latency, High-Reliability
Wireless Networks for Control Applications

Abstract

Low-Latency, High-Reliability
Wireless Networks for Control Applications

by

Matthew Geoffrey Weiner

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Borivoje Nikolić, Chair

In the near future, the number of wireless devices will outnumber humans by an order of magnitude, and most of these devices will communicate with each other instead of people. They will not only sense the environment, as most do today, but they will also manipulate it. This closed loop operation will not require the high data rates that today's people-centric networks provide; instead it will require low-latency and high-reliability communication at moderate overall data rates. Currently, high-performance industrial control is one of the few applications that has similar requirements to the IoT of the future. However, control systems exclusively use wired networks because existing wireless systems and standards cannot achieve the latency and reliability required since they are designed for either high-throughput or low-power communication between a small number of terminals.

The first part of this work focuses on the design and evaluation of a wireless system architecture appropriate for high-performance control systems with a large number of sensors and actuators. Based on a model and representative set of specifications, an initial wireless system architecture is developed that is aimed at addressing the issues current WLAN and cellular systems have with supporting low-latency and high-reliability operation for a large number of users. The initial architecture's achievable latency is a strong function of the available diversity, so it requires a scheme to generate the diversity in a low-latency manner without relying heavily on the channels provided by nature. One option is to use multiple, cooperating access points that are distributed around the system, as is done with coordinated multipoint in cellular systems. This has limited usefulness in many IoT and control systems since additional infrastructure is not possible or desirable, especially in the likely case that the access points are connected with wires. As an alternative option, a cooperative relaying system is developed that is based on decentralized relaying with semi-scheduled transmissions where relays can transmit simultaneously using a distributed space-time code, such as cyclic delay diversity. The proposed cooperative relaying system and other baseline schemes are analyzed using a simplified link level analysis, and the proposed system significantly outperforms the other schemes.

The second part of this work looks at aspects of implementing the physical layer of the proposed cooperative relaying system architecture, including the analog front end, modulation, baseband processing, and multiple access protocol. An emphasis is put on reusing as many blocks from current systems as possible.

Low-latency systems require efficient hardware, and error control decoders are an essential part of high-reliability wireless systems. The third part of this work looks at the implementation of a low-latency, low-power LDPC decoder for the IEEE 802.11ad standard, whose LDPC codes have many features applicable to wireless control. The decoder's highly parallel, deeply pipelined, flooding architecture balances latency and power. Row-merging, simultaneous processing of multiple codewords, reduced marginalization memory precision, and using back-biasing to optimally trade off active and leakage power further reduce latency and power. The decoder is implemented in a 28nm FDSOI technology, has sub-microsecond level latency, and consumes only 6.2mW at a throughput of 1.5Gb/s and 38.1mW at 6Gb/s.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my research advisor Professor Borivoje Nikolić for his guidance, support, and understanding during my time at Berkeley. He is always looking out for my best interests and sharing his experience with me. I want to thank Professor Anant Sahai and Professor Venkat Anantharam for supporting this work with their valuable insights and helping me understand tricky theoretical concepts with clear and intuitive explanations. I am also grateful to Jan Rabaey for reviewing this research proposal and sharing his views on the potential and directions for this research and to Paul Wright for reviewing my research proposal and this dissertation. A special thanks goes to my colleague Milos Jorgovanović for our extensive and successful collaboration. I also thank Milovan Blagojević, Sergey Skotnikov, Ana Klimović, Gireeja Ranade, Vasuki Swamy, and Paul Rigge for their valuable contributions to this research.

I feel very fortunate to have conducted my research in the Berkeley Wireless Research Center, which is one of the finest academic environments that I have encountered thanks to its excellent lab spaces, culture of open collaboration, faculty, and staff. I would like to thank Gary Kelson, Dave Allstot, Tom Boot, Leslie Nishiyama, Olivia Nolan, Sarah Jordan, and Bira Coelho for keeping the BWRC such a pleasant place to work. Additionally, I am grateful to Brian Richards and Fred Burghardt for their assistance with the CAD tools and the lab. My research was funded by the NSF Graduate Research Fellowship and NSF grants, and chip fabrication was donated by STMicroelectronics and aided by Philippe Flatresse.

My friends in the DCDG/COMIC research group made my time at Berkeley a memorable one. I would like to thank all of the members and alumni of the group, especially Sharon Xiao, Zhengya Zhang, Vinayak Nagpal, Katerina Papdoupoulou, Ruzica Jevtic, Zheng Guo, Renaldi Winoto, Seng Oon Toh, Ji-Hoon Park, Dusan Stepanovic, Lauren Jones, Jaehwa Kwak, Charles Wu, Olivier Thomas, Brian Zimmer, Ben Keller, Stevo Bailey, Nick Sutdarja, Sameet Ramakrishnan, Angie Wang, Pi-Feng Chiu, Rachel Hochman, Amanda Pratt, Luis Esteban Hernandez, Kathy Sun, Dajana Danilović, and Nicolas Le Dortz. I would also like to thank my friends outside my research group, including Shangliang Jiang, Yusef Shafi, Alan Malek, Yuanyuan Pao, Matthew Spencer, Steven Callender, Rikky Muller, Wen Li, Mitchell Kline, and Ana Ferreira.

I had two excellent internships at Marvell Semiconductor and Intel Labs. My supervisors, Engling Yeo and Jim Tschanz, helped make both of them a success and continue to be valuable mentors. I would also like to acknowledge Farhana Sheikh, Keith Bowman, and Arijit Raychowdhury for their guidance during my internship at Intel and beyond.

Finally, I would like to thank my wife Paula for encouraging me throughout my graduate studies and helping me get over the struggles I faced. I can only hope that I am as supportive to her as she has been to me. Also, I am grateful to my parents Andrew and Terry, my brother Joey, and my sister Mindy for always being there when I needed them.

Chapter 1

Introduction

1.1 Motivation

The explosion in the number and capability of consumer mobile devices, particularly smartphones, has fueled an insatiable demand for higher data rates. To increase throughput and deal with limits on available spectrum, the goal has been to maximize the spectral efficiency of wireless systems using information theoretic tools. These efforts have been extremely successful as evidenced by the continued growth of data rates in wireless local area network (WLAN) and cellular standards (Figure 1.1). However, these gains have come at the cost of secondary system parameters, such as latency, that do not fit directly into information theory's current framework. This is seen as an acceptable tradeoff as long as user experience is not impacted.

At the same time, the need for wireless networks that prioritize low-power operation over throughput has been steadily growing. Many techniques have been developed to boost energy efficiency, including aggressive duty cycling and multi-hop networking protocols. Again, the resulting efficiency gains have come at the cost of secondary parameters, particularly latency.

Because of the explosive growth of these markets, wireless devices have penetrated almost every aspect of our daily lives and are quickly becoming indispensable. Soon there will be a huge number of ubiquitously distributed, mobile embedded systems and access devices that will communicate both with each other and with the cloud forming an Internet of Things (IoT). This will open the door for truly immersive computing paradigms where wireless devices move beyond only sensing the environment; they will also be wirelessly connected to actuators that can manipulate the surrounding environment. In many instances, the sensors and actuators will operate in control loops with varying degrees of latency requirements. Examples of such systems include the smart grid, smart homes, industrial process automation, robotics, autonomous vehicles, telepresence, and immersive gaming.

Because of this trend, the bulk of wireless traffic will move away from being people-centric where a relatively small number of users per access point intermittently consume and generate large amounts of data, such as audio and video. Instead, it will move towards being

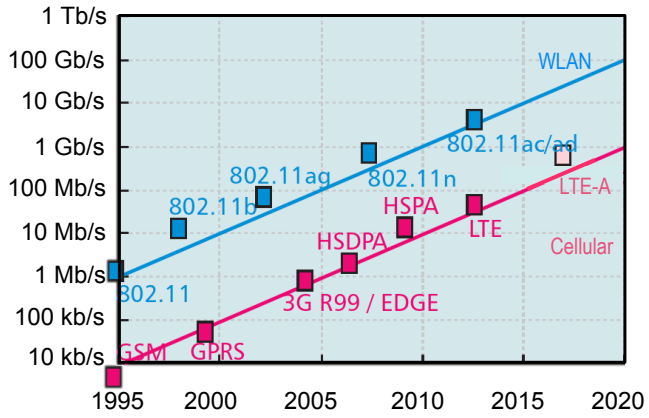


Figure 1.1: Evolution of the data rates of WLAN and cellular standards [1].

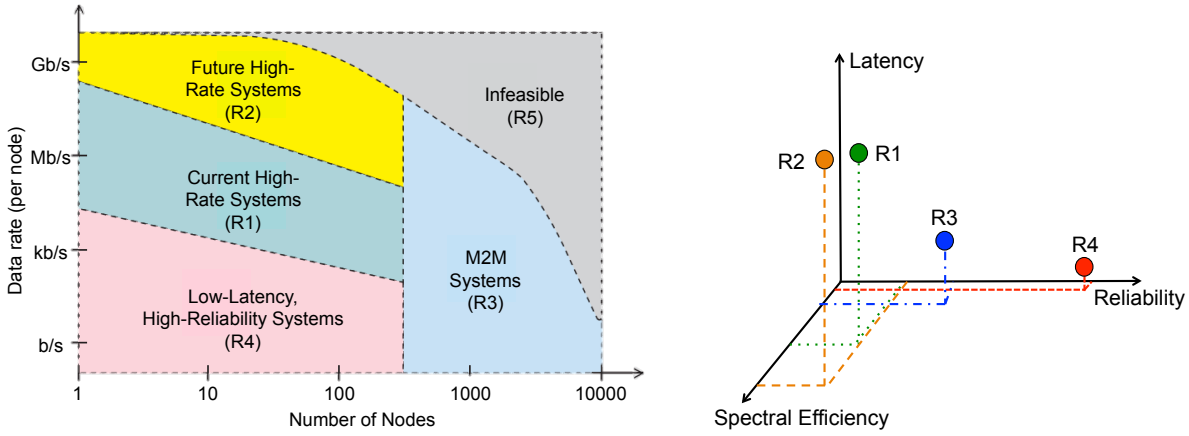


Figure 1.2: People-centric systems (R1 and R2) require increasingly large data rates, while device-centric systems require low-latency, high-reliability, and support for many nodes (R3 and R4). These specifications are shown (a) in terms of data rate and number of nodes (at a finite bandwidth, meaning there are information theoretic limits on the feasible specifications - R5 indicates the infeasible region) [2] and (b) in terms of spectral efficiency, latency, and reliability.

device-centric where thousands of users per access point regularly transmit small amounts of data reliably within a guaranteed time interval. The two models occupy very different regions of the design space (Figure 1.2), and supporting device-centric communication will require a departure from traditional throughput-oriented and energy-oriented system designs, particularly in regards to reducing latency and increasing reliability when there are a large number of active users.

Some wireless solutions exist for device-centric systems that have looser latency con-

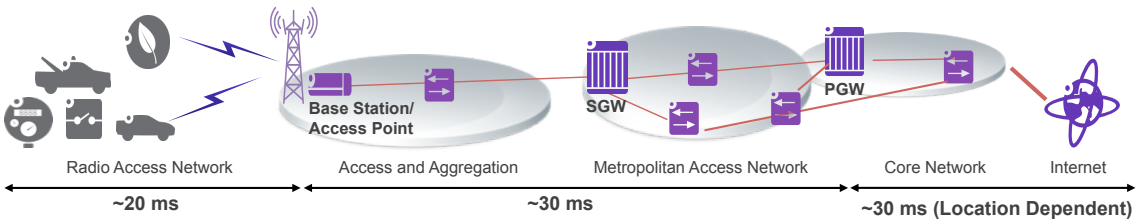


Figure 1.3: A breakdown of the average latency of a modern cellular network (LTE) between the core, metro, and radio access components for a single user [3].

straints, on the order of seconds to minutes, but only wired solutions exist for those requiring latencies on the order of milliseconds. Wired systems have a number of issues in practice, such as mechanical failure, weight, cost, and deployment complexity; therefore, a wireless system catering to low-latency and high-reliability communication between many nodes must be developed.

The first step involves rethinking the network structure. Figure 1.3 shows a breakdown of the latency in a cellular network, which is similar to that in other types of wireless networks [3]. The metropolitan access and core networks connect a wireless access point to the Internet, and the wireless access point connect all wireless terminals together. The propagation delay through fiber optic cables and the switching delay of routers limit the minimum latency of the core and metropolitan access networks to varying degrees, and this latency is difficult to decrease because the propagation delay is set by the material properties of the fiber optics, the network switches are extremely efficient, and the network topology is controlled by the service providers. On the other hand, multiple access overhead and retransmissions required to meet the (modest) reliability targets limit the radio access network's latency, which can be addressed with alternative system architectures.

Therefore, wireless communication systems for control should eliminate the metropolitan access and core networks and use only a local radio access network to connect devices. Still, 20ms for a transmission to a single device is far too long for high performance control, so alternative system architectures are required. These systems must overcome many challenges related to the wireless medium, such as fading, interference, bandwidth restrictions, and the inefficient existing protocols for multiple access. Many have been proposed ([4–8]), but none meet the unique requirements of high-performance wireless control.

1.2 Related Work

Reliable communication is a fundamental problem in information theory, and as such it has been well-studied. Shannon's landmark paper showed that it is possible to communicate with vanishing probability of error at non-zero rates [9], but it did not say how this could be practically achieved. The next several decades saw the advent of many error control coding (ECC) schemes for point-to-point communication. These efforts culminated in the

discovery of capacity-achieving codes, such as Turbo codes [10], low-density parity-check (LDPC) codes [11, 12], and Polar codes [13]. For wireless fading channels, diversity schemes were developed to deal with the deep fades that occur from multipath effects since ECC alone is not enough. Reliability has also been studied from the networking perspective to complement the techniques used on at physical layer (PHY). Some of the most popular schemes used are automatic-repeat-request (ARQ) and hybrid-ARQ (HARQ) on the media access control layer (MAC) and erasure coding coupled with ARQ on the upper layers. In all of the above examples, reliability is usually increased at the cost of latency because either longer blocklengths or retransmissions are used.

Low-latency communication has been studied at the PHY level in terms of the throughput-delay tradeoff curve [14], which has been developed for many classes of networks, but the protocols are not always practical or applicable for device-centric system specifications. Other theoretical investigations include the delay-limited capacity of a link [15] and the effective capacity of a network [16]. Signal processing engineers have lowered latency by simplifying their algorithms and implementing them efficiently in hardware. Networking engineers prioritized classes of data streams to lower the latency of quality-of-service (QoS) sensitive data at the expense of other types of data. Also, they have tried to simplify the networking stack to reduce overhead. In almost all cases, the solutions focus on lowering average latency instead of worst-case latency.

The combination of reliable and low-latency communication has mainly been studied in the context of control and industrial automation systems. On the theoretical side, researchers have examined how to change control algorithms to cope with the latency introduced by communication systems, ranging from using a modified form of optimal control to using non-uniform or event-triggered sampling [17–20]. On the implementation side, there have been a variety of wired networks developed starting with the MIL STD 1533 bus in 1970, which was followed by PROFIBUS in 1989 and SERCOS III in 2003. SERCOS III is an extremely reliable and low-latency communication network based on the FastEthernet standard that is used almost ubiquitously in industrial control applications.

As with traditional networks, there has been much interest in transitioning from wired to wireless in industrial control scenarios in order to eliminate expensive and bulky cabling and make modifications to the network easier. To do this, the WirelessHART standard was developed for low-performance control systems (with latency constraints on the order of seconds) [21]. To achieve lower latency, there has been interest in determining the performance of control systems using existing high data-rate wireless standards [22–25] and modifying those standards to increase performance for applications such as the smart grid, VoIP, and M2M communication [26–29]. Additionally, the wireless sensor and actor network (WSAN) community has developed numerous MAC protocols that have guaranteed latency bounds and acceptable reliability but that are geared for fast-fading channels and for systems with latency constraints on the order of seconds or minutes [30].

No work has been done on combining latency and reliability into a theoretical framework, although the groundwork has been laid by Polyanskiy's development of bounds on block error rates for finite blocklength codes [31]. Also, no wireless communication systems have been

proposed for systems with latency constraints on the order of milliseconds with tens to hundreds of nodes and with system reliability requirements of 10^{-6} or smaller.

1.3 Scope of Work

Traditional communication systems focus on delivering high data rates to a small number of people in order to support the streaming of multimedia content or the transfer large files. However, the majority of future users will be intelligent devices interacting autonomously. When the users are part of a closed-loop system that senses and manipulates the environment, such as a high-performance control system, they will require low-latency, high-reliability networks. Each device's individual messages will be short in length, but the overall data rate will be moderate due to the number of devices.

Current wireless system architectures can support control systems with tens of nodes and latency constraints on the order of seconds with moderate reliability, but they do not scale well to systems with tighter constraints or more nodes. Only wired networks have been able to meet the constraints of the highest-performance control systems, but they have many mechanical issues that limit their usefulness. The goal of this work is to develop a wireless network that can act as a drop-in replacement for the currently used wired networks. This allows the control law and sensor/actuator hardware to remain the same; all that needs to be replaced is the wired network, which is usually made out of commodity components. The focus is on centralized control systems with a single controller, but the results can be extended to more general systems.

This goal will be addressed at three levels. First, a low-latency, high-reliability wireless system architecture for control applications will be developed at the network and MAC levels. It will include a high-level analysis of its performance and a method to choose architectural parameters based on network specifications. Second, the physical layer implementation will be discussed with an emphasis on using blocks previously developed for existing standards to avoid needing to engineer the system from scratch. There is no bit-level simulation or hardware implementation. Finally, the design and implementation of a low-latency, low-power LDPC decoder that can be used in the physical layer will be discussed as a case study in how to design any custom blocks that are required in the physical layer.

1.4 Organization

Chapter 2 reviews the fundamentals of digital communication, including its history, descriptions of key blocks in a transceiver, and models for wired and wireless channels. Armed with these concepts, several communication standards are described, such as those used for Ethernet, consumer Internet access, WLANs, cellular networks, and ad-hoc networks. The latency and reliability limitations for each standard are identified.

Chapter 3 introduces high-performance control systems and the strict latency and reliability requirements of their communication networks. It describes and evaluates a wireless system architecture aimed at addressing the issues in current WLAN and cellular systems and finds that diversity is the key to lowering the system's latency (as long as generating the diversity does not add significantly to the latency). This leads to a cooperative relaying system based on simultaneous transmissions by all relays that can meet the low-latency, high-reliability specifications of high-performance control systems.

Chapter 4 discusses implementation aspects of the PHY layer for the cooperative relaying system architecture. It looks at the minimum changes required to existing blocks in current WLAN and cellular PHYs so that the wireless control PHY does not need to be completely re-engineered.

Chapter 5 details the implementation of a low-latency, low-power, flexible LDPC decoder for the IEEE 802.11ad standard that could be used in a wireless control system PHY. It covers the details of the LDPC decoding algorithm, the highly-parallel flooding decoder architecture, and the architectural and power optimizations. The decoder was implemented in 28nm FDSOI and achieves sub-microsecond latencies with a throughput of 6Gb/s and power consumption of 38.1mW.

Chapter 6 summarizes the key contributions of this work and gives future research directions at the protocol (MAC) and implementation (PHY) levels.

Chapter 3 contains material that appeared in [32] and [33] and is the result of collaboration with Milos Jorgovanović, Vasuki Swamy, Paul Rigge, Gireeja Ranade, and Sahaana Suri. Chapter 4 contains joint work with Paul Rigge. Chapter 5 contains material that appeared in [34] and [35] and is the result of collaboration with Sergey Skotnikov, Milovan Blagojević, Andreas Burg, Philippe Flatresse, and Zhengya Zhang.

Chapter 2

Latency and Reliability in Communication Systems

To understand the limits on latency and reliability that can be practically achieved, this chapter reviews some basics of digital communication and then explores several widely used system architectures and protocols for digital communication systems. In particular, it looks at the Ethernet, SERCOSIII, DSL, and DOCSIS standards for wired networks and the IEEE 802.11ac, IEEE 802.11ad, LTE, and IEEE 802.15.4-based standards for wireless networks. The Ethernet, DSL, DOCSIS, IEEE 802.11ac/ad, and LTE networks were chosen because they represent the majority of wired and wireless networks and would likely have their blocks used in any newly developed system architecture. The SERCOSIII and IEEE 802.15.4-based networks were chosen because they are currently used for high-performance and low-performance industrial control, respectively. From this exploration, the sources of latency and limits to reliability are determined.

2.1 Digital Communication

2.1.1 History

Communication is the process of transmitting information from a source across a physical medium and reconstructing that information correctly at a sink. The information can be transmitted in an analog or digital fashion. Analog communication usually involves modulating a carrier signal's amplitude or frequency based on the source signal. Examples of such systems are traditional telephony and AM/FM radios. On the other hand, digital communication converts the source signal into symbols and sends a specific signal for each symbol value. Examples of digital systems are plentiful and include the telegraph and any packet switching network. Both forms of communication were common before the development of information theory and the transistor, but since that time, digital communication has dominated and is an essential part of everyday life.

2.1.2 Communication System Architecture

Figure 2.1 shows the basic components of a point-to-point digital communication system. On the transmitter side, the source generates either analog or digital information that could represent anything from text to audio to video. If the information is analog, the source also digitizes it so that it can be represented in bits. The source encoder compresses the digital information by removing any inherent redundancy in order to get the most efficient representation of the information. This compression can either be lossless or lossy, where lossy compression can achieve better compression at the cost of inaccurate reconstruction when decompressed. The channel encoder, or forward error correction (FEC) encoder, adds structured redundancy to the output of the source encoder in order to correct errors that may occur when transmitting the information. The modulator takes groups of bits and converts them into an analog waveform from a predefined set, called a symbol. Examples include pulse-amplitude modulation (PAM), which encodes information in the amplitude of the waveform; phase-shift keying (PSK), which uses the phase to encode information; and quadrature-amplitude modulation (QAM), which is a combination of the previous two methods. This modulated information can either be sent over a single carrier (SC) or over multiple narrowband subcarriers using orthogonal frequency division multiplexing (OFDM). The analog components perform any pulse shaping, predistortion, or upconversion that is required. The analog signal is transmitted across the physical medium, called the communication channel. The model for the channel depends on the physical characteristics of the medium (copper wire, air, fiber optic cable) and the frequency spectrum of the transmitted signal.

On the receiver side, the same process is done in reverse. The analog components may adjust the gain of, filter, digitize, and downconvert the received signals. A demodulator calculates the probability that each symbol was transmitted based on the channel model and the processed received signal. The channel decoder attempts to reconstruct the compressed information sequence using the knowledge of the structured redundancy that was added in by the channel encoder and the probabilities provided by the demodulator. The outputs of the channel decoder may be probabilities or hard decisions on the bits. Finally, the source decoder attempts to decompress the information sequence, which is then converted back into the information's original form at the source. If all went well, the input from the source and output at the sink is (at least approximately) the same. The bit error rate (BER) is a common metric to judge how well the system performs, and it is usually measured after the channel decoder.

Digital communication systems can have many more blocks than those shown in Figure 2.1 depending on the channel and system requirements, such as blocks for synchronization and channel estimation. When the system is part of a network, it also needs a multiple access block that controls access to the shared medium. Examples are time division multiple access (TDMA), which separates different users by scheduling them in different time slots; frequency division multiple access (FDMA), which that separates different users onto non-overlapping frequency bands; orthogonal frequency division multiple access (OFDMA), which separates

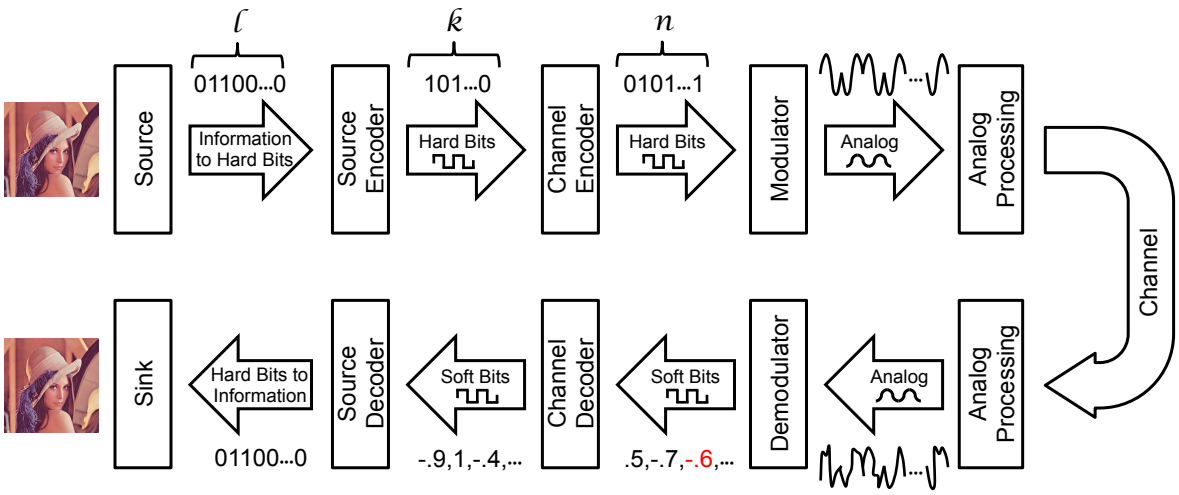


Figure 2.1: A basic digital communication system with examples of the form of information shown after each block with BPSK modulation. Note that $l > k$ and $n > k$.

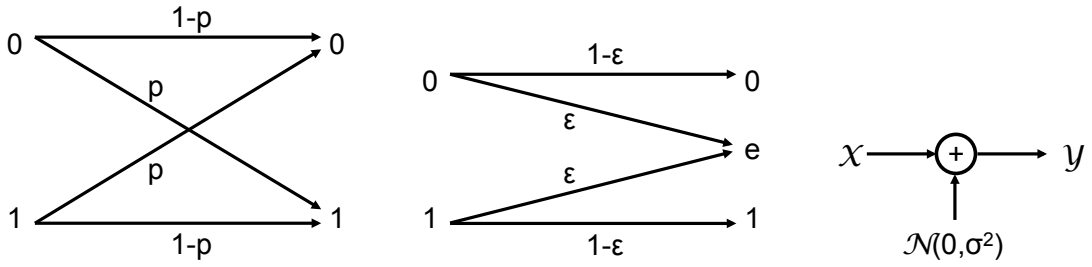
users onto narrowband subcarriers; and code division multiple access (CDMA), which separates users by assigning each uncorrelated pseudo-noise sequences. For a more detailed description of digital communication systems, refer to [36, 37].

2.1.3 Information Theory

Among many other applications, information theory provides a means to analyze the theoretical performance of digital communication systems over a noisy channel. An important aspect of this is the channel capacity C, which is the tightest upper bound on the information rate R that can be reliably communicated over a channel, usually given in units of bits/s/Hz. For $R < C$, there exist codes that allow the probability of error to go to zero. In order to ensure reliable communication, the length of the code goes to infinity. If $R > C$, then the converse states that there are no codes that can make the probability of error go to zero. For more information on capacity and information theory, refer to [38].

2.2 Communication Channel Models

In order to compute capacity and other information theoretic quantities or perform system simulations, the communication channel in Figure 2.1 must be modeled. Simple models are used as a first step in the analysis of a system and to give insight into the problem. More complex models capture the physical processes that modify the transmitted signal when the signal passes through the medium. In either case, most models are stochastic and consist of an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$, and a joint probability function $p(\mathcal{Y} = y|\mathcal{X} = x)$ that is the probability of observing element y given element x was transmitted.



(a) (b) (c)

Figure 2.2: Channel models for the (a) BSC, (b) BEC, and (c) AWGN channel.

2.2.1 Basic Models

The binary symmetric channel (BSC) is a binary-input binary-output model that has probability $1-p$ of having its input and output match and probability p of having its output flip from the input value (Figure 2.2a). The BSC is mainly a toy model and is one the simplest channels to analyze.

The binary erasure channel (BEC) is a binary-input ternary-output model that has zero probability of receiving an incorrect output, but it has probability ϵ of receiving an erased value e that gives no information about the input (Figure 2.2b). This models packets routed through the Internet because they are either received correctly or lost.

The additive white Gaussian noise (AWGN) channel is a discrete-input continuous-output model whose output is equal to the input plus a normal random variable with zero mean and variance σ^2 (Figure 2.2c). An important parameter for this channel is the signal to noise ratio (SNR), which is defined as the signal power P divided by σ^2. For quadrature amplitude modulated (QAM) signals, the inputs are complex numbers, and the noise is modeled as a complex normal random variable rather than a normal random variable. The AWGN channel models thermal noise in wired and wireless media. It can also model channels with interference where the interference is treated as noise by using the signal to interference plus noise ratio (SINR) instead of SNR.

2.2.2 Wired Channels

Wires have thermal noise and act as a low-pass filter. The latter property causes the impulse response to have several taps, which causes intersymbol interference (ISI). The impulse response varies significantly from wire to wire, but each wire's impulse response changes very slowly (on the order of days to months) and can be learned by transmitting a pilot sequence before communicating information. Therefore, in most cases a wire can be modeled as an FIR filter with known coefficients whose outputs go through an AWGN channel (Figure 2.3).

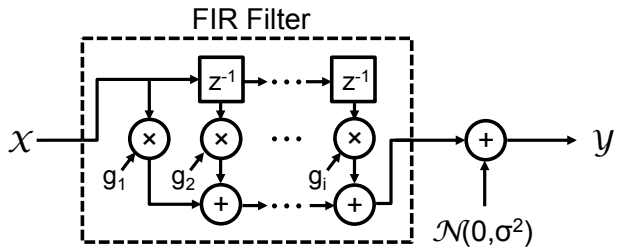


Figure 2.3: Time-domain model of a wired channel that has ISI and AWGN.

2.2.3 Wireless Channels

In a wireless environment, a transmitted signal takes multiple paths to the receiver due to reflections off surrounding objects, and each path carries the same signal but with different delays and phases. Depending on the sampling bandwidth of the receiver, signals from more than one of the paths contribute to a single tap in the impulse response of the channel, and each path can add either constructively or destructively depending on their phase. This effect, called fading, is modeled by a multiplicative random fading term h. If there is no line of sight path between the transmitter and receiver and there are many objects in the environment that scatter the transmitted signal, the magnitude of h is given a Rayleigh distribution. If there is a line of sight path and many scatters, a Rician distribution for the magnitude of h is more reasonable. Other distributions have been proposed, such as the Nakagami-m distribution [39].

If there are paths with significant energy whose delay is longer than those of the paths that contribute to the first tap, then the channel's impulse response has multiple taps. Each tap can have fading, although it may be correlated. The delay spread is the time difference between the first tap and last tap that have significant energy. In the dual view, the channel is frequency selective and in general is not low-pass. The coherence bandwidth is the frequency range over which the channel can be treated as approximately constant and is more formally defined as bandwidth for which the autocovariance of the signal amplitudes at the two extreme frequencies goes from 1.0 to 0.5 [39]. If the signaling bandwidth is smaller than the coherence bandwidth, there is flat fading since the entire channel of interest is constant over frequency (Figure 2.4).

Unlike the wired channel, the wireless channel is not static since the transmitter, receiver, and the surrounding objects that reflect the transmitted signal can all be in motion. This changes the phases of each received signal differently, which causes time variation in the channel. The Doppler shift D_s gives the maximum rate of phase change, which is seen as a frequency shift in the received signal's spectrum. Doppler shift is given by $D_s = vf_c/c$, where v is the relative velocity of the terminal, c is the speed of light, and f_c is the carrier frequency. The time over which the channel stays roughly constant, which is the time it takes for a path's phase to change by half a wavelength, is called the coherence time T_s. A channel with a coherence time longer than the latency requirement is said to have slow

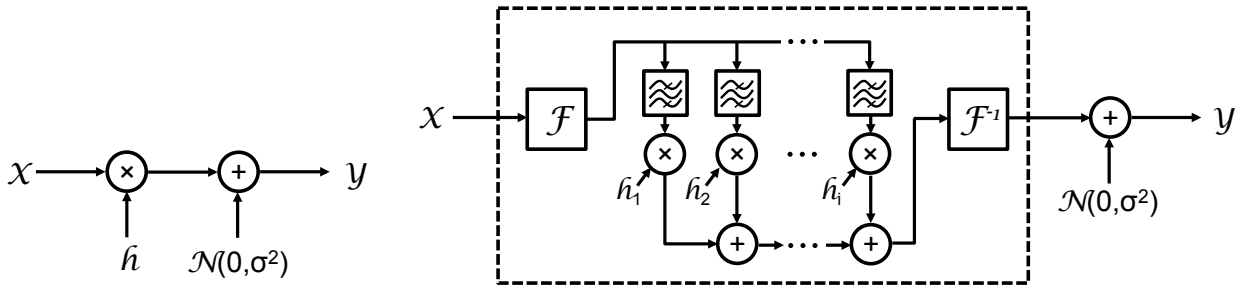


Figure 2.4: Time-domain model of the fading channel for (a) flat fading and (b) frequency selective fading. The $\mathcal{F}$ and $\mathcal{F}^{-\infty}$ blocks Fourier and inverse Fourier transforms. The frequency-domain fading coefficients h_j are time varying, and the rate at which they change depends on the coherence time.

fading; otherwise it has fast fading. Since fading depends on small changes in distance, it is a short term effect.

Fading causes the performance of the system in terms of bit error rate to decline linearly with increasing SNR instead of having the typical waterfall behavior as in the AWGN channel. Diversity improves performance by splitting up the transmitted information over additional degrees of freedom. Time diversity transmits the information in different time slots more than a coherence time apart so that the channel each transmissions sees is independent. Frequency diversity transmits the information across different portions of the spectrum that are more than a coherence bandwidth apart so that each portion again sees a different channel. Antenna diversity arises from having multiple antennas separated by more than a coherence distance at the transmitter (MISO system), receiver (SIMO system), or both (MIMO system), which allows multiple different channels between the transmitter and receiver. Multiuser diversity transmits the information between more users than just the source and destination, each of which have different channels than those between the source and destination. There are many schemes used to implement diversity, which are covered in detail in [39].

Signal power fluctuates over longer time scales when objects obstruct the the propagation paths between the transmitter and receiver, which is called shadowing. This is usually modeled as selected the realization of the SNR from a log-normal distribution with the non-shadowed power as the mean and a standard deviation between 3dB and 7dB, depending on the environment, and then adding in the effects of fading by the methods discussed above [40]. Path loss due to the propagation of the wave through space is the longest time scale effect. The loss is polynomial in the distance and the polynomial's exponent depends on the environment, transmitter and receiver antenna heights, and frequency [39]. Diversity cannot help with shadowing or path loss.

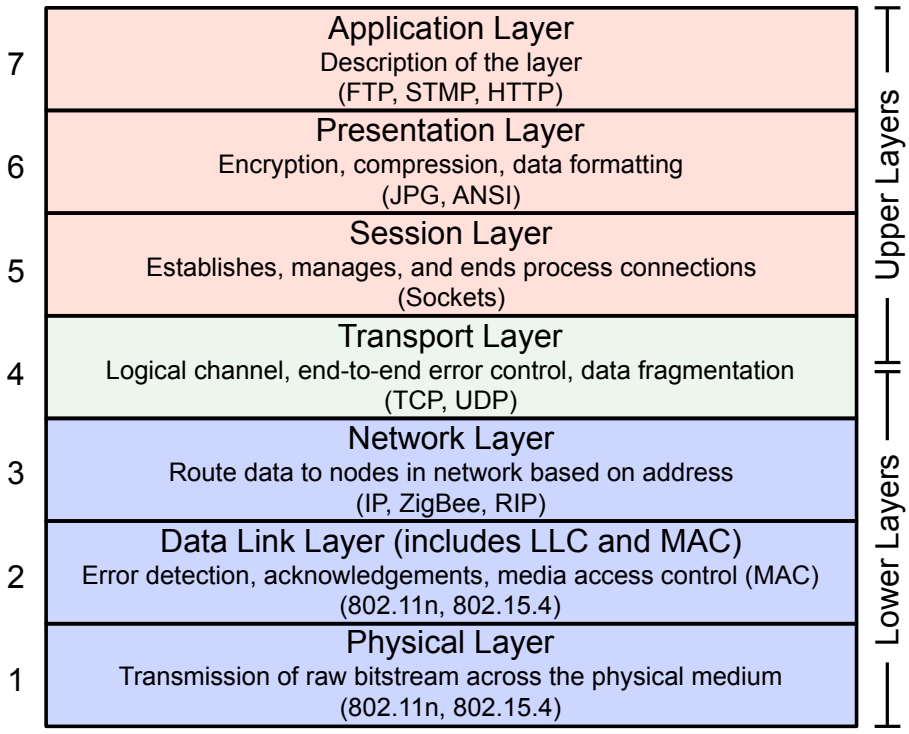


Figure 2.5: OSI layer model with descriptions and examples for each layer. Networks may implement all or a subset of the layers.

2.3 Communication Networks

When multiple devices exchange information over the same medium, they form a network. A variety of network architectures have been standardized for systems based on the communication medium, the distance between and number of nodes, and the required performance. Most systems follow the OSI model, which divides up the functions of a communication network into seven logical layers that only interact with the layers above and below them (Figure 2.5). The upper layers perform application specific functions, while the lower layers perform more basic functions tied to the implementation of the network. Therefore, the lower layers are of more interest to network and communication engineers, particularly the data link layer (DLL), the media access control sub-layer (MAC), and the physical layer (PHY). The following is a description of the lower layers of several representative standards used in current communication networks.

2.3.1 Ethernet (IEEE 802.2 and IEEE 802.3)

Most wired metropolitan and local area networks use Ethernet for their DLL and PHY, which are standardized as IEEE 802.2 and IEEE 802.3 [41, 42], respectively. Modern gigabit

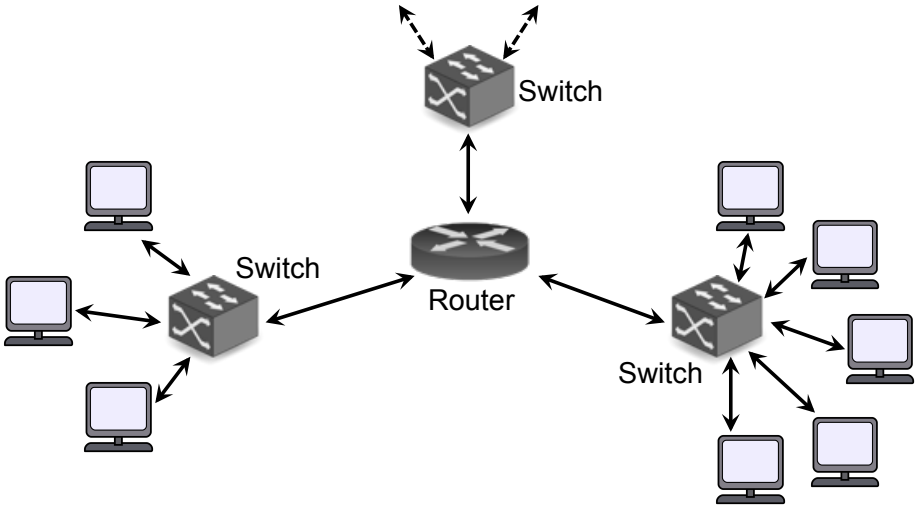


Figure 2.6: Network topology of Ethernet networks.

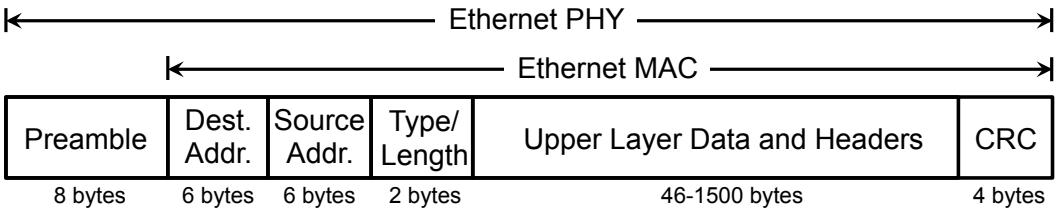


Figure 2.7: Encapsulated packet structure for Ethernet networks.

Ethernet networks have a physical and logical star network topology where the terminals are at the points of the star and a switch is at the center, and switches can be connected to form a tree of star networks (Figure 2.6). Other topologies are possible, but not commonly used. Communication between a terminal and a switch can be either half-duplex, meaning only one can transmit at a time, or full-duplex, meaning each can transmit and receive at the same time.

Data is sent in the form of variable-length packets, where each layer adds its own header and footer fields to the data it receives from the layer above it. For instance, the MAC adds a header containing the unique source and destination addresses and the length of the packet, and it appends a frame check sequence to the end that is used for error detection. The PHY layer may encode the MAC data and then adds a preamble used for synchronization and start of packet detection. Figure 2.7 shows Ethernet's encapsulated packet structure.

MAC

Ethernet has a simple media access policy because the switch has an independent communication link to each terminal and because terminals can only communicate with each other through the switch. In the half-duplex case, a terminal and switch use Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to determine which has access to the medium. CSMA/CD requires that a node senses the medium before it transmits to make sure it is not in use. If the medium is in use, the node waits until it is clear plus a fixed extra time. When the node transmits, it also listens to the medium to determine if another node is transmitting at the same time, which is called a collision. If a collision occurs, the node stops transmitting and sends a jam sequence to inform other nodes that a collision has occurred. Afterwards, it calculates a random backoff time that it must wait before sensing the medium again and retransmitting. In the full-duplex case, no collisions can occur, so no multiple access protocol is required.

The switch arbitrates access to a particular link when two terminals want to transmit to the same location at the same time by sending one packet at a time and storing the rest. The switch usually transmits stored the packets in the order in which they are received, but other policies are possible. Switches can operate in three different modes: (1) cut-through, (2) store-and-forward, and (3) fragment-free. Cut-through switching begins forwarding a packet as soon as it is received and the destination address is read. Store-and-forward switching waits until the entire packet is received and checks the packet for any errors before forwarding. Fragment-free switching waits until the entire 64 bytes of the packet's header is received, checks it for errors, and forwards it if there are no errors. Cut-through switching results in the lowest latency network, store-and-forward switching makes better use of the network's resources, and fragment-free switching is a trade-off between the other two methods [43].

Ethernet uses cyclic redundancy check (CRC) codes to detect errors in the packet and the automatic repeat request (ARQ) policy to correct errors. CRC codes are defined by a generator polynomial know to both the transmitter and receiver that divides the polynomial representation packet's data. The remainder of this division is appended to the packet. When the packet is received, the remainder is recalculated and compared to the remainder appended to the packet. If they do not match, an error has occurred. The probability of an error occurring and not being detected decreases as 2^{-B}, where B is length of the remainder appended to the packet [44]. To correct errors ARQ requires that all transmissions must be acknowledged with either an ACK if the packet is received successfully or a NAK if it is not received successfully (based on the CRC check). Depending on the type of switching, either an intermediate switch or the final terminal sends the acknowledgment. The transmitting terminal waits for the acknowledgment, and sends a new packet if it receives and ACK or resends the same packet if it receives a NAK, which is called stop-and-wait ARQ. After a specified maximum number of attempts, the transmitter will give up and discard the packet.

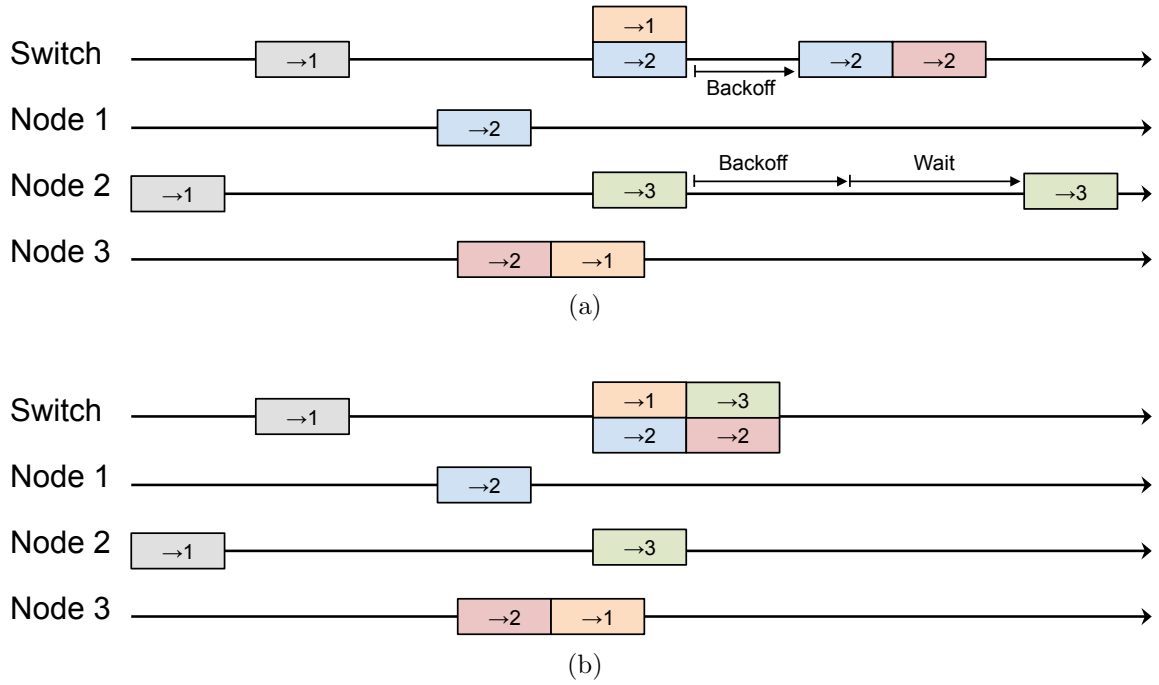


Figure 2.8: Timing diagram for an Ethernet transmission for (a) a half-duplex system and (b) a full-duplex system.

PHY

Ethernet physically transmits data over copper twisted pairs, copper coaxial or twinaxial cables, or fiber optic cables. There are many clauses in the IEEE 802.3 standard that define data rates and ranges using each of these media. Twisted pairs can achieve a maximum data rates of 10Gb/s at a range of 100m, twinaxial cable have a maximum data rate of 100Gb/s at a range of 7m, and fiber optics can reach 100Gb/s at a range of 40km [42]. Figure 2.9 shows the PHY layer for the transmitter and receiver specified by IEEE 802.3an, which is used for 10 Gigabit Ethernet over copper twisted pairs. First, the data passes through a line encoder and a scrambler, which modify the data to ensure there are enough transitions (from $0 \rightarrow 1$ or $1 \rightarrow 0$) for clock and data recovery circuits to operate correctly. In addition, the scrambler gives the transmitted signal a random power spectrum to reduce electromagnetic interference and to aid equalizer convergence. Next, an error-control code is used to protect the data from errors. The data is then modulated and pre-emphasized. Since all signaling is baseband, real modulations, such as pulse-amplitude modulation (PAM) must be used. Pre-emphasis attempts to pre-distort the transmitted data so that it inverts the channel's effects. Finally, the data is passed to the analog front end for transmission across the Ethernet cable. On the receive side, the receiver undoes the transmitter's signal processing, but it first cancels the crosstalk caused by its own transmitter using the known transmitted data. Other standards may use different implements of each block, but the order and function of the blocks is

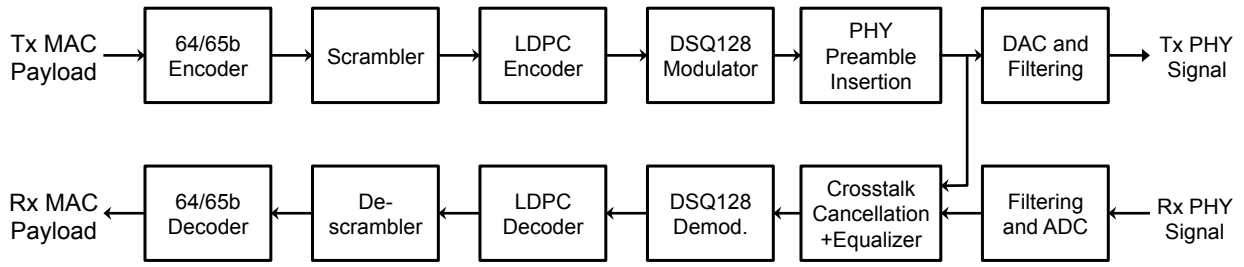


Figure 2.9: The PHY layer of the 10GBASE-T Ethernet (IEEE 802.3an) standard, with the transmitter on the top and receiver on the bottom. The transmitter uses a 64b/65b line code and 58-tap scrambler. The error control coding consists of a single LDPC code with rate $R = 0.84$. DSQ-128 modulation is used, which is based on PAM with 16 levels. Finally, the data is pre-emphasized using Tomlinson-Harashima precoding. The receiver has blocks that reverse the transmit signal processing, but also includes a block to eliminate crosstalk from its own transmit chain.

similar. In Ethernet, no forms of diversity are required because the wired channel does not experience fading.

Performance

Ethernet typically achieves latencies on the order of microseconds for a local point-to-point link. For reliability, it usually operates at a BER of 10^{-10} to 10^{-12} after the PHY layer, which translates to a BLER of approximately 10^{-8} to 10^{-10} depending on packet size. After the MAC layer, the error rate is reduced due to the ARQ protocol.

2.3.2 SERCOSIII

SERCOSIII targets low-latency, high-reliability industrial control and automation applications. It is built on top of the PHY and MAC layers of 100Mb/s Ethernet (IEEE 802.3u). Unlike typical Ethernet networks, SERCOSIII uses either a ring or a line topology with full-duplex links, and each node has the functionality of a network switch because it reads or modifies a packet in place and then forwards it. The ring topology acts as two redundant lines because the same information is sent in both directions along the ring, and the destination checks that both copies of the information are the same. In the case of a node failure when using a ring topology, the break is detected within a cycle time and the network automatically becomes a line topology. This behavior enables hot-swapping nodes since the system will change to a line while a node is taken out, and change back to a ring after the ring is reestablished. Figure 2.10 shows the line and ring topologies. Multiple rings can be connected together to form larger networks, but the underlying functionality is still circular message passing.

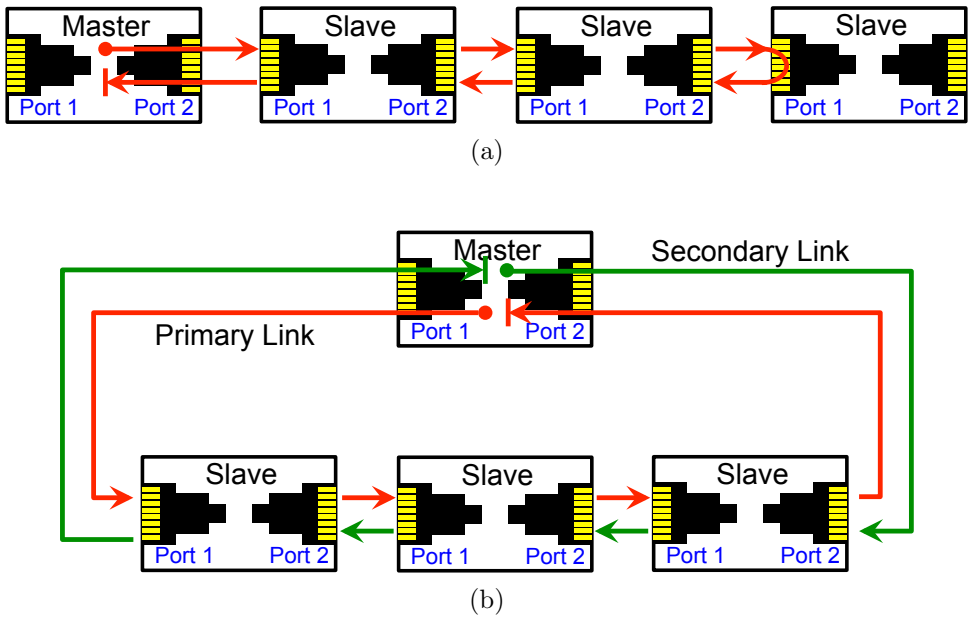


Figure 2.10: Network topologies for SERCOSIII: (a) line and (b) ring.

Since SERCOSIII is used for control applications, nodes can either act as a master or a slave. The data transmission pattern is periodic with period T_{SERCOS}. A master node initiates each of the two types of data transmissions, which are called master data telegrams (MDT) and acknowledgment telegrams (AT). Both MDTs and ATs have the same packet structure as an Ethernet frame. MDTs contain control information from the master for each of the slave nodes. Slaves read their portion of the packet and then forward it without modification. ATs are data requests from the master to the slaves that start out as a blank Ethernet packet with only a preamble and header. Each slave adds its own data into the appropriate position in the packet and forwards the AT, which reduces the overhead by sharing the preamble and header. Up to four MDTs and four ATs can be sent every T_{SERCOS}, and any left over time can be used to send any other form of Ethernet traffic (Figure 2.11). SERCOSIII supports direct slave to slave transmissions by having the slaves put information into a telegram intended for another slave node (the master will ignore this information). Direct master to master communication is also possible.

Only the first MDT contains synchronization information, which further reduces the SERCOSIII's overhead. During initialization and whenever a node is added to the network, the master measures the ring time and transmits that to all slaves. The MSTs, which are sent at equal intervals once every cycle time, are used by slaves as incremental updates for their timing in case their clocks drift from the master's. Using this method, SERCOSIII achieves a synchronization accuracy of 20ns and simultaneity of 100ns [45].

Table 2.1 gives typical performance numbers for a range of applications using SERCOSIII in a single ring topology. SERCOSIII can achieve an extremely low cycle time of 31.25μs for

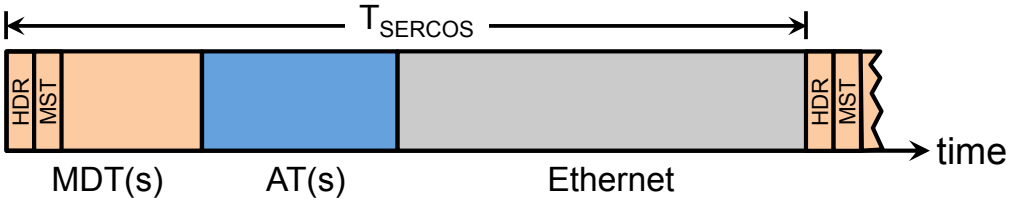


Figure 2.11: SERCOSIII cycle timing diagram.

Bytes/ Node	Cycle Time (μs)	Number of Nodes
8	31.25	8
12	250	70
32	1000	150
16	1000	254

Table 2.1: Performance examples for the SERCOSIII protocol [46].

8 nodes and transmitting 8 bytes of data each and can scale up gracefully to a cycle time of 1ms for 254 nodes transmitting 16 bytes of data each. The protocol maintains a high level of reliability since it uses Ethernet and uses a ring topology for redundancy.

2.3.3 DSL and Cable (ITU G.993.2 and DOCSIS 3.0)

Internet service providers deployed digital subscriber line (DSL) and cable networks to provide high-bandwidth connections between their customers and their point of presence (PoP) in the region through intermediate stations called DSLAMs (for DSL) or CMTSs (for cable). DSL operates over existing copper phone lines that are each dedicated to a single user, while cable operates over coaxial lines used for cable television that are shared among many users in the same area. For these reasons, DSL uses a physical and logical star network topology, while cable uses a physical bus and logical star network topology. Both have full-duplex communication.

DSL

MAC DSL can use either Ethernet's protocol or the Asynchronous Transfer Mode (ATM) protocol as its media access policy, although the latter is more common. ATM creates virtual circuits between a source and destination on a packet-switched network, which enables guarantees on the quality of service because the virtual circuits are essentially dedicated links. Information is transmitted in the form of fixed-length packets, where each packet has a 5 byte header and a 48 byte payload, and there is no delay between packets. Since the network has to set up a virtual circuit before transmitting, the initial delay may be up to

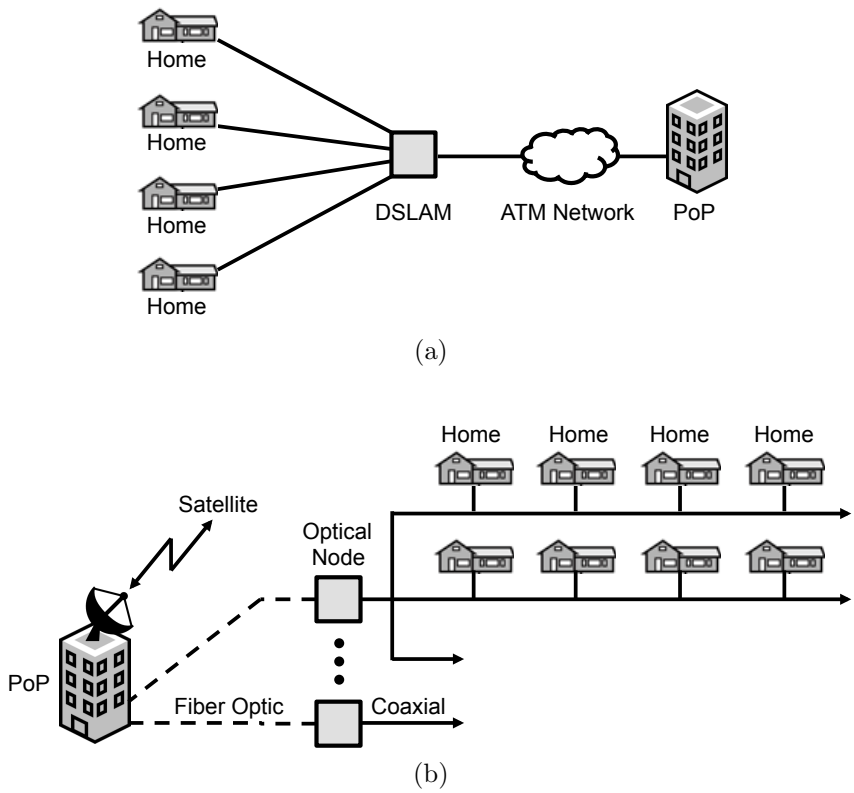


Figure 2.12: Network topologies for (a) DSL and (b) cable networks.

30 seconds, but afterwards the transmission has lower latency than Ethernet because no additional processing is required for routing.

PHY DSL transmits data over copper phone lines between a user and the PoP via a DSLAM (the local hub near a group of users). Legacy phone services transmit voice as a baseband analog waveform with a bandwidth of 3.4kHz. In order to be compatible, DSL transmits data at frequencies between 25kHz and up to 30MHz, depending on the version and region (Figure 2.13). Frequency division duplexing is used to enable full-duplex communication, and the exact allocation depends on the provider and region. The aggregate data rate depends on the distance between the user and the DSLAM, with a maximum of 200Mb/s when near the DSLAM down to 4Mb/s when 4-5km from the DSLAM.

Figure 2.14 shows the PHY layer for a DSL transceiver. The forward error correction uses a concatenation of a Reed-Solomon code and a convolutional code. The bits are interleaved between the two codes in order to break up burst errors that can occur during transmission due to crosstalk. The coded bits are mapped to symbols from a QAM constellation with a number of bits between 1 and 15 (BPSK to 32768-QAM). The symbols are modulated using orthogonal frequency division multiplexing (OFDM) by taking the inverse fast Fourier transform (IFFT) of a block of symbols and inserting a cyclic prefix. The IFFT size is

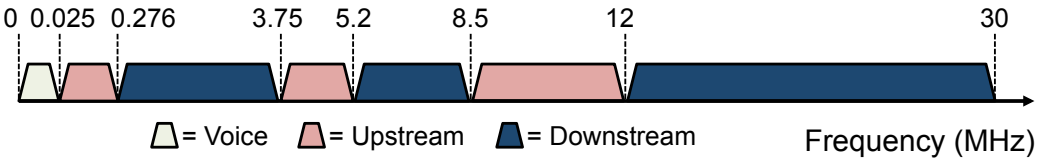


Figure 2.13: Frequency plan for DSL (ITU G.993.2)

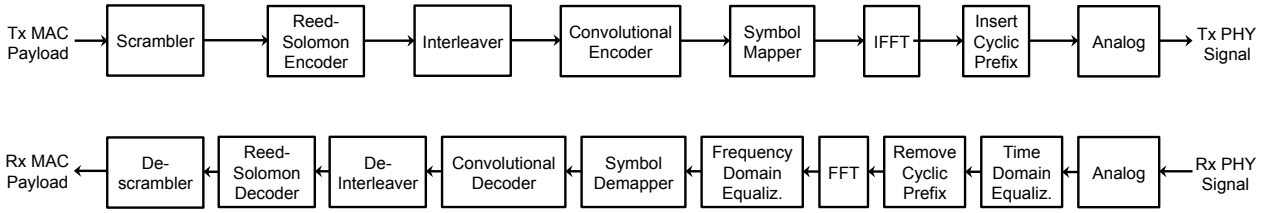


Figure 2.14: Physical layer for a DSL transceiver.

between 64 and 8192, depending on the bandwidth used, which is double the number of subcarriers used since the transmitted signal is real. The subcarrier spacing is 4.3125kHz if the bandwidth is below 30MHz; otherwise, it is 8.625kHz. Note that each subcarrier can carry a symbol from a different constellation, which is fixed during initial training of the link. The receiver performs the reverse processing with the exception of having time and frequency domain equalizers to undo the effects of the channel, including crosstalk. Some DSL transceivers include an echo cancellation block that cancels crosstalk at the receiver using knowledge of the transmitter's data.

Performance DSL has two processing chains: one for low latency and another for larger burst error protection through interleaving. Ignoring retransmissions, the low latency chain achieves a latency smaller than 1ms to the DSLAM, and the high latency chain has a latency up to 20ms to the DSLAM. DSL typically operates at a BER of at least 10^{-7}, or approximately a BLER of 10^{-5}, after the PHY layer. ARQ corrects most of these errors before the data reaches the user.

Cable

MAC Cable uses different media access policies on its downlink (from CMTS to users) and uplink (from users to CMTS). The downlink broadcasts all packets, and only the targeted users retain the packet. The uplink uses either time-division multiple access (TDMA) or synchronous code division multiple access (S-CDMA) to keep users' transmissions from colliding. TDMA orthogonalizes users by assigning each user time slots during which only they may transmit. S-CDMA also assigns users time slots that they may transmit during, but many users share the same time slot. However, each user is assigned a unique binary

spreading code that orthogonalizes their transmission with respect other users transmitting at the same time. This takes more time to transmit, but it is more immune to noise. In either the TDMA or S-CDMA case, the CMTS controls resource allocation and synchronization. The CMTS schedules downlink traffic according to each users' buffer status and their subscription level, but users must send scheduling requests during a MAP slot to get resources. The MAP slots typically occur every 2ms, and resources are granted within 4-8ms if the network is not congested.

PHY Cable transmits data over coaxial cables at RF frequencies up to a distance of 160km. The downstream channels are 6MHz wide and occupy frequencies between 108MHz and 1.002GHz, and the upstream channels are also 6MHz wide and occupy frequencies between 5MHz and 42MHz. The downstream uses either 64-QAM or 256-QAM symbol mapping with concatenated Reed-Solomon (RS) coding and Trellis Coded Modulation (TCM). The RS code is over GF(128) an can correct up to 3 symbol errors. Bits are scrambled between the two stages of coding, and a PHY preamble is prepended after TCM encoding. The upstream uses QPSK, 8-QAM, 16-QAM, 32-QAM, or 64-QAM symbol mapping in TDMA and S-CDMA mode and can additionally use 128-QAM in S-CDMA mode. In TDMA mode, only RS coding is used, but S-CDMA mode can use concatenated RS-TCM coding. The RS code is over GF(256) and can be configured to correct between 0 and 16 errors. The Reed-Solomon code is over GF(256) and can correct between 1 and 16 errors depending on the configuration. Bits are scrambled after the RS coding, and the PHY preamble is attached after all encoding. The maximum throughput over a single channel is 42.88Mb/s (38Mb/s not including overhead) on the downstream and 30.72Mb/s (27Mb/s not including overhead) on the upstream. Channel bonding enables a single user to communicate over multiple channels simultaneously in the downstream or upstream, and the bonding of m channels increases the throughput by a factor of m.

Performance The average latency of the UL (ignoring retransmissions) is in the range of 4-8ms in an uncongested network. On the DL, the minimum latency varies based on the amount of codeword interleaving used to protect against noise bursts and the modulation, and can be between 0.15ms to 4ms. For the UL, a typical BER after the PHY is 10^{-6}, which corresponds to a BLER of 10^{-4}. For the DL, a typical BER after the PHY is 10^{-8}, which corresponds to a BLER of 10^{-6}. An ARQ protocol is used in the MAC and transport layers, so the final error rate for the user is lower.

2.3.4 2.4GHz/5GHz Wireless LAN (IEEE 802.11ac)

In the late 1990's and early 2000's, most residential and public-space networks began transitioning from wired to wireless connections because the latter is easier to install, can support many users with minimal equipment, and allows mobility. The IEEE 802.11 standard is the almost universal choice for the physical and data link layers of these wireless local area

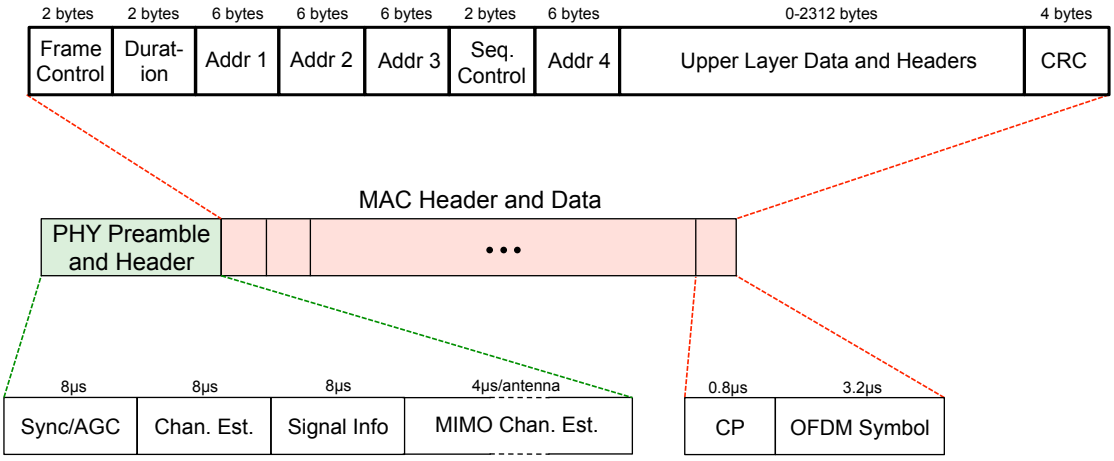


Figure 2.15: Encapsulated packet structure for IEEE 802.11ac networks.

networks (WLAN). IEEE 802.11 has been amended several times since its creation in 1997, where each amendment adds features to increase data rate while maintaining a measure of backward compatibility with previous versions. The current amendments are "ac" and "ad". This section focuses on amendment "ac", and the next focuses on amendment "ad".

IEEE 802.11ac uses a logical star network topology where all terminals connect to an access point (AP), and the AP may either connect to another AP or to a wired network. Physically, all nodes broadcast over the same wireless medium, which means neighboring wireless networks using the same frequency channels will interfere with the operation of network. All communication between each terminal and the AP is half-duplex. Similar to Ethernet, all data is sent in the form of variable-length packets, and each layer encapsulates the data from the layer above it (Figure 2.15). The MAC adds a header containing control information, the unique addresses of up to four terminals involved in transmission of the packet, and a packet identifier, and it appends a frame check sequence used for error detection. The PHY adds a preamble that is specific to the amendment of IEEE 802.11 being used, but usually contains a long and short training sequence used for synchronization, automatic gain control tuning, and channel estimation. It also has information on the modulation, coding, and length of the data portion of the packet.

MAC

IEEE 802.11ac divides up time into superframes that start with a beacon transmitted by the AP. The beacon contains information terminals require to join the network. The rest of the superframe is divided between the two media access policies defined in IEEE 802.11: contention and contention-free (Figure 2.16). Regardless of the policy, any errors that occur during transmission are detected using CRC codes and are corrected using the ARQ protocol.

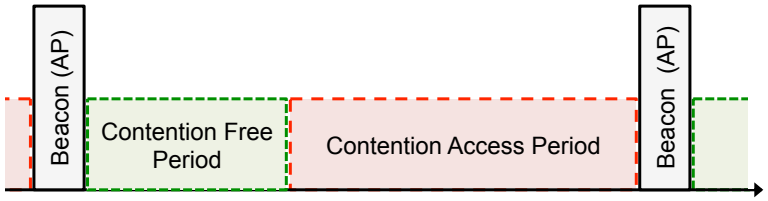


Figure 2.16: Superframe structure of IEEE 802.11.

Contention-based access uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol, illustrated in Figure 2.17 when CSMA/CA is combined with ARQ. To begin, a node that wants to transmit senses the medium to see if it is clear. In the case of nodes 2 and 3, the medium is not clear when they initially want to send data, so they must wait and periodically sense the medium to check when it becomes free. After node 1 gives up access to the medium, both nodes 2 and 3 sense that the medium is idle, but they cannot begin transmitting yet. First, they wait a distributed-coordination-function interframe spacing (DIFS) time of 32μs to make sure that medium is idle and to allow higher priority traffic to gain medium access first. After, if the medium is still idle, they calculate a random backoff time between zero and 15μs in order to avoid their transmission colliding. In this case node 3 has a smaller backoff time, so it gains control of the medium and transmits its data. Node 2 pauses its backoff time, and waits until the medium is idle. When node 3 finishes transmitting its data, the AP has an short interframe spacing (SIFS) time of 16μs to receive and process the incoming data, and then it has to send an ACK or NAK packet, which has a minimum length of 32μs. Since node 3 receives a NAK (or if it had not received any ACK or NAK), it must contend for the medium again to retransmit its data, and it must double the upper limit of its random backoff time. If it received an ACK, it would either contend for the medium to transmit new data or wait until it had more data to send. If the AP wants to send data, it follows the same protocol. To enhance throughput, block acknowledgments are used, which allow a terminal to send multiple data frames before the receiver responds with an acknowledgment for all the data. In addition, frame aggregation is used, which allows a terminal to aggregate information from higher layers of the network into one MAC frame in order to reduce overhead and the number of channel accesses.

Contention-free access is based on the AP polling each terminal connected to the network to check if it has data to send, illustrated in Figure 2.18. The poll from the beacon may contain an ACK for the previous terminal and data for the current terminal. Each poll and response must be followed by at least an SIFS time, and a point-coordination-function interframe spacing (PIFS) time that is longer than the SIFS time must elapse in the case a terminal does not respond with data.

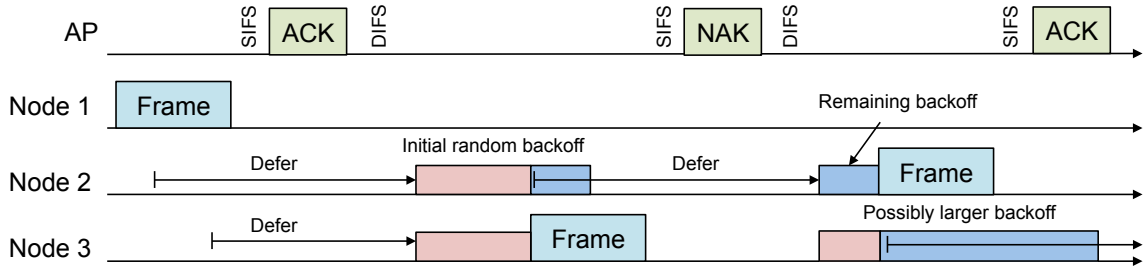


Figure 2.17: Contention-based media access policy based on CSMA/CA for IEEE 802.11ac.

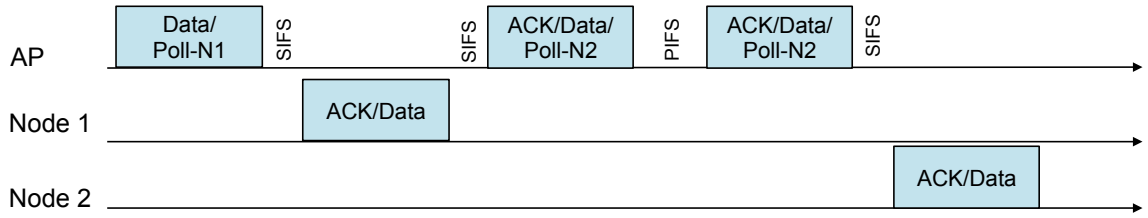


Figure 2.18: Contention-free media access policy for IEEE 802.11ac.

PHY

IEEE 802.11ac operates in the ISM band located between 5.17GHz and 5.925GHz (previous amendments also used the ISM band around 2.4GHz, but "ac" does not). The bands are broken up into 20MHz wide, non-overlapping channels that can be bonded together to form channels that are 40MHz, 80MHz, or 160MHz wide, including bonding two non-contiguous 80MHz channels (Figure 2.19). At short ranges and in an ideal environment, IEEE 802.11ac can achieve data rates up to 433Mb/s using a SISO link up to 6.93Gb/s using 8x8 MIMO, but this will decline with distance and environmental conditions.

Figure 2.20 shows the transmit and receive chain for IEEE 802.11ac devices for all channel widths except 160MHz and for a single user. The MAC payload is padded so that its length is compatible with the current configuration of the PHY. Next, the bits are scrambled and then fed to a FEC encoder. The FEC codes can be either binary convolutional codes (BCC) or a low-density parity-check (LDPC) codes. Since the wireless channel changes quickly in time compared to wired channels, different code rates are specified to adapt to different channel conditions. A single BCC code of rate 1/2 is specified that can be punctured to create codes of rates 2/3, 3/4, and 5/6. The LDPC codes specified have blocklengths of 648, 1296 and 1944, and each blocklength has a rate 1/2, 2/3, 3/4, and 5/6 code defined for it. If a BCC code is used, the bits are separated into disjoint subsets and encoded separately

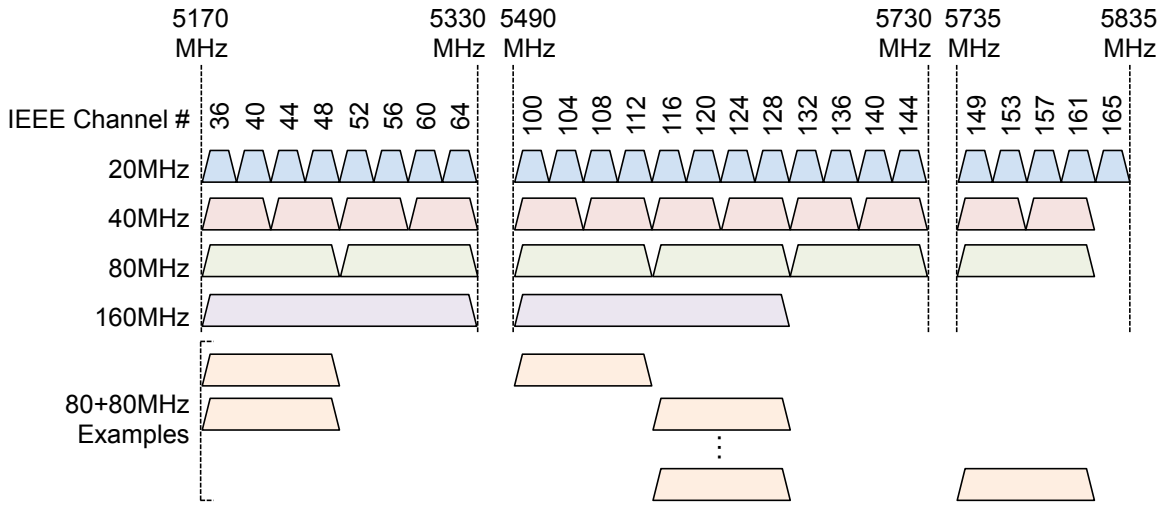


Figure 2.19: Frequency plan for IEEE 802.11ac in the United States.

using the same BCC code. The number of subsets depends on the modulation and coding scheme (MCS) currently used. If an LDPC code is used, only a single encoder is used.

Next, the output(s) of the FEC encoder are mapped to the specified number of spatial streams, which are independent data streams that will be sent across the MIMO link simultaneously. The number of spatial streams, N_{SS}, can range from 1 to 8, where the maximum is set by the minimum number of antennas at the transmitter and receiver. If BCCs are used, the bits are interleaved so that if a burst error occurs, the errors will be spread out over the entire codeword. The bits are mapped to either a BPSK, QPSK, 16-QAM, 64-QAM, or 256-QAM constellation. If LDPC codes are used, the constellation points are mapped to OFDM subcarriers separated by a specified distance to reduce the probability of successive bits are corrupted by frequency-selective effects.

After, a space-time block code is optionally applied to transmit one spatial stream across multiple antennas for additional redundancy. This maps one spatial stream onto multiple transmit chains, forming space-time streams. Each space-time stream has pilot symbols inserted to aid in channel estimation and other receiver training and has cyclic shifts applied to aid in detection. The space-time streams are mapped onto the transmit chains using the spatial mapper. A small number of pilot symbols are added to each transmit chain's stream in fixed locations to track channel changes and frequency drift.

Finally, OFDM modulation occurs for each transmit chain, which consists the IFFT and guard interval insertion. The OFDM symbols are 3.2μs long with a guard interval/cyclic prefix of 400ns or 800ns. The IFFT size used is 64, 128, 256, or 512 for a 20MHz, 40MHz, 80MHz, and 160MHz channel, respectively. This gives each configuration a subcarrier spacing of 312.5kHz. Due to pilot symbols and nulling subcarriers around DC, the number of subcarriers that carry information is 52, 108, 234, and 468 for the 20MHz, 40MHz, 80MHz,

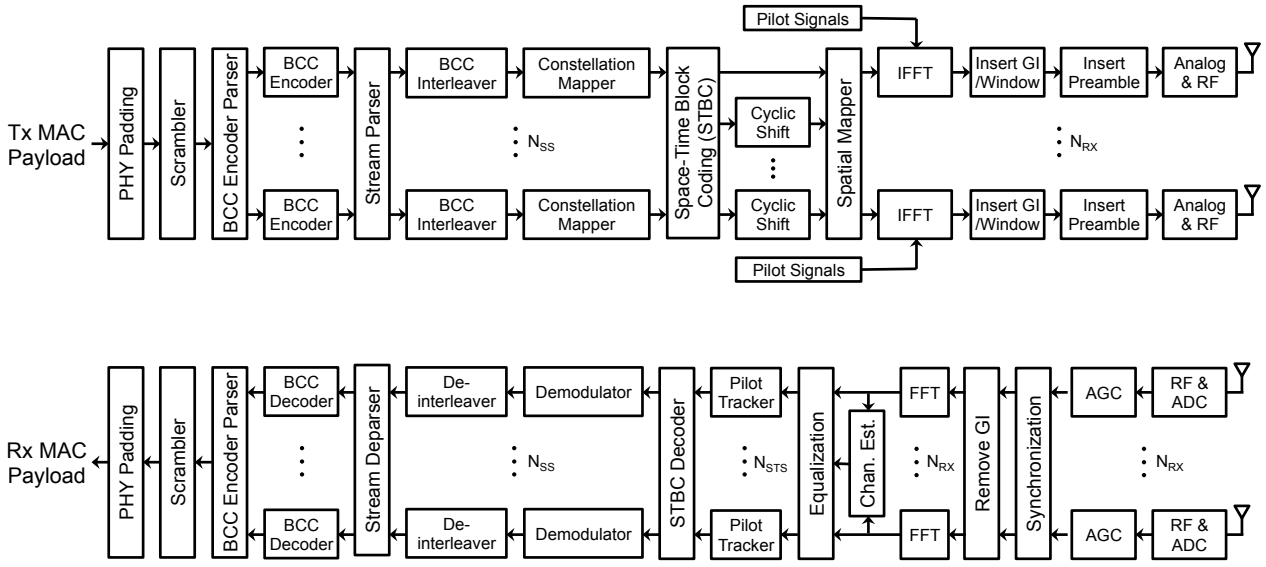


Figure 2.20: An IEEE 802.11ac transceiver.

and 160MHz channel widths, respectively. The PHY preamble is inserted, which can be between 28μs and 40μs, and the digital data is sent to the analog and RF components for transmission.

In the case of using a 160MHz channel, the input bits are separated into two segments after FEC encoding that undergo the same interleaving and constellation mapping. Afterwards, the streams from the two segments are recombined. In IEEE 802.11ac, it is also possible to transmit to up to 4 different users using a technique called multi-user MIMO. In this case, each user's data is processed separately up to the spatial mapping stage. In the spatial mapping stage, each users' streams are beamformed for that user, and the OFDM modulation occurs as with a single user.

The receiver uses an automatic gain control loop (AGC) to maximize the dynamic range of the receiver, a burst detector that detects the beginning of a frame, and a timing/frequency recovery block to obtain a sampling time lock and carrier frequency lock. It then removes the guard interval and takes the FFT of the data, using the preamble and pilot tones in the data segment to estimate the channel. It uses the channel estimates to equalize the data in the frequency domain, and then deparses and decodes the data.

Performance

There are no full IEEE 802.11ac deployments yet, but the latency should be better than that of IEEE 802.11ac's predecessor IEEE 802.11n because IEEE 802.11ac has essentially the same MAC layer but much higher data rates arising from an upgraded PHY layer. IEEE 802.11n's minimum latency is on the order of 10ms; a large portion of this latency

comes from interframe spacing times and random backoff times. Since IEEE 802.11ac shares these sources of MAC overhead, its latency will still be on the order of several milliseconds. IEEE 802.11n typically operates at a BER of 10^{-4} to 10^{-6}, and IEEE 802.11ac will have a similar reliability since it has the same error correction mechanisms and coding/modulation adaptation.

2.3.5 60GHz Wireless LAN (IEEE 802.11ad)

Due to the limited bandwidth and large number of devices using the 2.4GHz/5GHz ISM bands, interest turned to using large portions of the unregulated spectrum around 60GHz. Instead of the maximum bandwidth of 160MHz in IEEE 802.11ac, bandwidths on the order of a gigahertz can be used by each device.

Compared to ISM channels, 60GHz channels have much larger propagation losses. However, 60GHz carrier frequencies enable antennas to be placed much closer together than 2.4GHz or 5GHz carrier frequencies. In fact, multiple antennas can be placed on a single chip that can be used for beamforming to increase the received power. IEEE 802.11ad extends the IEEE 802.11 standard to operate in the 60GHz band, so the network topology and packet structure is similar to that in IEEE 802.11ac. One difference is that the ad-hoc mode may be more common where one of the devices is elected to act as AP, user to user communication is allowed, and the APs of different networks can communicate with one another to schedule resources efficiently. Also, the standard requires each device to have a set of predefined directions, called sectors, along which it can focus its transmitter or receiver.

MAC

IEEE 802.11ad adapts the IEEE 802.11ac MAC for devices that have highly directional transmissions, primarily by modifying the superframe structure and adding a beamforming training protocol. The superframe is broken up in to four periods: (1) beacon, (2) adaptive beamforming training (A-BFT), (3) announcement time (AT), and (4) data transmission interval (DTI). During the beacon period, the AP transmits a single beacon packet containing network parameters in each of its sectors sequentially, which additionally serves as the beginning of beamforming training from the AP to users. In the A-BFT, each user contends for the medium in order to learn the optimal sector for transmitting to the AP and to finish the AP to user training. During the AT, the AP transmits scheduling information, control information, and beam refinement packets for further beamforming training. The DTI is used for scheduled and/or contention-based access (based on a directional version of CSMA/CA) to the channel for data exchange between either the AP and users or pairs of users. Also, It can be used for additional beamforming training between users and the AP in case there was not enough time in the first three periods or between pairs of users. The same CRC and ARQ techniques as used in IEEE 802.11ac are used to increase reliability.

If there are multiple networks close together, the APs of each network can coordinate and schedule transmissions at the same time as long as they do not spatially collide. Likewise,

during contention-based access, simultaneous transmissions do not necessarily collide and separate backoff timers are required for each transmission direction. This enables better frequency reuse in dense network environments.

The beamforming training protocol is a low overhead method to find the optimal sectors for transmission and reception between a user and AP or a pair of users. Beamforming training is done in up to three phases, as shown in Figure 2.21b for an AP-user link. First, the AP performs a sector level sweep by transmitting a packet over each of its sectors while the user maintains an omnidirectional receive pattern. Next, the targeted user responds with its own sector level sweep and the transmission number from the AP that resulted in the highest power while the AP maintains an omnidirectional receive pattern. The AP responds with the transmission number from the user that results in the highest power, and the user responds with an ACK packet to conclude the sector level sweep. The same training can be performed for a receiver. Beam refinement is based on a request-response protocol where fields appended to the packet are used to fine tune beamforming parameters. For a user-user link, either a user must gain access to the channel during a contention based access period and begin a sector level sweep or a user requests a scheduled period from the AP. In either case, the basic protocol is the same as for AP-user links.

PHY

IEEE 802.11ad operates in the 60GHz band located between 57.24GHz and 65.88GHz and defines four 2.16GHz wide channels (Figure 2.22). It has an operating range on the order of 10m or less in line-of-sight channels, but it can also be used at shorter distances in non-line-of-sight channels. To increase power gain, a large array of antennas are used to beamform the signal, so only a single spatial stream is supported (no MIMO).

The standard defines four PHY modes: (1) control, (2) single-carrier (SC), (3) low-power single-carrier (LP-SC), and (4) OFDM. Control is used for carrying control channel messages and is the most reliable, lowest-throughput mode. It uses a rate 1/2 LDPC code with 32x spreading and BPSK modulation, which achieves a throughput of 27.5Mb/s.

SC is the primary PHY mode for IEEE 802.11ad due to its combination of low power consumption and high throughput. It supports data rates from 385Mb/s to 4.62Gb/s, depending on the MCS. Figure 2.23 shows the transceiver architecture for SC mode. The MAC payload is scrambled and then encoded with an LDPC code with blocklength 672 and rate 1/2, 3/4, 5/8, or 13/16. The bits are mapped to symbols from either a $\pi/2$-rotated BPSK, QPSK, or 16-QAM constellation. The data is then segmented into blocks of 512 symbols, where the first 64 are guard symbols from a Golay sequence and the last 448 are data symbols. The preamble is sent first using $\pi/2$-BPSK symbols, and then the data blocks are sent. The preamble consists of a training sequence, a channel estimation sequence, and signal information fields. The preamble takes 2.473μs to transmit, and each data block takes 0.291μs to transmit. On the receive side, frequency domain equalization is used, so the received signal goes through a 512-point FFT, is equalized, and goes through a 512-point IFFT. Since only one spatial stream is used, the rest of the signal processing is straightforward.

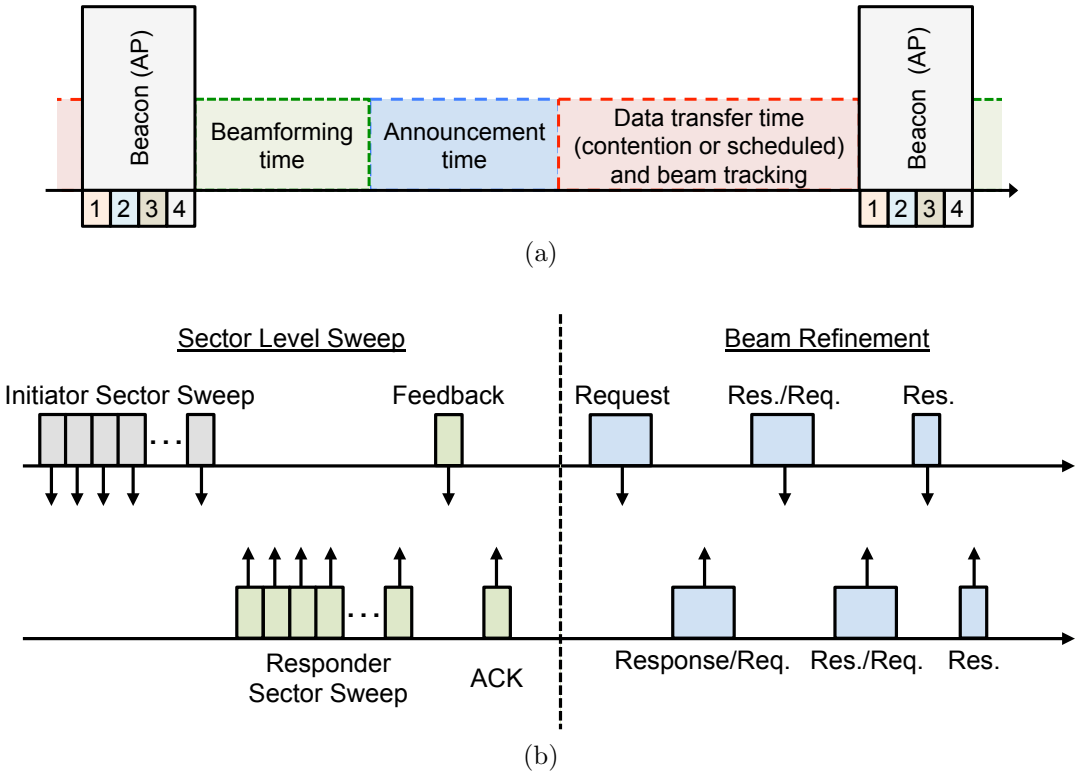


Figure 2.21: (a) Superframe structure for personal access points that adds in beamforming and announcement times to the IEEE 802.11ac superframe structure. The beacon is transmitted in each configured direction. The data transmission period can be used for scheduled or contention-based access. (b) Beamforming protocol that occurs during the beacon (gray), beamforming (green), and announcement (blue) periods of the superframe consists of a sector sweep by the access point, a sector sweep by users, and a request/acknowledgment refinement procedure. All of these can also occur during the data transmission period.

LP-SC mode has several features in common with the SC mode, but there are a few key differences. Instead of LDPC codes, LP-SC uses a concatenation of an RS(224,208) and Hamming(n,6) codes, where n is 8, 9, 12, or 16. Only $\pi/2$-BPSK and QPSK constellations are used. Finally, the 512 symbol blocks consist of the initial 64 guard symbols and 7 sub-blocks consisting of 56 data symbols followed by 8 guard symbols. The achievable data rates range from 625.6Mb/s to 2.503Gb/s.

OFDM mode is less commonly used because it has higher power consumption due to a higher peak to average power ratio than single-carrier modulation, which requires additional power amplifier backoff. This mode has a typical OFDM transmit and receive chain for a single stream, similar to Figure 2.14. The OFDM symbols have a 48.4ns cyclic prefix and 194ns symbol length. OFDM mode uses the same LDPC codes as SC mode, and it uses an FFT size of 512. This corresponds to a subcarrier spacing of 5.16MHz, and there are 336

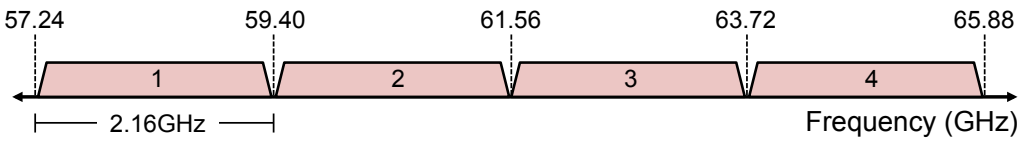


Figure 2.22: Frequency plan for IEEE 802.11ad. Channels 1-3 are used in North America and South Korea, channels 1-4 are used in Europe and Japan, and channels 2-3 are used in China.

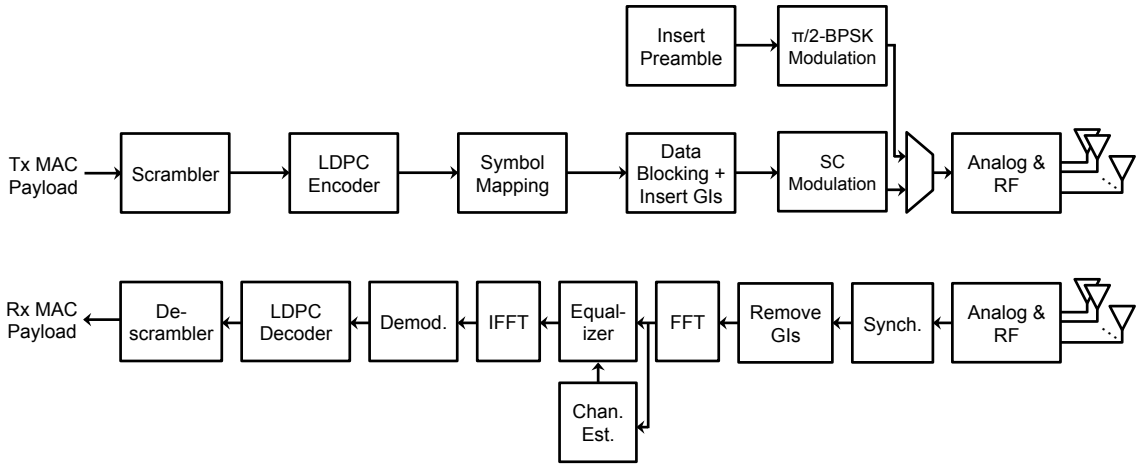


Figure 2.23: An IEEE 802.11ad transceiver.

data subcarriers and 16 fixed pilot tones. Symbols come from QPSK, 16-QAM, or 64-QAM constellations. The header has the same training and channel estimation fields as SC mode, but different signal information fields. The header takes 2.13μs to transmit. The achievable data rates range from 683Mb/s to 6.76Gb/s, which makes OFDM the highest throughput mode in IEEE 802.11ad.

Performance

IEEE 802.11ad has very limited deployment, and latency distributions are not widely available for environments where objects are in motion. The latency is difficult to estimate since the MAC and PHY are significantly different for IEEE 802.11ad compared to other amendments of IEEE 802.11. It is fair to say that the minimum latency (after initial beamforming) will be much smaller due to the higher data rates. Even with beamforming, the average latency might still be lower because of the data rates and the lower probability of collisions. The reliability will be similar to IEEE 802.11n and IEEE 802.11ac.

Table 2.2: History of cellular standards.

Gen.	Year	Standard	New Functions	Multiple Access	Modul-ation	Duplex-ing	Peak Data Rate
1G	1983	AMPS	Voice	FDMA	FM	FDD	N/A
2G	1990	GSM	↑ Voice QoS	TDMA/ FDMA	GMSK	FDD	6.5kb/s -13kb/s
2.5G	1998	EDGE	SMS	TDMA/ FDMA	GMSK	FDD	80kb/s -120kb/s
3G	2001	WCDMA	Multimedia	CDMA	DSSS: QPSK	FDD	384kb/s- 2048kb/s
3G	2004	HSPA	↑ Data QoS	TDMA/ CDMA	DSSS: 64QAM	FDD	DL: 14.4Mb/s UL: 5Mb/s
4G	2008	WiMAX	↑ Data QoS	OFDMA	OFDM: 64QAM	TDD, FDD	DL: 46Mb/s UL: 7Mb/s
4G	2010	LTE	M2M	OFDMA	OFDM: 64QAM	TDD, FDD	DL: 150Mb/s UL: 75Mb/s

2.3.6 Cellular (3GPP Release 8+)

Cellular networks consist of many cells that together cover a large geographic area. Cellular networks were originally intended to provide wireless voice services for a small number of terminals, but they have evolved to support a huge number of smart devices that require bursty, high-speed wireless data. In the future, they will support machine to machine (M2M) type communications where autonomous devices will communicate with each other without human intervention. To keep up with this increased demand for data and functionality, a succession of wireless standards based on different modulation techniques have been deployed over the years, which are summarized in Table 2.2. The current deployed standard is Long Term Evolution (LTE), which is standardized as 3GPP Release 8. 3GPP Releases 9 and 10 are currently being developed under the name LTE-Advanced.

In LTE, a large cell has a logical star network topology with a base station (BS), at its center and the mobiles (UE) at its branches (Figure 2.24). The BS controls communication resources for all devices in the network, making each cell a centralized network. BSs for large cells are connected to each other via a wired backbone that also connects them to landline phone and data/Internet services. In addition to large cells, there can be so-called small cells

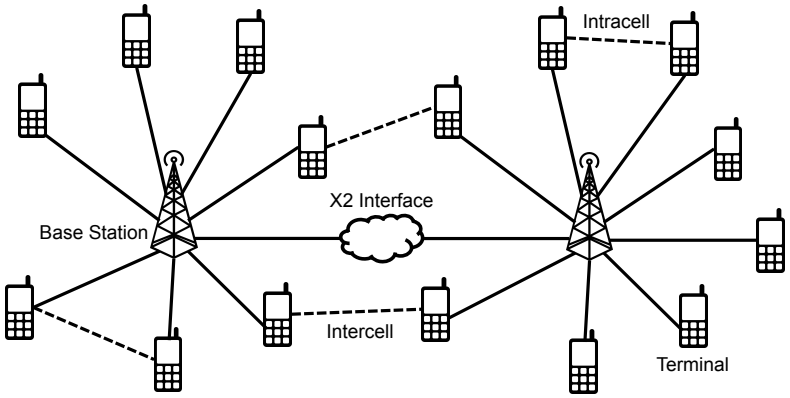


Figure 2.24: Network topology for the LTE cellular standard. Users connect to a single base station, and base stations can communicate with other base stations using an X2 interface. Depending on the type of modulation and the frequency reuse, users in the same cell or users in different cells may interfere with each other. Connections to traditional phone service and data services exist, but are not shown.

that have a smaller range, but provide better links for users near the cell edge or in an area with many obstructions. A small cell BS can connect directly to the wired infrastructure using an Internet connection, or they can connect wirelessly to large cell BSs.

When a UE moves to the edge of one cell's range, it is handed off to another cell (large or small) that has a stronger link between its BS and the mobile. Neighboring cells may use the same frequencies at the same time and cause intercell interference. Small cells transmit at lower powers to avoid interfering with large cells that it may be located within. Depending on the access methods used and devices' RF characteristics, intracell interference between UEs in the same network is possible (particularly with 3G systems using CDMA).

LTE networks have a somewhat different structure than previous cellular networks and networks conforming to the OSI model. The upper layers have been collapsed in order to lower overall latency in the data plane (for data exchange between the BS and UE) and the control plane (for exchanging network control information). Layer 2 contains three sublayers: packet data convergence protocol (PDCP), radio link control (RLC), and MAC. The PDCP is responsible for IP header compression, in-sequence packet delivery, and ciphering. The RLC handles error correction through ARQ and data segmentation or concatenation for efficient use of the communication resources. The MAC layer schedules transmissions and provides further error correction through hybrid-ARQ. Layer 1 is the physical layer, which uses orthogonal frequency division multiple access (OFDMA) modulation on the downlink (DL) and single-carrier frequency division multiple access (SC-FDMA) modulation on the uplink (UL). The next two sections discuss the MAC and PHY layers in more detail since they differ significantly from previously discussed networks.

MAC

LTE's MAC layer manages the scheduling of the entire network's communication resources in the face of changing channel and buffer conditions and incorrectly received packets. It also handles the discovery of new UEs and the handover of UEs from one BS to another.

Scheduling Based on current channel conditions, UE DL and UL buffer status, and the interference situation in neighboring cells, the BS schedules both DL and UL resources on the physical channels (specific communication resources) that carry user data, called the physical downlink shared channel (PDSCH) and physical uplink shared channel (PUSCH), for all connected UEs in the network. Since LTE uses OFDMA and SC-FDMA, resources are scheduled both in time and frequency. The algorithm used to allocate these time-frequency resources to each user is not standardized, but some methods in the literature include the fair throughput scheduler that tries to give all users the same overall throughput, the fair time scheduler that grants the same number of resources to each user, the maximum C/I scheduler that grants more resources to users with better channels, and the proportional fair scheduler that balances the a user's channel conditions with the throughput they have already received.

Most commonly, resources are granted dynamically where the BS sends scheduling information to the selected UEs every TTI (1ms). For the PDSCH, the scheduling information is sent on the physical downlink control channel (PDCCH) in the same TTI as the data on the PDSCH. For the PUSCH, the scheduling grant is sent on the PDCCH 4 TTI before the data may be sent on the PUSCH. The scheduling grants on the PDCCH indicate not only the resource blocks to use, but they also indicate the signaling parameters to use, such as constellation and code rate.

Alternatively, resources can be granted on a semi-persistent basis where a BS indicates the initial time for the resources and a periodicity for which the resources are granted to the UE, which lasts until further notice. If a dynamic scheduling command is detected, it takes precedence over the semi-persistent scheduling in that sub-frame. Semi-persistent scheduling is useful for latency-constrained applications, such as VoIP.

Adaptive Modulation and Coding In order to make each transmission as efficient as possible, LTE adapts its communication parameters to the current channel conditions. Using the reference signals embedded in each slot, each UE calculates a channel quality index (CQI) that estimates the SINR on the subcarriers, a rank indicator (RI) that specifies how many layers can be decoded, and a precoding matrix indicator (PMI) that selects the precoding matrix to be used. These are fed back to the BS on the physical uplink control channel (PUCCH) periodically or aperiodically. The CQI is used to select a modulation and coding scheme (MCS) for the DL transmissions based on a look-up table. The RI and PMI are used to select the MIMO mode.

Similarly, the BS selects an MCS for each UE's UL transmission based on sounding signals sent by the UE. The choices are communicated to the UE when sending it a scheduling grant. Also, based on the link quality measured during each UE's transmissions on the PUCCH or PUSCH, the BS decides if the UE should increase or decrease its transmit power on that channel. The power control commands are sent on the PDCCH.

Retransmission Handling To correct errors that were not able to be corrected by the channel coding in the PHY layer, the MAC uses hybrid ARQ (HARQ). It operates in a similar fashion to ARQ, but, instead of throwing away previous transmissions, HARQ makes use of the original transmission and all retransmissions in one of two ways. When using Chase combining, the transmitter sends the same packet each time, and the receiver calculates the overall received soft information as a weighted average, where the weights are the channel strength for that transmission. When using incremental redundancy HARQ (IR-HARQ), new parity information that can be decoded independently of other transmissions is transmitted each time, and all information is used in the decoding process.

LTE uses 8 parallel stop-and-wait HARQ processes, meaning it can support transmissions for up to 8 codewords at a time, and each codeword can be retransmitted a maximum of 4 times. This approach helps to reduce latency and increase throughput since the transmitter is not stalled while waiting for one packet to be received correctly.

On both the DL and UL, the ACK/NACK feedback (on the PUCCH for the UL or physical HARQ indicator channel (PHICH) for the DL) must arrive 4 TTI (4ms) after the initial transmission. On the DL, the retransmission may occur at any time after the initial transmission and may be on a different set of subcarriers. On the UL, the retransmission occurs at a pre-defined time after the initial transmission, and it is on the same set of subcarriers as the original transmission. The minimum round-trip time for one retransmission on the UL or DL is 8 TTI (8ms). Figure 2.25 shows an example of scheduling with retransmissions for an FDD system.

If higher error rates are acceptable, the system may run in unacknowledged mode to reduce latency. This may be desirable for multimedia streaming applications or VoIP.

Random Access A scheduled system must have a method for new devices to join the network and to be assigned communication resources. LTE uses a dedicated random access channel for this purpose. Connecting to the network has 5 steps: (1) UEs listen for the PSS and SSS sequences to synchronize to the BS and then for the cell-specific information on the physical broadcast channel (PBCH) to identify the physical random access channel (PRACH) resources and signaling parameters; (2) each UE randomly selects one out of 64 sequences (preambles) and transmits it over the PRACH; (3) the BS responds on the PDCCH with a PUSCH resource grant, an ID, and the amount that the UE must delay its transmission for each preamble received (timing advance); (4) UEs send a frame on its assigned PUSCH resource; (5) the BS acknowledges each UE with an ending frame. If more than one UE chooses the same preamble during (1), they will collide in (4). After discovery,

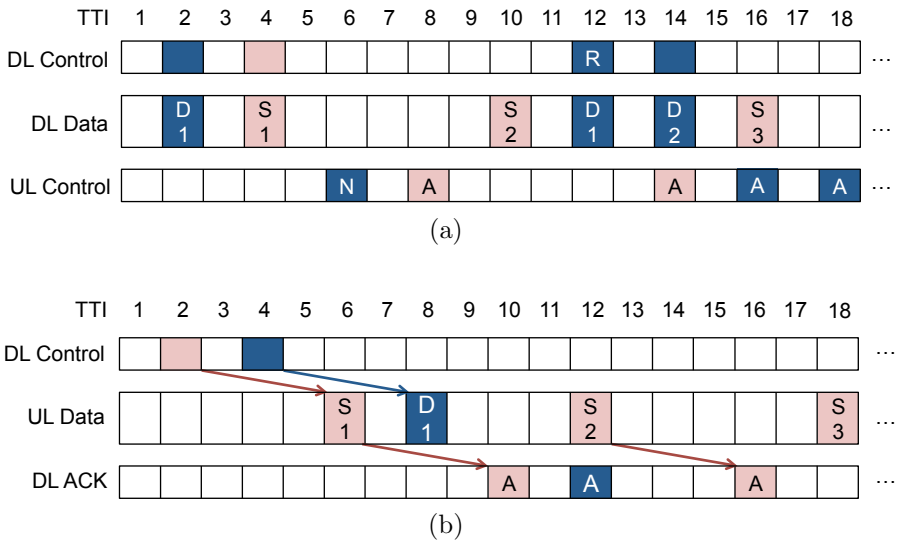


Figure 2.25: Examples of FDD resource allocation scheduling and resource usage for the (a) downlink and (b) uplink of LTE networks.

the connected UEs can request resources on the PUSCH. In the case of the BS handing over a UE to another BS, the random access is not contention-based, meaning no collisions can occur, because the new BS assigns a preamble and PRACH resources for the UE.

While connected to the network, the UEs may reach a time when they do not have resources assigned to them on the PUSCH or PUCCH channels and have data to transmit. If this occurs, they must request resources on the PRACH.

Since the UEs are mobile, the original timing advance values given to the UE by the BS that ensure each UE's transmissions arrive at the BS at the same time become outdated. In order to keep the simultaneous transmissions from different UEs orthogonal, the timing advance must be updated. The BS accomplishes this by sending updated values on the PRACH channel.

PHY

LTE currently defines 37 frequency bands ranging from 700MHz to 3.6GHz, and the bands used depend on region and cellular provider (2.26). The total width of bands vary from 20MHz to 200MHz, and they can be broken up into 1.25MHz, 2.5MHz, 5MHz, 10MHz, 15MHz, or 20MHz channels. Non-reserved bands between 1 and 28 use FDD, and bands between 33 and 44 use TDD. The range of a standard BS and UE may be up to 100km in ideal environments, but more typical ranges in urban and suburban environments are 0.6km to 1.4km and 1.5km to 3.4km, respectively [47].

LTE has a different PHY for the DL and UL because of the difference in BS and UE specifications. In particular, the majority of traffic is on the DL because users download

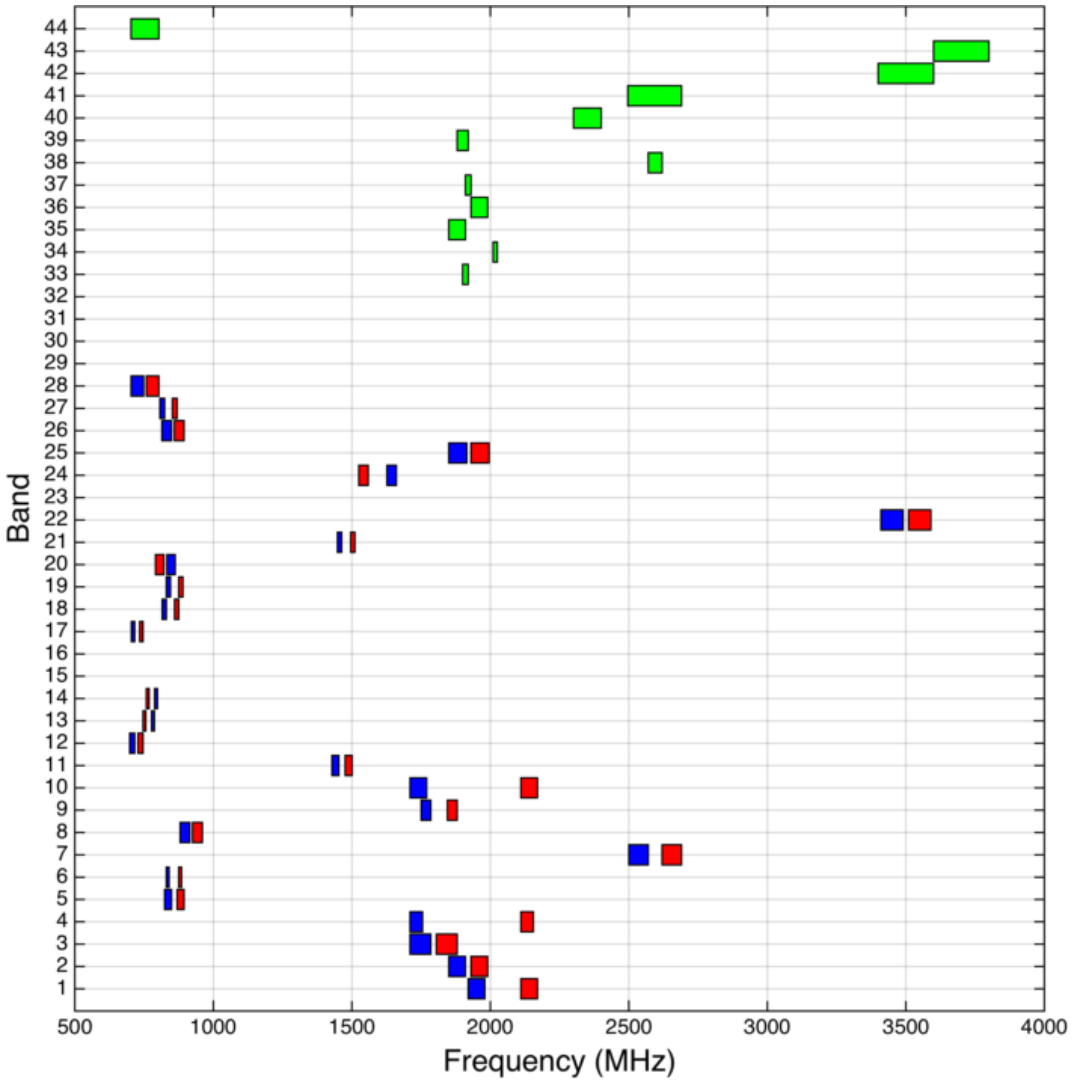


Figure 2.26: LTE frequency band plan, where bands 1-28 are for FDD (blue is used for the UL, and red is used for the DL) and bands 33-44 are for TDD.

large files, and users want to a long battery life for their UE so that they do not have to recharge it often. For these reasons, the DL uses a higher power, higher data rate PHY based on orthogonal frequency division multiple access (OFDMA) to support large file downloads, and the UL uses a lower power, lower data rate PHY based on single carrier frequency division multiple access (SC-FDMA) to maximize battery life. The DL supports up to 4x4 MIMO (4 transmit chains on the BS and 4 receive chains on the UE), and the UL supports up to 1x2 SIMO (1 transmit chain on the UE and 2 receive chains on the BS). The DL and UL support a maximum useful data rate of 300Mb/s and 75Mb/s (using highest order

Table 2.3: LTE physical parameters related to the OFDMA/SC-FDMA modulation.

Channel Bandwidth (MHz)	1.4	3	5	10	15	20
Subcarrier Spacing (kHz)	15					
Sampling Frequency (MHz)	1.92	3.84	7.68	15.36	23.04	30.72
FFT Size	128	256	512	1024	1536	2048
Non-Guard Subcarriers	72	180	300	600	900	1200
Number of Resource Blocks	6	15	25	50	75	100

modulation/MIMO and 20MHz bandwidth), respectively.

Downlink Figure 2.27a shows the transceiver architecture for the DL PHY based on OFDMA, which can transmit one or two code blocks at the same time. The code blocks must be between 40 and 6144 bits, including 24 CRC bits. If either MAC payload too large, it will be padded with zeros or segmented into smaller code blocks, each with their own CRC bits, that conform to the size constraints. Each code block is encoded using a rate 1/3 turbo code that outputs three streams. The streams are interleaved separately to protect against burst errors, combined, and then punctured to achieve the desired code rate in the rate matching block. The bits that are left out in the puncturing process can be used later in an IR-HARQ retransmission. The selected bits are scrambled and mapped to symbols from either a QPSK, 16-QAM, or 64-QAM constellation. The layer mapper segments symbols from each codeword into disjoint sets called layers. The precoder multiplies the layers by a space-time matrix that generates a sequence of symbols for each antenna port. The matrix used depends on whether the precoding is used for spatial multiplexing (increased throughput), transmit diversity (increased reliability), or if there is only a single transmit antenna. For more detail on layer mapping and precoding, refer to [48]. Next, the sequence is combined with reference symbols, and then it goes through the standard OFDM modulation whose parameters are given in Table 2.3. On the receive side, the reference signals are used to fine tune timing and frequency synchronization, and then the reverse processing is done to recover the transmitted data.

Uplink Figure 2.27b shows the transceiver architecture for the UL PHY based on SC-FDMA with the transmit chain of one UE and the receive chain of the BS receiving from multiple UEs. The channel coding processing is the same as with the DL PHY. The control payload for the PUCCH, which is encoded using a Reed-Muller(20,n) code with $n \leq 13$, is combined with the encoded and punctured data, and the resulting stream is scrambled. Next, groups of bits are mapped to symbols from either a QPSK, 16-QAM, or 64-QAM constellation. The symbols are grouped into blocks of size M_i, which is determined by the number of RBs allocated to the user. An M_i point FFT is taken of the block, and the

values are mapped to contiguous elements in an N point sequence, where N is the total number of subcarriers. The rest of the elements in the sequence are 0, except for those that carry reference signals. An N point IFFT is taken to convert the signal back to the time domain (but with frequency content only on the frequencies corresponding to the M non-zero subcarriers), the cyclic prefix is inserted, and the data is transmitted. On the receive side, the signal is corrected for time and frequency offsets using the reference signals, the cyclic prefix is removed, and the N point FFT is taken. Using the reference signals, MIMO channel estimation is performed, and the resource mapping is undone, and the result is processing using a MIMO receiver. After, the bits corresponding to different users are processed in separate chains that take the appropriately sized IFFT and undo the rest of the processing done in the transmitter, including splitting the data and control information.

Frame Structure In LTE, communication resources are divided into time-frequency blocks (Figure 2.28). For the DL and UL, time is divided into frames that repeat every 10ms. Each frame is divided into 10 1ms sub-frames, which are in turn divided into 2 0.5ms slots. The length of the sub-frame defines the transmission time interval (TTI). Depending on the bandwidth of the channel, each slot is divided into 6 to 100 resource blocks (RB) that are composed of 12 subcarriers. An RB is the smallest element that the BS can allocate. One slot has 6 OFDM/SC-FDMA symbols (66.7μs each) if a long cyclic prefix is used (16.7μs) or 7 OFDM/SC-FDMA symbols if a short cyclic prefix (5.2μs for the first cyclic prefix and 4.69μs for the rest) is used. For the DL, each RB has specific time-frequency blocks carrying known reference signals that are used for synchronization and channel estimation. With multiple transmit antennas, each antenna uses a disjoint set of time-frequency blocks for reference signals so that each element of the channel matrix can be estimated independently. The distance between reference symbols in time and frequency relates to the maximum tolerable Doppler spread and the coherence bandwidth, respectively. For the UL, the reference signals are time-multiplexed with the data instead of time- and frequency-multiplexed as in the DL in order to maintain the single-carrier nature of the UL signal. The UL does not need to handle training multiple transmit antennas since UE have a single antenna.

Duplexing LTE supports both frequency division duplexing (FDD), where the DL and UL are on separate frequency bands and communication is full-duplex, and time division duplexing (TDD), where the DL and UL share the same band and communication is half-duplex. In FDD operation, the DL and UL bands each have the frame structure shown in Figure 2.28. In TDD operation, a disjoint subsets of sub-frames are allocated for the DL, the UL, and for switching between DL and UL. The switching sub-frames consist of DL pilot symbols, followed by a guard period, and then UL pilot symbols. Figure 2.29 shows the potential allocations of the sub-frames, and there are a total of 9 formats defined.

Physical Channels Groups of specific time-frequency elements in the frame form physical DL and UL channels and signals. The specific groups allocated to each channel are different

between FDD and TDD modes/formats and can be found in [48], and an example for the FDD DL is shown in Figure 2.30. The following are several of the DL and UL physical channels and signals:

D1. **PDCCH**: The Physical DL Control Channel is used for DL resource grants in the current TTI, UL resource grants in 4 TTI, and for UL power control.

D2. **PDSCH**: The Physical DL Shared Channel is used for transmitting data and control information that could not fit into the PDCCH from the BS to UEs.

D3. **PBCH**: The Physical Broadcast Channel carries cell specific information that UEs use during cell discovery.

D4. **PHICH**: The Physical HARQ Indicator Channel is used for ACK/NACK feedback to the UE.

D5. **PSS/SSS**: The Primary/Secondary Synchronization Signals are used when discovering BSs and for DL time/frequency synchronization.

D6. **Reference Signals**: used for DL channel estimation and incremental synchronization between PSS/SSS transmissions.

U1. **PUCCH**: The Physical UL Control Channel is used for CQI, precoding matrix selection, ACK/NACK, and other channel-related feedback to the BS.

U2. **PUSCH**: The Physical UL Shared Channel is used for transmitting data and control information that could not fit onto the PUCCH to the BS.

U3. **PRACH**: The Physical Random Access Channel is used for UL path delay estimation and requests for resources on the PUSCH.

U4. **DRS**: The Demodulation Reference Signal is used for UL channel estimation.

U5. **SRS**: Sounding Reference Signal is used for UL timing synchronization and frequency-based scheduling.

Performance

The typical BLER after the PHY is tuned to be 10^{-1} by the adaptive modulation and coding. After HARQ in the MAC, the BLER drops to between 10^{-3} to 10^{-4}, and, after ARQ in the RLC, it drops to 10^{-6}. Latency for data transmissions is usually on the order of 30ms including retransmissions, but it can be as low as 5ms in ideal network and channel conditions with no retransmissions.

Improvements in LTE-Advanced

In LTE-Advanced (LTE-A), several more features are added to those of LTE, although the underlying PHY parameters and MAC functions remain the same. The already implemented and planned enhancements include the following:

1. **Increased MIMO Order**: The DL supports a maximum of 8x8 MIMO instead of 4x4 MIMO, and the UL supports a maximum of 4x4 MIMO instead of not supporting MIMO. This increases the number of streams that are spatially multiplexed or increases the number of redundant streams if diversity precoding is used.

2. **Channel Bonding**: The maximum bandwidth is increased to 100MHz by bonding contiguous or non-contiguous channels. This increases the maximum throughput of the system. The different channels will be processed by separate chains, so the PHY parameters stay the same.

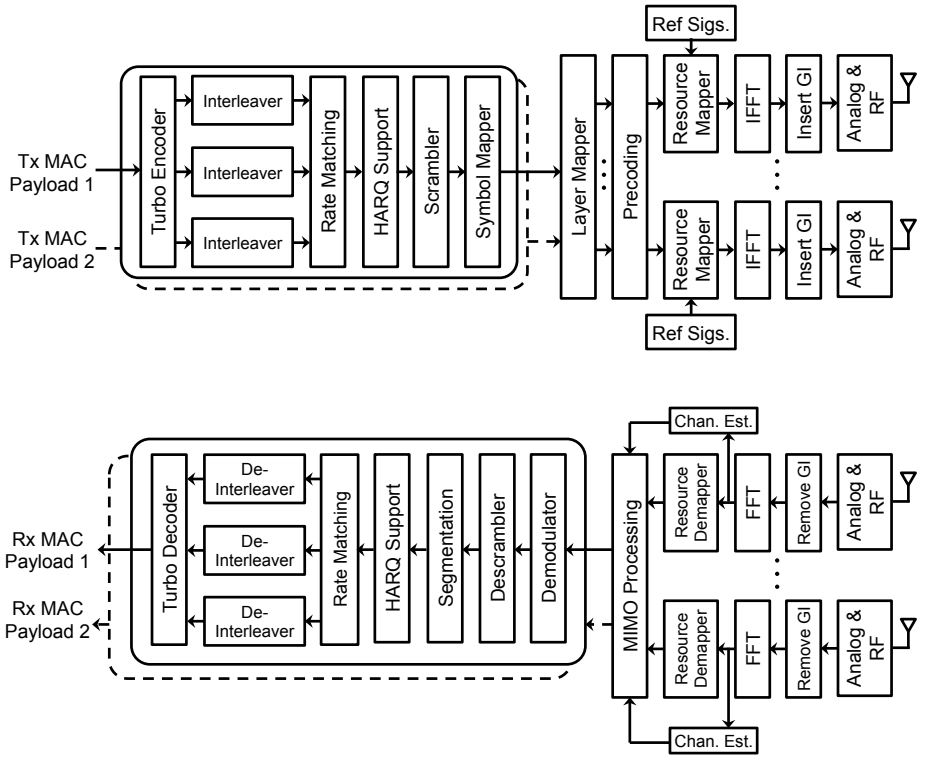
3. **Cooperative Multipoint**: Multiple BSs cooperate using their X2 interfaces so that they can beamform to a UE or jointly decode a UE's transmission. This increases the coverage and reliability of cellular networks.

4. **Distributed SC-FDMA**: the UE no longer has to map its resources to contiguous resource blocks, which allows additional freedom in frequency-based scheduling.

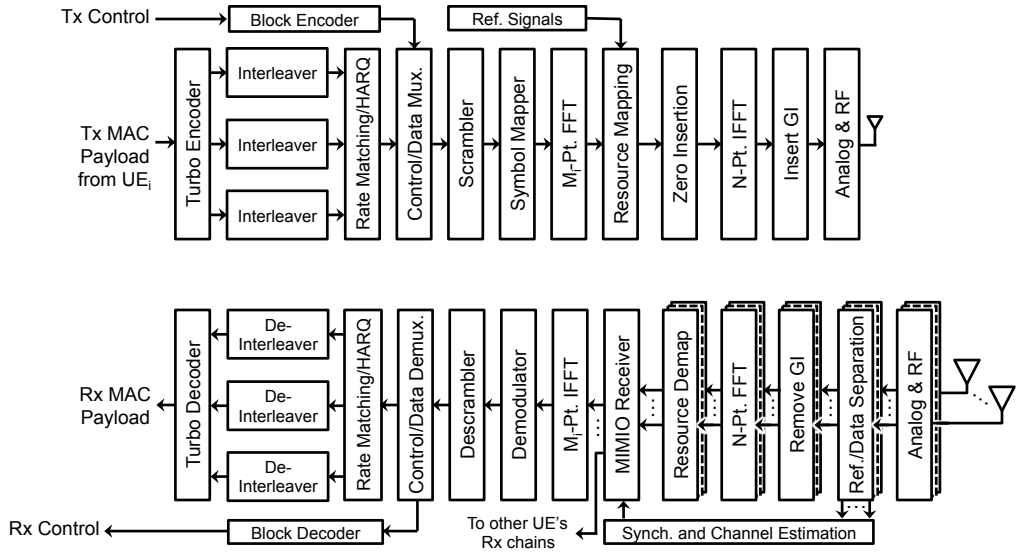
5. **Relaying**: UEs act as relays for other UEs in order to increase coverage, particularly at the cell edge or in rural areas with sparse BSs. This can also be used to increase spectral efficiency.

6. **Het-Nets with Interference Coordination**: Small cells are deployed that overlap with large cells in order to increase coverage and spectrum reuse. The overlapping cells coordinate with each other to reduce inter-cell interference.

In all, LTE-A increases the maximum throughput on the DL and UL to 1Gb/s and 500Mb/s, respectively. The latency for connecting to the network and transitioning from idle to active states will be reduced, but the minimum latency for data transmissions will be roughly the same as LTE (5ms).



(a)



(b)

Figure 2.27: LTE PHY processing for the (a) downlink (b) uplink.

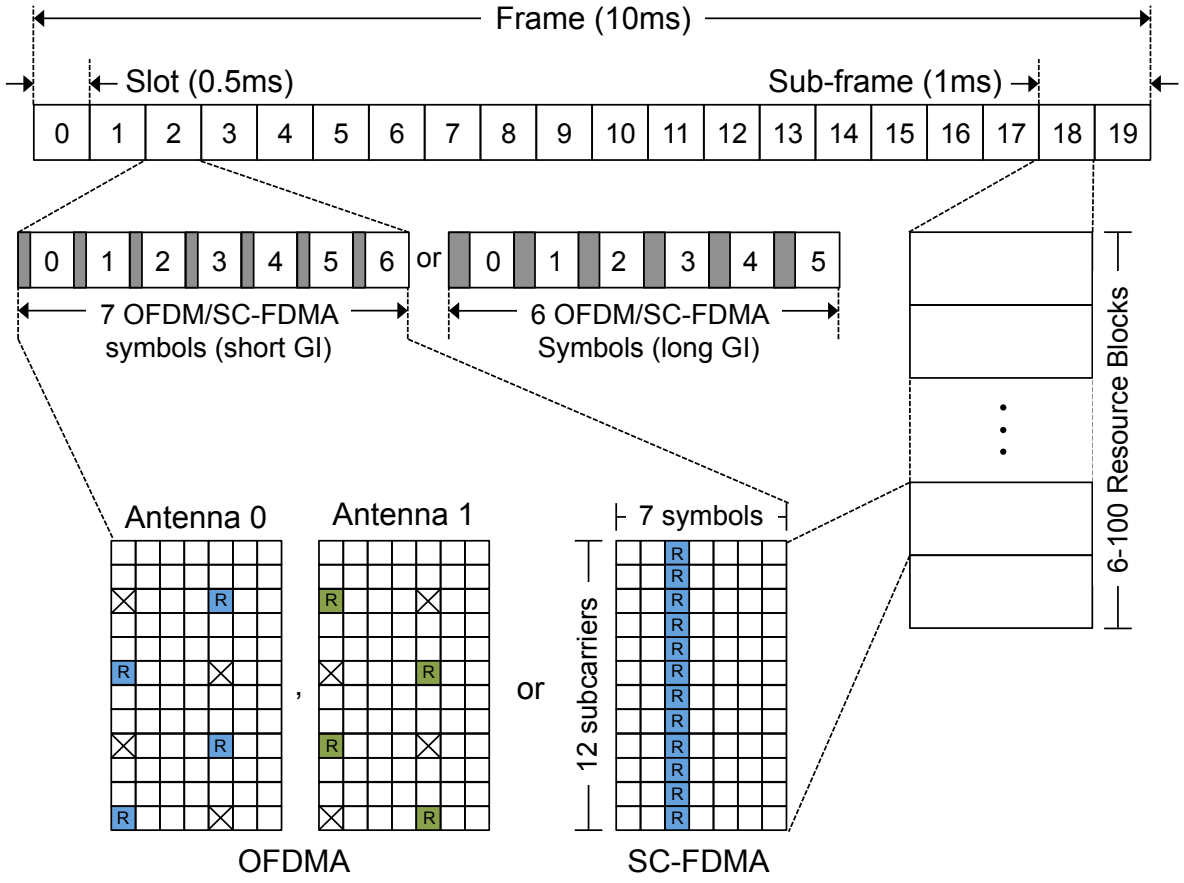


Figure 2.28: Resource mapping for the LTE DL and UL from frames, to sub-frames, to slots, to resource blocks. In each resource block, reference signals are embedded for synchronization and channel estimation. When using multiple antennas in the DL, additional disjoint time-frequency blocks are reserved for reference signals from each antenna that go unused by the other antenna (marked by "X").

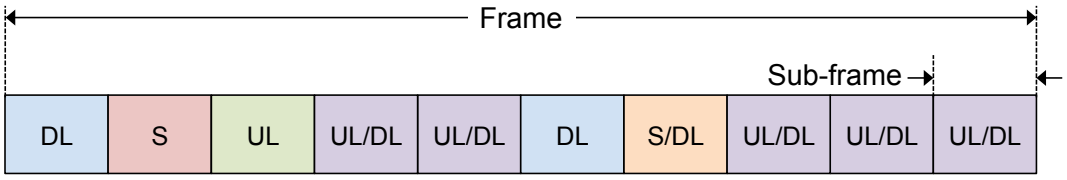


Figure 2.29: Frame format for the TDD mode of LTE. DL are dedicated downlink sub-frames, UL are dedicated uplink sub-frames, and S are guard sub-frames for switching between DL and UL. A "/" indicates that the sub-frame is used for one of the purposes listed and depends on the format used.

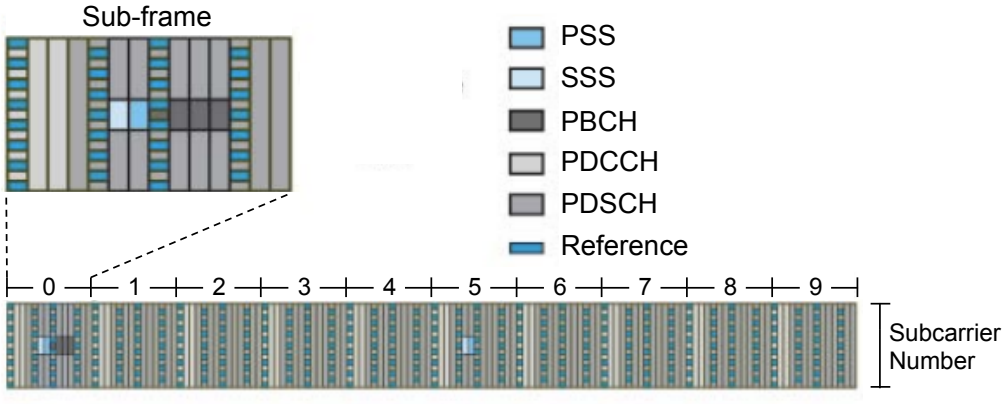


Figure 2.30: Time-frequency/resource block allocations to form physical channels for the FDD DL [49].

2.3.7 Wireless PAN (IEEE 802.15.4/ZigBee/WirelessHART)

Wireless personal area networks (WPAN) consist of many small, low-power devices that coordinate to observe or control a system. Prominent applications of WPANs include wireless building lighting systems, the smart home, smart grid, and industrial process automation. Two of the most popular and representative WPAN standards used for these applications are ZigBee [50] and WirelessHART [21]. Both ZigBee and WirelessHART use mesh network topologies that have a central coordinator but allow peer-to-peer links between the other devices (Figure 2.31). If a node cannot reach its destination directly, it uses other nodes as intermediate relays to deliver its information based on a routing table it creates dynamically. This provides a form of multi-user diversity.Like WLAN and Ethernet, all data is sent in the form of variable-length packets, and each layer encapsulates the data from the layer above it (Figure 2.32). ZigBee uses both the MAC and PHY layers defined in IEEE 802.15.4, and WirelessHART defines a different MAC layer but uses the IEEE 802.15.4 PHY layer.

ZigBee MAC

ZigBee's MAC layer shares many characteristics with that of IEEE 802.11. In a beaconed network, it divides time into superframes that are subdivided into 16 equally sized slots (Figure 2.33). The slots can be used for a beacon, contention access to the medium based on slotted CSMA/CA, or contention-free access to the medium where nodes have guaranteed time slots assigned by the coordinator in which to communicate (one of the 16 time slots).

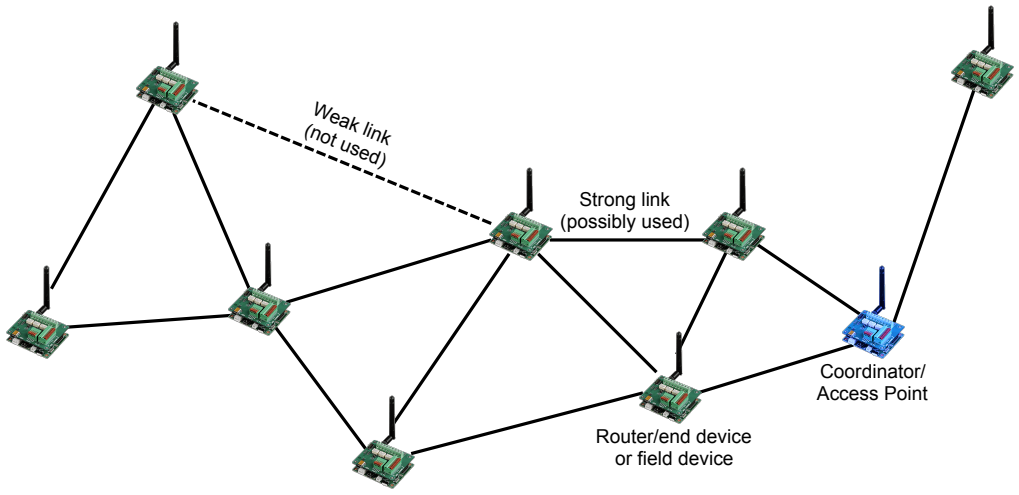


Figure 2.31: The mesh network topology for ZigBee and WirelessHART networks. In ZigBee networks, one device acts as a coordinator, and the rest are routers or end devices. In WirelessHART networks, several devices can be gateways connected to the same wired backbone network, and the rest are field devices.

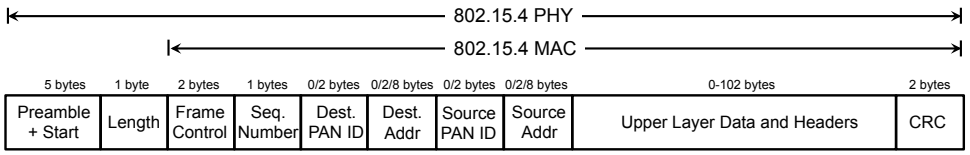


Figure 2.32: Packet structure for ZigBee and WirelessHART networks.

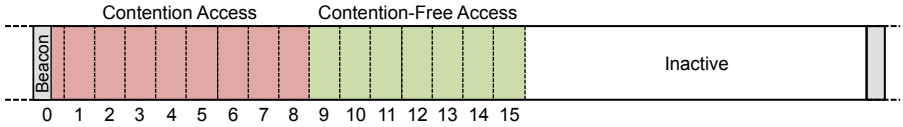


Figure 2.33: Superframe structure for ZigBee networks.

Figure 2.34 shows the timing of messages in the contention access period, which can operate in either an acknowledged or unacknowledged mode. In acknowledged mode, an incorrectly received packet (detected using a CRC check) will be retransmitted using the ARQ protocol. If the superframe is longer than the time for 16 slots, the devices are in an inactive state after the 16^{th} slot. In a non-beaconed network, there is only contention access to the medium using unslotted CSMA/CA. This operates similar to the timing diagrams in Figure 2.34, but backoff times are not synchronized between devices.

WirelessHART MAC

WirelessHART's MAC layer is based on a policy called time-slotted channel hopping (TSCH) (Figure 2.35). Like ZigBee, time is divided into superframes, which are divided into 10ms time slots. Each time slot has multiple frequency channels that different users can transmit over and not interfere with each other. In a dedicated time slot, a network manager attached to an access point assigns each link (pair of transmit and receive devices) a time slot and corresponding channel in the superframe that only that link can use. In subsequent superframes, the relative time slot stays the same, but the frequency channel assigned to the link is incremented based on a predetermined value (and wraps around after the last frequency channel is reached). In a shared time slot, multiple links use the time slot using CSMA/CA, which is advantageous if the links have a low probability of sending a transmission.

In a given time slot and frequency channel, the receive devices begins listening when the time slot begins, and the transmit device begins transmitting after a fixed time from the beginning of the slot. This eases the synchronization requirements between devices since the transmitter and receiver do not need to begin as soon as a slot starts. After receiving a transmission, the receive device sends an ACK back to the transmit device. In the case

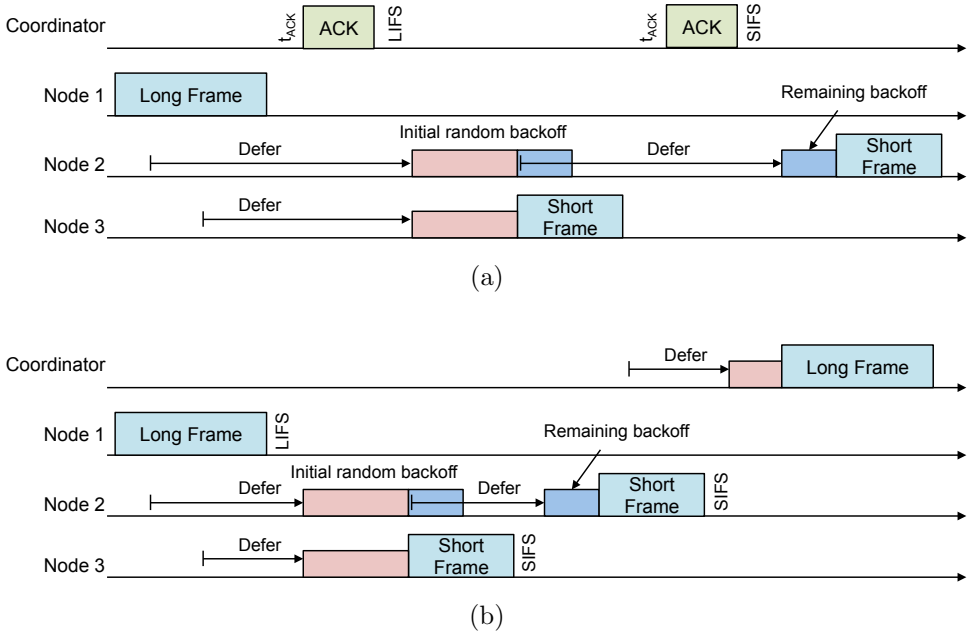


Figure 2.34: Contention access protocol for ZigBee networks (a) with acknowledgments and (b) without acknowledgments.

of a NAK (caused by a failed CRC check), the transmit device uses the ARQ protocol to resend its information in the next superframe in the next planned frequency channel, which increases the probability of successful transmission due to frequency diversity.

Common PHY

The version of the IEEE 802.15.4 PHY layer shared by ZigBee and WirelessHART is much simpler than that used in IEEE 802.11 and 3GPP Rel 8+ in order to reduce power. It specifies three frequency bands that devices can use that contain a different number of frequency channels: 868MHz (1 channel), 915MHz (10 channels), and 2.4GHz (16 channels). Devices that use the 868MHz/915MHz bands use a different PHY than those that use the 2.4GHz band. Both PHYs use direct-sequence spread spectrum modulation, which multiplies each symbol of a a lower rate signal with a an N-bit pseudo-noise (PN) sequence (the N samples of the symbol multiplied by the PN sequence are called chips). This increases, or spreads, the bandwidth of the signal, but it also makes it more immune to noise and frequency-selective fading. Neither PHY uses error control coding, MIMO, or other advanced signal processing. The PHY only maps bits to symbols, maps symbols to chips, and modulates the chips. Table 2.4 summarizes the parameters of the PHY for each frequency band.

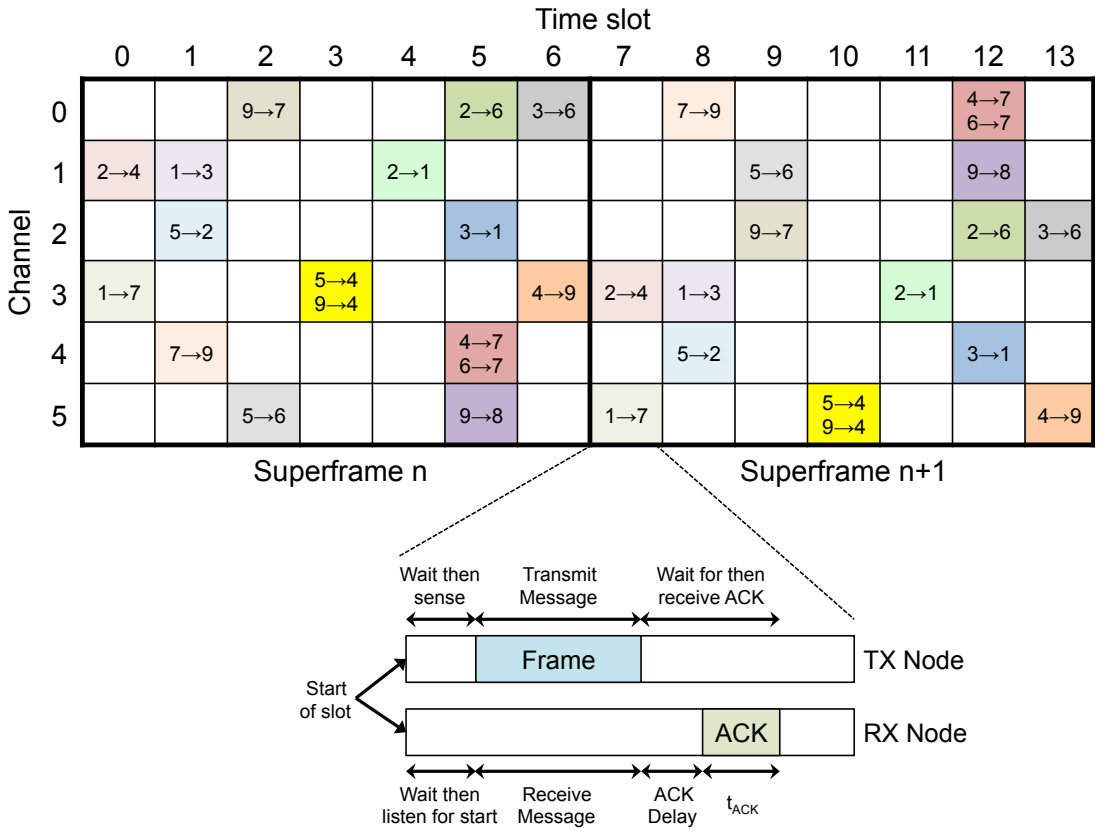


Figure 2.35: The time-slotted channel hopping media access protocol used in WirelessHART networks.

Frequency Band (MHz)	Channel Width (MHz)	Bit rate (kb/s)	Data Modulation	Spreading Factor	Chip rate (kchips/s)	Chip Modul.
868	1	20	BPSK	15	300	BPSK
915	2	40	BPSK	15	600	BPSK
2400	5	250	16ary-orthog.	32	2000	O-QPSK

Table 2.4: PHY parameters the 868MHz/915MHz and 2.4GHz PHY layers of IEEE 802.15.4.

Performance

ZigBee and WirelessHART typically have low reliability individual links with a BLER of 0.01, but the overall network is more reliable due to retransmissions. For example, WirelessHART achieves an end-to-end BLER of 0.001 including all intermediate hops. ZigBee can achieve similar reliability. The price paid for this is in the networks' latency. WirelessHART has a minimum latency of 10ms per hop, but, if retransmissions are needed, they must wait

until the next superframe, which occurs at least 70ms later. Similarly, ZigBee's latency for a single hop is on the order of milliseconds, and any retranmissions must wait until the next superframe, which is a minimum of 15.36ms.

2.3.8 Summary

Tables 2.5 and 2.6 summarize the main features of the wired and wireless standards discussed above, respectively.

Table 2.5: Summary of wired standards.

	IEEE 802.2/802.3an	SERCOSIII	ITU G.993.2	DOCSIS 3.0
Network structure	Star	Line or ring	Star	Bus
Medium access	Switched	Switched	Switched	Scheduled
Retransmissions	ARQ	-	ARQ	ARQ
Signaling	Baseband	Baseband	DMT/OFDM	SC (per channel)
FFT Sizes	-	-	64-8192	-
Bandwidth (MHz)	500MHz	31.25MHz	30MHz total	6MHz per channel
Reference signals	Preamble	Preamble	Pilots	Preamble
FEC code types	LDPC	No FEC	Concatenated RS-Convolutional	DL: RS-TCM UL: RS or RS-TCM
FEC code rates	0.84	-	0.56-1	DL: 0.93-0.94 UL: 0.73-1
Line code	64b/65b	4b/5b	-	-
Modulation	DSQ-128 (PAM-16)	MLT-3	BPSK-32768-QAM	DL: 64-QAM, 128-QAM UL: BPSK-128-QAM
Peak data rate (Mb/s)	10000	100	DL+UL: 200	DL: 42.88 UL: 30.72
Typical latency	1-10μs	$\tilde{4}\mu$s	1-20ms	0.15-8ms
PHY BLER	10^{-8}-10^{-10}	10^{-8}-10^{-10}	10^{-5}	$10^{-4} - 10^{-6}$

Table 2.6: Summary of wireless standards.

	IEEE 802.11ac	IEEE 802.11ad	LTE	ZigBee	WirelessHART
Network structure	Star	Mesh	Star	Star or Mesh	Mesh
Medium access	1st: CSMA/CA 2nd: Scheduled	1st: Scheduled 2nd: CSMA/CA	Scheduled	1st: CSMA/CA 2nd: Scheduled	TSCH
Retransmissions	ARQ	ARQ	HARQ	ARQ	
Signaling	OFDM	SC or OFDM	DL: OFDMA, UL: SC-FDMA	DSSS	
FFT Sizes	64-512	512	128-2048	-	
Bandwidth (MHz)	20-160	2160	1.4-20	1, 2, 5	
Reference signals	Preamble, Pilots	Preamble	Pilots, Sounding	Preamble	
FEC code types	Convolutional, LDPC	LDPC	Turbo	No FEC	
FEC code rates	1/2, 2/3, 3/4, 5/6	1/2, 2/3, 3/4, 5/6	Punctured 1/3	1	
Modulation	BPSK-64QAM	BPSK-16QAM	QPSK-64QAM	BPSK, OQPSK	
Peak data rate (Mb/s)	6930	SC: 4620 OFDM: 6760	DL: 300, UL: 75	0.25	
Maximum antennas	8	-	DL: 4, UL: 1	2	
Multi-user MIMO	Yes	No	Yes	No	
Multi-hop	No	No	No	Yes	
Diversity sources	Frequency, Time, Beamform or Space-time BC	Frequency, Time, Beamform, Space-time BC	Frequency, Time, Beamform or Space-time BC	Time, Multi-user	
Typical latency	50μs-10ms	5μs-1ms	5-30ms	1ms-500ms	10ms-1s
PHY BLER	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}

2.4 Latency and Reliability

Latency and reliability will be extremely important for networks that support next-generation applications that involve closed-loop machine-to-machine communication. Instead of merely affecting the users' quality of service (QoS), it will dictate the success or failure of the application. A standard's latency and reliability is a function of all of its networking layers, so it is important to understand how each layer's design affects these parameters in order to minimize their latency while maximizing their reliability. Also, a solid understand of the limits on current standards' latency and reliability will enable the design of new system architectures that will support next-generation applications and beyond. Chapter 2.3 reviewed several standards in detail, and the rest of this section will focus on the limits to these standards' latency and reliability.

2.4.1 Primary Sources of Latency

A communication network's latency has both deterministic and random components that are either fixed or scale with the number of nodes in the system. The deterministic components set the minimum latency, while the random components shape the latency's distribution. Deterministic latency components consist of the time to transmit information and overhead (i.e. parity bits, reference signals, and control data), constant length hardware computations, and wait times between transmissions. The random components include the time to retransmit information and overhead when necessary, queuing delays, random wait times between transmissions, variable length hardware computations, and software computations for higher layer functions. The importance of each component on overall network latency will depend on the standard and implementation.

Ethernet

Ethernet can achieve low latency because each user has its own (possibly full-duplex) channel and switches arbitrate access when multiple users want to access the same channel. This eliminates scheduling overhead and retransmissions due to collisions; however, delays are increased when the network is congested because frames are queued for transmission and buffers may overflow, which requires retransmissions. Retransmissions are handled by the transport layer, which is usually implemented in software, meaning the time between retransmission is best effort. Depending on the Ethernet amendment, data rates are between 100Mb/s and 100Gb/s, which leads to a short data transmission time, but each transmission has significant overhead from the PHY preamble and all higher layer headers (especially when the number of information bits is small). To summarize, supporting multiple users with no collisions or scheduling overhead and high data rates enables low latencies over Ethernet, but best effort retransmissions on an uncoordinated, congested network and high overhead leads to large tails in the latency distribution and a lower bound on latency, respectively.

SERCOSIII

SERCOSIII specializes the 100Mb/s Ethernet standard in order to achieve predictable and lower latency. It restricts the topology to either a line or ring, which means each device has at most two neighbors. Traffic received from one neighbor is only forwarded to the remaining neighbor, and at most one packet is received from each neighbor every cycle. This eliminates collisions and network congestion. SERCOSIII does not support retransmissions because the ring topology sends redundant messages in each direction and the raw BLER is sufficient for the intended applications. All functions critical to latency are implemented in hardware (PHY and parts of the MAC), and no networking or transport layer is required for the critical traffic. Overhead for each transmission is minimized by sharing a header for all MDTs and ATs, and each device only modifies/reads its section of a packet and then forwards it immediately. In summary, SERCOSIII addresses most of Ethernet's latency weaknesses described above, so its latency is deterministic and limited by the data transmission times and switch forwarding times. The main drawback is that the restriction to line and ring topologies causes latency to increase linearly with the number of nodes.

DSL

DSL can have lower latency than Ethernet since it sets up a dedicated, full-duplex virtual circuit between the source and destination. Once the link is set up, packets are sent back-to-back without any delay between them, and the routers do not spend time examining the headers to see where the packets should go. Each packet has a 53 bytes of overhead, which is fairly large. As in Ethernet, retransmissions are handled in the transport layer (implemented in software), so there is a large random delay for the transmission of the acknowledgment and the retransmission of data. When the network is congested, a user with a virtual circuit already set up does not suffer any performance degradation. However, other users may not be able to set up a dedicated circuit for communication, in which case they will have to wait until a clear path opens before they can transmit. Current versions of DSL have asymmetric throughputs for the DL and UL due to unequal bandwidth allocations, and the UL data rate is significantly slower than the DL's, which increases the latency on the UL. In summary, DSL has low latency when looking at a single link, but many links may have trouble all achieving low latency because of congestion. Also, the moderate data rates on the UL and the large overhead per packet will lead to larger total data transmission times.

DOCSIS

DOCSIS has higher latency than other wired standards partly because up to 8192 users share the same physical bus, so the shared channel's resources must be divided up among all active users. A scheduler arbitrates access to the channel, which incurs delay on both the DL and UL. Although the DL is centralized at the CMTS, there can be large buffering delays. On the UL, users must use periodic MAP slots to request communication resources in a contention-based manner and wait for a response from the CMTS. The interval between

MAP slots and the subsequent resource grant time can lead to an additional latency of at least 6-10ms on the UL in lightly congested networks (collisions in a congested network can increase this significantly). Furthermore, when even a small percentage of users are active, the amount of resources for each user is small, which leads to low data rates. The amount of resources allocated to the UL is much smaller than that allocated for the DL, which limits the performance of a network that has balanced traffic flow in each direction. All of the above points are compounded when retransmissions, initiated by the transport layer, must be scheduled and transmitted. In summary, due to the scheduling approach's large wait times and the inefficient sharing of communication resources, the latency of DOCSIS can be very large.

IEEE 802.11ac

Wireless networks generally have higher latency than wired networks because, like DOCSIS, they share a common wireless channel that each device must access without corrupting other devices' data. IEEE 802.11ac achieves this primarily through the CSMA/CA protocol, which has a large amount of fixed and variable overhead that scales with the number of nodes. The fixed overhead includes the DIFS, preamble, SIFS, and ACK packets, which add a deterministic overhead of at least 108μs to each data packet. On top of this, every data packet contains cyclic prefixes for each OFDM symbol and headers from higher layers.

The main sources of random latency are the random backoff time and retransmissions. Before any data packet is sent, a node must wait its random backoff time. As the number of nodes increases, the chances of multiple nodes having the same backoff time increase, which increases the probability of collision and larger backoff times. Each time a receiver does not succeed in decoding a packet, either because of a collision, noise, or fading, a new packet must be sent with all the overhead from the original transmission including having to contend for the medium again. The secondary multiple access mechanism, polling, eliminates random backoff and collisions at the expense of increased deterministic latency. Each node must be polled by the AP before transmitting, all polls receive at least an ACK, and interframe spacings are still present. In either access mechanism, a beacon must be transmitted every superframe by the AP, which adds additional deterministic overhead. Also, the data rates adapts to channel conditions (between 8Mb/s to 6.93Gb/s), so the data transmission time varies with time.

IEEE 802.11ad

IEEE 802.11ad has potentially very low latency because of its high achievable data rates and the spatial selectivity of its transmissions coupled with peer-to-peer transmissions. The former reduces control and data transmission times, and the latter allows multiple pairs of users to communicate simultaneously without colliding and lightens traffic to and from the AP. To enable these features in the 60GHz regime, beamforming and beam tracking is necessary. Full beamforming training is required before initial channel access and whenever

the channel changes enough that the beam cannot be updated incrementally. Beam tracking is performed when channel conditions change enough to cause a degradation in performance. Both add random latency that scale with the number of nodes in the system because every node must perform this procedure and it is not known when or how much iteration it will take to converge on a beamforming result.

In a spatially crowded network, every sector that a user can transmit across may be occupied. If the system uses contention-based access, this will lead to many collisions and large backoff times even though each spatial sector has its own backoff timer. If the system uses contention-free access, it will have high scheduling overhead from scheduling. In either case, there will be a high overhead for beamforming because the channel will change by the next time a user can access the channel. Finally, no broadcasting is possible because all transmissions are beamformed, which removes the most efficient option for point to multipoint communication and requires the beacon to be repeated in each sector.

Every transmission contains a lengthy preamble sent at the lowest data rate. On top of this, SC systems have guard symbols in each SC-FDMA symbol, and OFDM systems have cyclic prefixes before every symbol. The guard symbols and cyclic prefixes are sent at the full-speed data rate, so their overhead is lower than that of the preamble.

ZigBee and WirelessHART

ZigBee and WirelessHART have the highest latencies of the wireless standards discussed because of low data rates, using multiple hops from the source to destination device, and unreliable links. Since the highest data rate is 250kb/s, transmitting just the preamble and ACK frame takes 0.5ms, and transmitting 250 bits takes 1ms! Multi-hop transmission helps to increase the reliability of single transmissions and reduce the number of retransmissions, but it requires multiple channel accesses, multiple transmissions, the propagation of acknowledgments, and the maintenance of routing tables. Routing tables must be rebuilt whenever the channels change between users, and it takes a long time to do this. Although the links are shorter and spread-spectrum techniques are used, the links are not reliable since transmissions are uncoded, which leads to more retransmissions.

ZigBee's channel access mechanisms are similar to those for IEEE 802.11ac, so it shares the same weaknesses. On top of this, each superframe has only 15 non-beacon communication slots, so in a crowded network many beacons are transmitted before every user gets a turn to transmit. WirelessHART uses a scheduled channel access mechanism coordinated by a central node, but it does not lead to low latency because slots are 10ms long and the schedule is only updated every 70ms. This sets the minimum delay for retransmissions to 70ms! In addition, both ZigBee and WirelessHART start transmitting a fixed time after a slot starts in order to ease the synchronization requirements, but this adds deterministic overhead to each transmission.

LTE

LTE has low latency for a wireless standard because of its strict scheduling of all transmissions and advanced communication techniques, but it still limited by significant deterministic and random overhead. The elimination of contention-based access and polling reduces multiple access overhead, but there remains overhead for sending control data that schedules each user's DL and UL transmissions. In addition, LTE does not use preambles before every transmission (which are usually sent at the lowest possible data rate), but it uses approximately 30% of the total available time/frequency resources for reference, control, and feedback signals. Note that the available subcarriers are only up to 60% of the FFT size because the rest are used as guards between adjacent channels. The DL reference signals' resources do not scale with the number of users, but the UL reference signals scale. The cyclic prefix and guard intervals are long compared to other standards in order to support high mobility. As with all other standards except SERCOSIII, LTE's PHY payload contains headers from all the higher layers.

LTE has random latency from resource allocation, adaptive coding and modulation, retransmissions initiated in either the MAC or transport layers, and the networking layers messages must travel through from source to destination. Depending on the scheduling algorithm and network conditions, DL and UL communication resources are not granted at predictable times. Since resources are not granted evenly and the constellation and coding rate adapt to the channel, data rates for each device are not constant leading to variable transmission times. Also, the adaptation can take several retransmissions to converge on the optimal system parameters. Retransmissions are handled at both the MAC and transport layers. The MAC layer's HARQ protocol requires an ACK/NAK to be sent 4ms after the initial transmission. If a user is the source, it must retransmit 4ms after a NAK. If the BS is the source, it can choose when to retransmit as long as it is at least 4ms after the NAK. There are 8 parallel HARQ processes, so it is not likely that all processes will require retransmissions. Up to 8 total retransmissions are allowed before the MAC layer gives up and the transport layer (implemented in software) resets the retransmission process using the ARQ protocol. Therefore, retransmissions can be very costly in terms of latency because the minimum turnaround time is 8ms, and it is not uncommon to exhaust all MAC retransmissions. When the transmission is received successfully, the data must propagate through all the networking layers to the core network and then back again through the layers to the target device, even when the target is in the same cell. The time to accomplish this is highly variable, and the latency specification for LTE does not include the core network latency.

Since LTE reserves resources for UL signaling requests on the PUCCH, more nodes in the network reduces the number of resources for the PUSCH and eventually no more nodes can be supported. This means nodes will have to resort to using the PRACH to request resources, which takes longer and can result in collisions. Semi-persistent resource allocation helps with this, but it can be difficult to schedule retransmissions when there are many preexisting semi-persistent allocations.

2.4.2 Primary Bottlenecks for Reliability

A communication network's reliability in terms of block error rate (BLER) depends on many factors, such as the channel, constellation, FEC, error detection codes, modulation technique, diversity, retransmission mechanisms, and the networking layer. Wired networks have an inherent advantage because their channels have less noise and interference and are time-invariant, allowing each link to be optimized for the specific channel. Wireless channels have to cope with fading, interference, and higher propagation losses (especially at 60GHz). For a given channel realization, higher order constellations, higher rate FEC codes, and FEC codes with shorter blocklengths have higher error probability. Likewise, error detection codes with shorter blocklengths have a lower probability of detecting errors.

Most wired standards in Chapter 2.3 rely on high-rate, moderate-blocklength FEC with ARQ to correct all errors. The only exception is SERCOSIII, which uses a redundant ring topology to increase reliability instead of FEC and ARQ and relies on a well-behaved wired channel. Link distance is one of the largest factors affecting the reliability of wired standards because many can have wires between 100m or several kilometers long.

On the other hand, wireless standards use a variety of techniques to increase reliability. First, all wireless standards have low-rate code options (except those based on 802.15.4 that do not use FEC) in order to have enough redundancy when channel conditions are poor. For example, the IEEE 802.11 standards have rates as low as 1/2, and LTE has a minimum code rate of 1/3. Second, retransmissions are used to correct any errors that make it past the PHY, and all standards use ARQ in the transport layer. LTE also uses HARQ with either Chase combining or incremental redundancy to correct errors in the MAC layer. Third, the standards use diversity and beamforming to provide multiple independent paths from the transmitter to the receiver and to boost received SNR. Transmit diversity sends a transmission from multiple antennas to a single receive antenna, and beamforming is a form of transmit diversity where all antennas transmit at the same time and add coherently at the receiver. Receive diversity involves a single antenna transmission being received on multiple antennas. MIMO techniques can achieve both transmit and receive diversity. Frequency diversity occurs when information is transmitted over a frequency selective channel and when the receiver can take advantage of this frequency selectivity, such in an OFDM system with FEC. Time diversity occurs when a FEC codeword is spread out over many coherence times so that it sees many different channels, which can occur with HARQ, or when information is retransmitted. Multi-user diversity arises when a transmission is relayed by different users on its way from the source to the destination. IEEE 802.11ac and LTE use transmit, receive, frequency, and time diversity and beamforming. IEEE 802.11ad uses massive beamforming and frequency diversity. ZigBee and WirelessHART use frequency, multi-user, and time diversity. Fourth, preambles and reference symbols embedded in transmissions are used to estimate and adapt to the channel. Finally, modulation techniques that increase SNR or mitigate channel effects are used. ZigBee and WirelessHART use DSSS that effectively acts as a repetition code for the data and increases received SNR. IEEE 802.11ac and the LTE DL uses OFDM(A), which is robust to intersymbol interference and allows frequency selective

power allocation.

The main factors affecting the reliability of wireless standards are collisions with other users due to uncoordinated channel access, coexistence with other systems in the same frequency channels, interference from users in adjacent channels, Doppler shifts from moving devices, the difficulty of synchronization, and time-varying channel effects (fading, shadowing). Also, the standards focusing on data rate (IEEE 802.11ac/ad and LTE) specify that the PHY BLER should be maintained at 1% by changing the constellation and coding rate. This maximizes system throughput, but limits the overall system reliability.

2.4.3 Conclusions

General Observations

Contention-based MACs with many nodes in the same collision domain and packet-based networks have a large deterministic and random latency overhead due to using a preamble, interframe spacings, and random backoffs. Systems with a central node scheduling medium access and that periodically broadcast reference signals have comparatively less overhead and can achieve tighter synchronization, but they have difficulty informing nodes of assigned retransmission slots quickly over poor channels. Also, they must reserve resources for the reference signals and for distributing the schedule, which decreases the useful data rate or limits the maximum number of connected users. In either type of network, higher networking layers add several bytes of headers to the PHY payload, which increases latency greatly.

To achieve high-reliability, error control is applied at multiple network layers in wired and wireless systems. In current systems, the PHY uses FEC, the MAC detects errors via a CRC code and may act on it using ARQ or HARQ, and the transport layer may use erasure coding and usually uses ARQ to correct any remaining errors. Error correction at the PHY requires the insertion of redundant bits (more for lower code rates) that adds a deterministic overhead. Depending on the standard, if an error is detected at the MAC layer, either a retransmission request is sent to the transmitter or it passes the erroneous packet to the transport layer. Retransmissions add random latency and the cost is usually far greater than additional redundant bits. In the former case, the round trip transmission time is added to the latency. At the transport layer, which is implemented in software, errors may be attempted to be corrected if the packets had erasure coding applied, and retransmission request will be sent if there are any errors remaining. Using erasure coding incurs the waiting time for for packets that are needed to correct the erroneous packet(s). Retransmissions require more time than in the MAC layer because the request must pass through more networking layers on top of the time for retransmission. Also, since the transport layer is implemented in software, there is additional nondeterministic latency added for when resources are granted by the OS.

In addition to error control, wireless systems use diversity to increase reliability. The most common forms are time and frequency diversity, but antenna and multi-user diversity are also used. Time diversity inherently adds latency since a message must span multiple coherence times, and multi-user diversity usually involves multiple hops between users controlled by

the networking layer. Depending on its implementation, antenna diversity may not add latency, but space-time codes can add multiple transmissions worth of latency and require a continuous stream of data to make it efficient. Frequency diversity usually does not require additional latency.

To guarantee low latency and maintain high reliability, the PHY needs to be as dependable as possible by using FEC and, for wireless systems, must have diversity mechanisms that do not require many transmissions or long time periods. Retransmissions should also be used and should be balanced with the FEC overhead, and they should be initiated in the MAC so that the request does not need to propagate to higher layers. Communication must be scheduled to avoid collisions and contention for the channel, and the overhead from preambles and headers must be minimized, which may require using a continuously connected system. Adaptation to the channel to achieve higher data rates should be avoided, and as many of the higher layers as possible should be eliminated.

Current Standards with Lowest Latencies and Highest Reliabilities

Tables 2.7-2.10 categorize and summarize the sources of latency and the factors affecting reliability for wired and wireless standards. Out of all the standards reviewed, SERCOSIII has the lowest, most predictable latency due to its sharing of headers, elimination of collisions, moderate data rate, ring topology, and scheduled communication. It has high reliability due to the redundant ring topology and the relatively rates over the low-noise, low-interference wired channels. Unfortunately, most of what makes SERCOSIII work can only be applied on wired networks.

Out of the wireless standards, LTE is the most promising for low-latency, high-reliability operation since it has tightly scheduled transmissions to eliminate collisions, uses advanced wireless techniques to achieve high peak data rates that can be divided among multiple users, uses a low base-rate Turbo code paired with HARQ, and has many options for diversity. However, LTE has several major drawbacks that increase its latency (both average and the density of the distribution's tail) and decrease its reliability, particularly the overhead and delay in granting communication resources, the delay between retransmissions, the preamble/header/cyclic prefix sizes, and the iterative process of adaptive coding and modulation. These drawbacks must be addressed in a wireless system designed for guaranteed low-latency, high-reliability operation that approaches the performance of SERCOSIII.

Table 2.7: Summary and classification of sources of latency in wired standards. ReTx includes the times for software detection of an error and request for the lower layers to retransmit, data transmission time, and time for additional acknowledgments. Grant is the wait time for the scheduling node to grant communication resources to the user.

		Ethernet	SERCOSIII	DSL	DOCSIS
Determ-inistic	Fixed	-	Preamble Headers	Null subcarr.	Scheduling request slots
	Scales	Data size Preamble Headers HW processing Initial ACKs	Data size HW Processing	Data size Preamble Headers HW processing Initial ACKs	Data size Preamble Headers HW processing Initial ACKs
Random	Fixed	-	-	-	-
	Scales	ReTx Queuing	-	Circuit setup ReTx	Grant (UL) Queuing (DL) ReTx Data rate

Table 2.8: Features of wired standards that affect reliability.

Standard	Increases BLER	Decreases BLER
Ethernet	LDPC FEC (version dependent) Single collision domain ARQ with CRC	Moderate FEC blocklength Moderate distance links
SERCOSIII	Redundant ring topology No collisions	No FEC No ReTx
DSL	RS-Convolutional FEC No collisions ARQ with CRC	High order constellations Static channel adaptation Long distance links
DOCSIS	RS-TCM FEC ARQ with CRC	Adjacent channel interference Long distance links

Table 2.9: Summary and classification of sources of latency in wireless standards. ReTx includes the times for software detection of an error and request for the lower layers to retransmit, data transmission time, and time for additional acknowledgments. Grant is the wait time for the scheduling node to grant communication resources to the user. Multi-hop is the overhead for additional data and ACK transmissions.

		IEEE 802.11ac	IEEE 802.11ad	ZigBee	WirelessHART	LTE
Determ-inistic	Fixed	Beacon Null subcarriers	Beacons Null subcarriers	Beacon	-	Ref. signals (DL) Null subcarriers Control chan. (DL)
	Scales	Data size Preamble/headers HW processing Initial ACKs Cyclic prefix Interframe spacing Pilot symbols	Data size Preamble/headers HW processing Initial ACKs Beamforming Scheduling Guard/cyc. prefix IFS Pilot symbols	Data size Preamble Headers HW processing Initial ACKs Pre-Tx wait IFS	Data size Preamble Headers HW processing Initial ACKs Pre-Tx wait Scheduling	Data size Headers HW processing Initial ACKs Ref. signals (UL) Cyclic prefix Scheduling
Random	Fixed	-	-	-	-	-
	Scales	Data rate Backoff ReTx	Data rate Backoff ReTx Beam refinement Grant (UL)	Data rate Routing table Multi-hop Backoff ReTx	Data rate Routing table Multi-hop ReTx	Data rate HARQ ARQ Grant (UL) Queuing (DL) Core network

Table 2.10: Features of wireless standards that affect reliability.

Standard	Increases BLER	Decreases BLER
IEEE 802.11ac	Convolutional or LDPC FEC Transmit/receive diversity Frequency diversity Beamforming Moderate subcarrier spacing Intra-frame pilot symbols ARQ with CRC	Collisions due CSMA/CA Adaptive coding and modulation Coexistence Adjacent channel interference Cyclic prefix short
IEEE 802.11ad	LDPC FEC Massive beamforming Sparse subcarrier spacing ARQ with CRC	Short FEC blocklength Adaptive coding and modulation Beamforming inaccuracy Guard interval short
ZigBee	DSSS processing gain Multi-hop/multiuser diversity ARQ with CRC Low-order constellations Looser synchronization allowed	No FEC Collisions Coexistence No intra-frame reference signals
WirelessHART	DSSS processing gain Multi-hop/multi-user diversity No collisions in scheduled slots ARQ with CRC Low-order constellations Looser synchronization allowed Frequency hopping	No FEC Coexistence No intra-frame reference signals
LTE	Turbo FEC No collisions HARQ and ARQ with CRC Transmit/receive diversity Frequency selective scheduling Beamforming Long cyclic prefix Intra-frame reference signals	Long distance links Adaptive coding and modulation Dense subcarrier spacing Tight synchronization required Inter-cell interference Adjacent channel interference

Chapter 3

System Architectures for Control Applications

In the near future, the number of wireless devices will outnumber humans by an order of magnitude, and most of these devices will communicate with each other instead of people. They will not only sense the environment, as most do today, but they will also manipulate it. This closed loop operation will not require the high data rates that today's people-centric networks provide; instead it will require low-latency and high-reliability communication at moderate overall data rates [1]. Currently, high-performance industrial control is one of the few applications that has similar requirements to the IoT of the future, but they exclusively use wired networks. Existing wireless systems and standards cannot achieve the latency and reliability required since they are designed for either high-throughput or low-power communication between a small number of terminals.

Wired control systems have many mechanical drawbacks and future IoT applications will require wireless networks, so a wireless solution must be developed. This chapter investigates system architectures for current industrial control systems. The proposed architectures focus on delivering the control data in a timely and reliable fashion, not on delivering high-speed data, which can be handled by a separate communication system or buffered ahead of time if needed. It is determined that the key component to decreasing latency for wireless control systems is increasing diversity, so a cooperative relaying system is proposed that uses a low-latency multi-user diversity technique to meet latency and reliability specification under harsh channel conditions.

Some or all of the work in this chapter appeared in [32] and [33]. Additional details are given in Chapter 3.3 and Chapter 3.4.

3.1 Wireless Control Systems

3.1.1 Control Systems

The objective of a control system is to make the outputs of a dynamic system behave in a desired way by manipulating the inputs to the system. Control systems have three basic elements: sensors, actuators, and controllers. The sensors measure a subset of the state variables of the system, the actuators manipulate the inputs of the system, and a controller is the algorithm that determines how to adjust the inputs based on available measurements and information from other controllers.

At the lowest level, control systems are either centralized or distributed. Centralized systems have a single decision maker because there is either only one control system involved or because multiple control systems communicate their sensor data to a central processor that makes decisions for everyone and relays those decisions back. Decentralized systems are simply systems that are not centralized, which can be thought of as having multiple control systems that make their own decisions based on their own measurements. In other words, passing information between the control systems can only occur by measuring the effect of other control systems' manipulations on the dynamic system instead of sharing sensor data.

The number of sensors and actuators in each control system depends on the system's dynamics and the desired level of control over the system's outputs. Simple systems may require only a single sensor and actuator, in which case the control system is called single-input single-output (SISO). These types of systems include cruise control in automobiles and temperature control in furnaces. More complex systems require the manipulation and observation of many state variables, so more actuators and sensors are put into the control system. This is called a multiple-input multiple-output (MIMO) control system. MIMO control is usually applied to high-performance systems that cannot be decoupled in to multiple SISO systems, such as robotics, aeronautics, and chemical and industrial processes.

The design of controllers for centralized and decentralized SISO and MIMO systems has been studied extensively. Classic techniques, such as PID controllers designed in the frequency domain, are used for centralized and some distributed SISO control systems. MIMO control systems require modern state-space techniques and detailed models of the dynamic system in order to use all of the available inputs to achieve desirable behavior in all of the outputs. For more information on the design of controllers, the interested reader is referred to [51, 52].

When the control system is implemented, a communication system links the controller to the actuators and sensors in a logical star network topology with the controller at the center, although the physical network topology does not have to be a star. Figure 3.1 shows a block diagram for an implemented control system that can either be a centralized single system or one element of a distributed system. The links might have random, unbounded delays associated with them, and they can inject (non-Gaussian) errors into the transmitted data. The distribution of the latency and the errors depends on the network size and topology, the medium over which the information is transmitted, and the protocols that are used, as

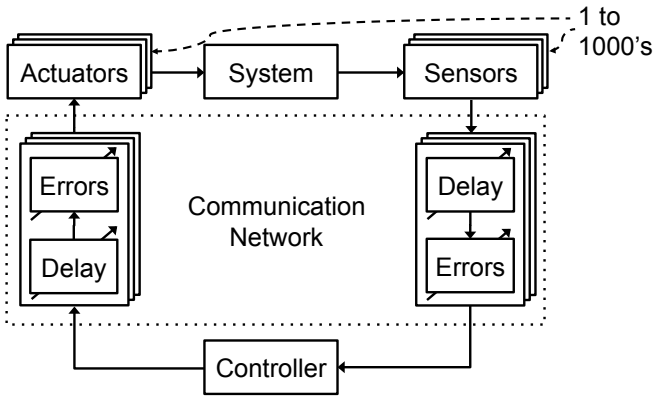


Figure 3.1: Components of a single implemented control system.

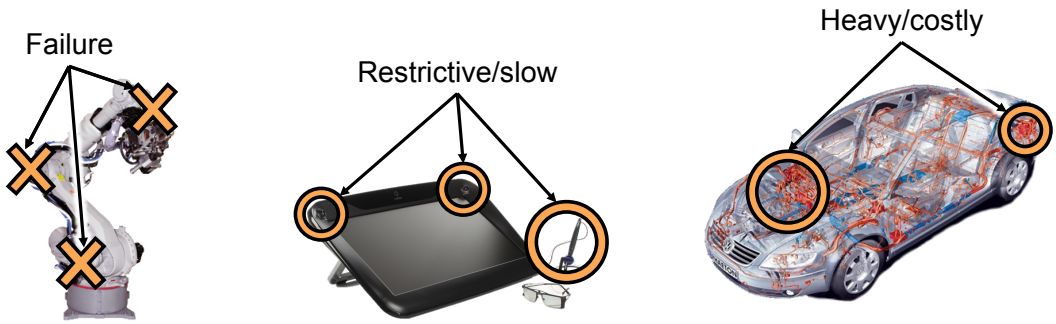
discussed in Chapter 2.4. Both communication latency and communication errors can have serious consequences on the control system.

Controllers are usually designed assuming no delay for transmitting measurements from the sensors to the controller, computing instructions for the actuators, and transmitting those instructions from the controller to the actuators. Delay degrades the performance of the control system. If the communication delay of a SISO control system is larger than 20-60% of the time constant of the closed loop system, here called the critical delay t_{crit}, the controller cannot respond to changes in the system quickly enough, and the control system fails [53]. MIMO systems are affected by delay similarly, and they have more severe latency issues because of its possibly large number of sensors and actuators.

In addition, control systems are typically designed to tolerate modeling errors or disturbance signals from a particular set, but anything outside of this can cause the system to act unpredictably. As the designs become more aggressive, they cannot be as robust. Errors in the transmitted data can cause the system to fall out of specification or become unstable. Since subsequent measurements are correlated, the system's state may be estimated if a sensor measurement is lost or has errors [54], but this results in suboptimal performance at best. Errors in instructions transmitted to the actuators cannot be corrected in the same manner since the actuators simply execute the received instructions, so this situation is best avoided. Again, MIMO systems must be particularly aware of this because the overall probability of error with many sensors and actuators in the system is much higher than if there was just a single sensor and actuator.

3.1.2 Drop-In Wireless Replacement for Wired Industrial Control

Control systems with a small number of sensors and actuators and plenty of margin in their design do not have to worry much about the communication network that links all of



(a) (b) (c)

Figure 3.2: Drawbacks of using wires: (a) wires are the main point of failure in industrial and medical robotics; (b) wires lead to restrictive and processing-intensive solutions in human-computer interfaces and immersive environments; (c) wires are one of the heaviest and most costly components in automobiles and aircraft.

their components because they can tolerate the network latency required to have error-free operation using long-blocklength, low-rate error control codes and as many retransmissions as necessary. However, large MIMO systems that have tight specifications that limit their robustness will have strict specifications on the communication network that they use. The more sensors and actuators that are added to the system, the stricter the specification will be, which will limit the design options.

Industrial control systems fall into the latter category, particularly manufacturing and medical robotics, automobiles, aircraft, human-computer interfaces, and immersive environments. They require a cycle consisting of all sensors reporting observations and all actuators receiving instructions to be on the order of milliseconds, and an error in any part of the cycle must occur with probability 10^{-8} or lower. All of these applications have addressed their communication problem by using wired networks, such as SERCOSIII, since they provide lower latency and higher reliability than existing wireless networks, as discussed in Chapter 2.3. Unfortunately, using wired networks has many drawbacks in practice. Wires are the main point of failure in robotics since the wires fatigue and eventually break from the motion of the robot (Figure 3.2a). In HCI and immersive environments, wires lead to restrictive, slow, and sometimes inaccurate solutions. For example, having a wired stylus restricts the users movement when operating the device, and using cameras for motion tracking involves significant processing latency (Figure 3.2b). In cars and aircraft, wires are one of the heaviest and most costly components. Also, it can be very hard to route the wires from the front to the back of the cabin because most of the room is occupied by passengers (Figure 3.2c).

There are already millions of wired industrial control systems in existence today. All of them use specialized hardware for their sensors and actuators and have custom control algorithms tuned to the application and network specifications, but the communication network that connects them is commodity (i.e. SERCOSIII). Therefore, any wireless network that

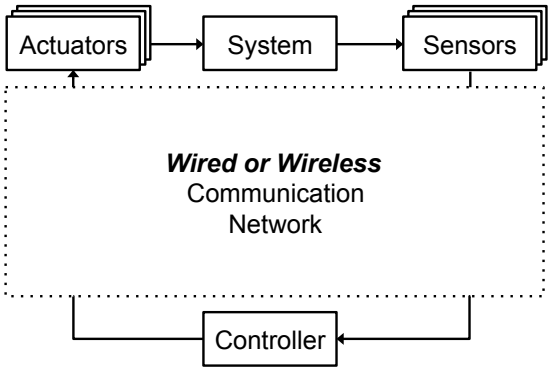


Figure 3.3: A drop-in wireless replacement for wired industrial control networks. The control system will be oblivious to what type of communication network is used as long as the it can meet the specifications.

would replace the wired network would have to have the same interface and meet the same specifications so that the control system is oblivious to the actual network implementation (Figure 3.3). Current wireless systems cannot act as such a drop-in replacement for wired industrial control networks since they have large deterministic and random overhead that scales with the number of nodes in the network, cannot efficiently cope with the impairments of the wireless channel, and have significantly degraded performance in the presence of interference. Therefore, new wireless system architectures are needed that are tailored to the requirements of industrial control systems.

The rest of this chapter will focus on centralized control systems because many industrial control systems are centralized or can be broken down into centralized subsystems. Also, they form the basic element of any general control system, so the techniques developed can be applied to a wide range of systems. In a hierarchical control system, the parent and children controllers can act as an additional node within the centralized control system network. In a distributed control system, each part of the system is an individual centralized or hierarchical control system.

3.1.3 Industrial Printer Example

Before developing new wireless system architectures for industrial control, the system must be modeled to understand its unique challenges and to analyze their performance. A large-scale industrial printer provides an excellent representative example of centralized, high-performance industrial control systems due to its strict latency and reliability requirements and the number of sensors and actuators in the system. It comes with a practically-deployed wired control protocol based on SERCOSIII that any proposed architecture can be measured against to see if it meets the requirements of a drop-in replacement.

The printer has approximately 30 moving printing heads that move at speeds up to 3m/s over distances up to 10m. The heads have sensors on board to measure velocity and other

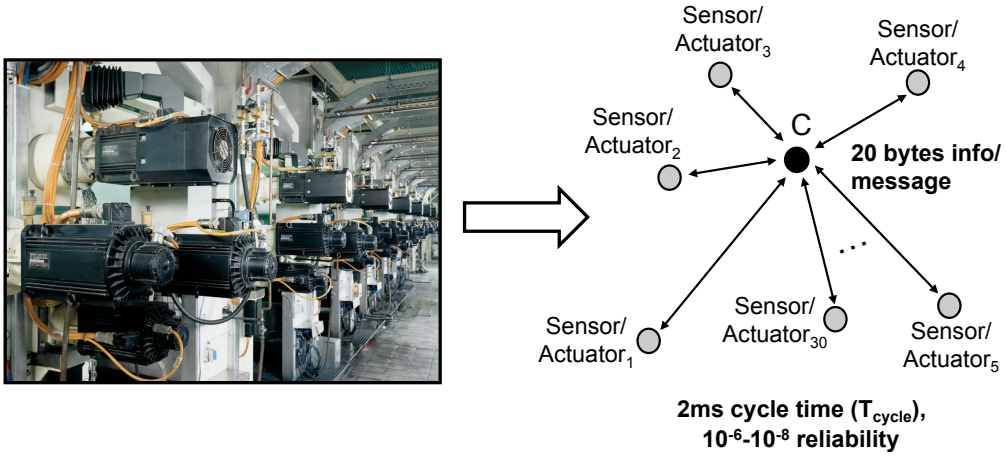


Figure 3.4: The industrial printer and its specifications.

state variables, and they have actuators that move them in 3-D space. Since the printer is a centralized control system, all messages either originate at the controller and are sent to the actuators (broadcast traffic) or originate at the sensors and are sent to the controller (convergecast traffic). The printer requires all sensors to transmit their observations and the controller to transmit instructions every 2ms, which is called a cycle. Each observation or instruction message has 20 bytes of information. For these specifications, the SERCOSIII protocol supports the printer's required cycle time of 2ms with a minimum error rate of 10^{-6} (although an error rate of 10^{-8} is preferred for better results and longer uptime). Note that the error rate is the probability of at least one transmission not being received successfully by the controller or the actuators (a system error rate), not BER or BLER.

3.1.4 High-Performance Industrial Control System Model

By studying the industrial printer's operation and specifications, a representative model for a centralized, high-performance control system, which is an atomic element of many classes of industrial control systems, is developed and summarized in Figure 3.5.

Since sensors and actuators only exchange information with the controller, the network has a logical star topology with a controller at the center and the sensors and actuators at the leaves. Leaf node to leaf node links are possible in the physical network topology, but in this case the leaves act as relays. Also, the topology of network may change as the channel conditions for the physical links change because links between nodes may come in or drop out.

As discussed in Chapter 3.1.3, the traffic pattern between the center and leaf nodes is periodic with period $t_{\text{cycle,req}} < t_{\text{crit}}$ and consists of two parts. First, the controller issues instructions to all actuators; second, each sensor transmits its observation to the controller. If at least one of these transmissions does not succeed within $t_{\text{cycle,req}}$, the system declares

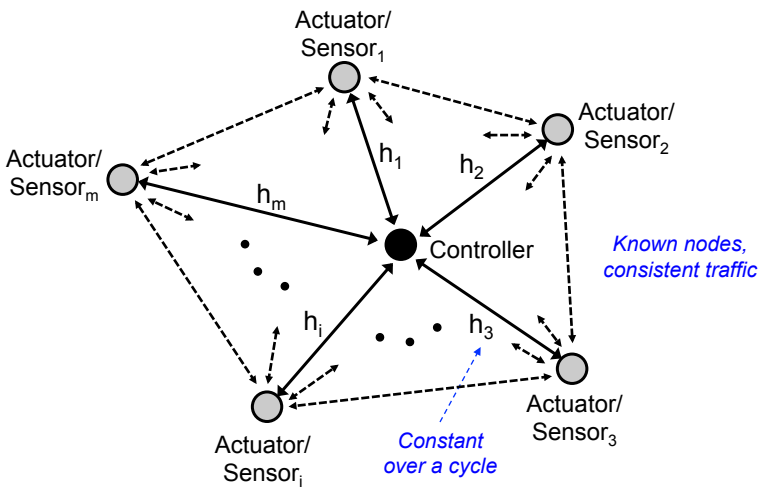


Figure 3.5: A model for industrial control systems, including slow fading channels and known network topology. The solid lines indicate logical and physical links in the network, and the dashed lines indicate possible physical links that may exist depending on the network topology and channel conditions.

an error. Since the latency for high-performance control is very small, $t_{\text{cycle,req}}$ will be less than the coherence time of the channel, and all physical links experience slow fading.

A few simplifying assumptions are used throughout the rest of the chapter. First, each leaf node has both a sensor and actuator, as is true with the industrial printer. This will not be required for the schemes developed to operate correctly, but individual sensors and actuators may need act as a full sensor/actuator node in the network. Second, the number of nodes in the network is known *a priori* since nodes do not join and leave a high-performance industrial control system (and it can be reset if this is necessary). Third, there is perfect error detection, which is reasonable given modern FEC and CRC codes, so convergence to a false codeword will never occur. Therefore, messages will be retransmitted until the correct information is received or $t_{\text{cycle,req}}$ is exceeded. Fourth, nodes are not energy-constrained (although their maximum transmit power is constrained) since the main concern is in developing schemes to see the best possible latency-reliability performance. Finally, all radios are TDD and half-duplex since this is how the majority of radios are built, especially in the ISM band.

One important note is that industrial control systems need a method to transfer non-control data. In the case of the printer, this could include the image information that each printer head uses to decide color to print in each location. The systems discussed in this chapter do not consider this aspect since a separate communication network operating in a different frequency channel can be implemented for this purpose or that data may be buffered ahead of time using the control system communication network. The main concern of this chapter is how to implement low-latency, high-reliability networks that handle communication of the control information.

3.1.5 Evaluating Performance

A communication system used for control cannot have delays larger than $t_{\text{cycle,req}}$ or allow any errors in transmitted data. However, communication systems can never guarantee this because of channel impairments, such as noise and fading. A tolerable probability of failure p must be defined, which should be selected small enough such that errors are not expected to occur during the system's lifetime. This is analogous to an allowable bit- or block-error rate in a traditional communication system. Assuming that failing to detect errors in the received data has probability much smaller than p (which is possible to accomplish with FEC and CRC), errors in the received data can be avoided by retransmitting the data until it is received correctly. The communication system fails if any one of the messages intended to be sent within the cycle is not received successfully during that cycle.

The performance of the system in a given channel is measured as the required SNR to achieve the required probability of error p within the cycle time constraint $t_{\text{cycle,req}}$. This gives a stochastic guarantee on the performance of the system, which equates to looking at a "worst-case" latency instead of the average or minimum latency.

3.2 Related Work

Low-latency, high-reliability wired communication networks for control systems have been successfully deployed for many years using SERCOSIII and other fieldbus standards. As discussed earlier, there is a strong desire to move to wireless networks because wired systems are costly to install, difficult to modify, and have many mechanical issues that cause system failure. Researchers looked at solutions on both the theoretical and practical sides, but a sufficient solution has not yet been developed.

The primary avenues of research on the theoretical side have involved modifying control algorithms to cope with the random latency introduced by communication systems, ranging from using a modified form of optimal control to using non-uniform or event-triggered sampling [17–20]. Many control algorithms are robust to errors from sensor noise or from an outside disturbance with an assumed distribution [52], but they are not necessarily robust to the random errors that occur from an error in the communication system.

On the implementation side, researchers have determined the performance of control systems using existing high data-rate wireless standards [22–25] and modified them at the MAC layer to increase performance for applications such as the smart grid, VoIP, and M2M communication [26–29]. Similarly, standards committees are currently putting much effort into adding features to LTE(-A) and IEEE 802.11ac/ad for low-latency operation with a large number of nodes. Based on the observations in Chapter 2.4, none of these efforts will achieve the specifications required for high-performance industrial control because of the large overhead and poor scaling properties underlying modern standards. If efforts are to succeed, the standards must be modified at both the MAC and PHY level.

Wireless sensor networks (WSN) have had much success in highly-reliable networking

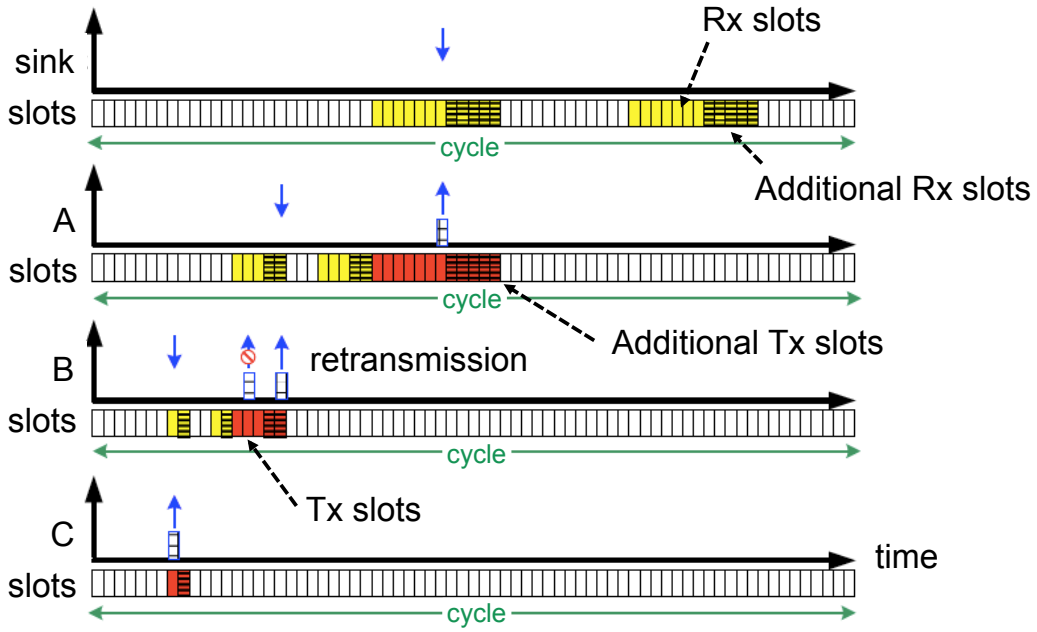


Figure 3.6: Operation of the GinMAC protocol labeled with Tx, Rx, and retransmission slots that are preallocated during an offline dimensioning procedure that determines how many slots are required to achieve the desired reliability [30].

because they can route around obstacles and weak links and use channel hopping to overcome frequency selective fades. WirelessHART and ZigBee are popular WSN standards that have been adapted for high-latency industrial control applications [55]. However, just as with high-data-rate standards, the WSN standards cannot be used for high-performance industrial control due to their low data rates, slow routing, and high overhead (Chapter 2.4). Again, any efforts to use the WSN ideas must modify both the MAC and PHY.

The wireless sensor and actor network (WSAN) community has developed numerous protocols that ensure timely and reliable data transport in WSANs, among which Burst and GinMAC are the best examples [30]. They use offline dimensioning and preallocated transmission time slots to obtain a guaranteed latency bound (Figure 3.6). Unfortunately, they share many of the same weaknesses as WSN protocols because they rely on time and multi-hop diversity to achieve reliability. They also rely on multiple rounds of ARQ to achieve reliability. Waiting for ACKs has a large latency overhead, and, if the channel is in a slow fade, it is difficult to recover using only retransmissions and not modifying other system parameters. Also, the WSAN solutions only target the MAC layer, instead of co-optimizing with the PHY layer.

Other attempts have been made to lower latency and increase reliability that are not specifically targeted at industrial control. The best examples are spatial diversity techniques such as MIMO, distributed MIMO, and relaying. In particular, the development of dis-

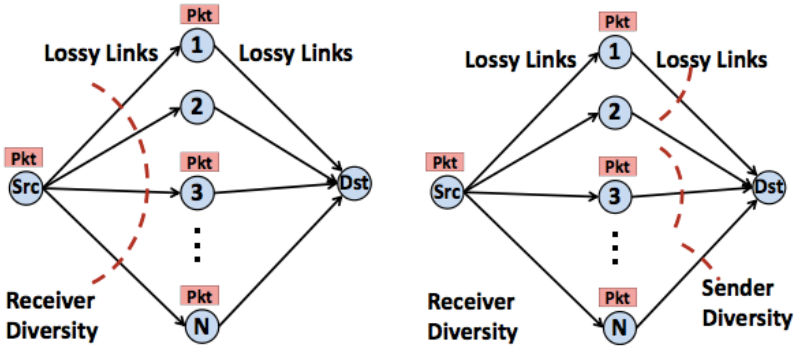


Figure 3.7: The two phases of the SourceSync protocol: (a) transmission from the source to the relays (receive diversity) and (b) simultaneous transmissions to the destination from the relays that successfully decoded the source's message using a distributed space-time code (transmit diversity) [59].

tributed space-time codes that enable multiple users to simultaneously transmit the same information is a promising method to increase diversity without increasing latency proportionately to the number of relays [56–58]. SourceSync is an example of a protocol that achieves this for a logical point-to-point link [59] (Figure 3.7).

3.3 Initial System Architecture

Most high-performance wireless standards, particularly IEEE 802.11ac/ad and LTE, have increased latency and decreased reliability as additional users are added to the system due to the complications of scheduling medium access, handling retransmissions, and mitigating interference and collisions with many users. Taken as they are now, the current systems based on these standards cannot meet the latency and reliability specifications required by industrial control, but many optimizations can be made to improve performance. This section identifies the optimizations that can be made at the MAC and PHY levels, develops an initial architecture based on those optimizations, and reveals the strong dependence of the architecture's achievable latency on the available diversity.

3.3.1 Assumptions

Consider a system that has both a logical *and* physical star topology instead of the general network model in Figure 3.5. The underlying inter-sensor/actuator links present

Some or all of the work in Chapter 3.3 appeared in [32] and benefited from extensive discussions with Milos Jorgovanović.

in the general network or additional nodes that are added to the system to act as part of a distributed antenna array can be captured by increasing the diversity on the controller-sensor/actuator links. Although this abstracts how the diversity is obtained, it simplifies the design and analysis of the system architecture so that we can focus on initial optimizations and gain more insight into the architecture's behavior and the major factors affecting its performance.

3.3.2 Overview of Optimizations

The main focus of the initial system architecture is identifying high-level optimizations that can be made to existing standards and proposed systems, such as those listed in Chapter 2.3 and Chapter 3.2, quantifying the reduction in latency, and finding the limiting factors to the achievable latency.

First, since a wireless control system only needs a very basic interaction with other wireless control systems to coordinate resource usage, the network layer can be greatly simplified or combined with the MAC layer and the other higher level layers can be eliminated. This reduces the complexity of the system, keeps as much of the system in hardware as possible, and reduces the amount of overhead in the transmitted data because headers from the higher layers are eliminated.

This leaves only the MAC and PHY layers, which have the following key optimizations compared to current systems described in Chapter 2.3 and Chapter 3.2 that mitigate the overhead for having frequent, short transmissions from many nodes:

- Fixed resource schedule only for initial transmissions
- Controller broadcasts initial data and ACK
- Sensors/actuators combine their initial data and ACK
- Fixed retransmission durations
- Optimized, fixed coding rates

The specific system described below is just one of many possible implementations that embody the above points.

3.3.3 MAC Layer

The MAC is defined by its operation over one cycle because the data transmission of the control system is periodic. As shown in Figure 3.8, the protocol has five phases in every cycle: (1) controller broadcasts instructions to actuators, (2) sensors individually transmit observations and actuators send ACKs (in the same message), (3) controller acknowledges all sensor transmissions, (4) controller retransmits to actuators that did not receive their instructions, and (5) sensors that did not succeed retransmit to the controller. Each phase

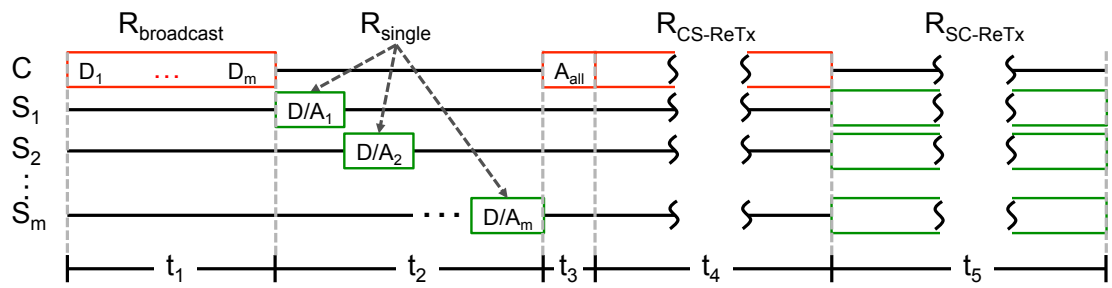


Figure 3.8: Timing diagram of the initial system architecture with no frequency multiplexing.

starts at a fixed time within the cycle, so synchronization is necessary for all nodes to have a global sense of time. This consumes extra communication resources, but it is preferable to distributing a schedule each cycle since a node in a deep fade cannot receive its schedule.

In phases 1 through 3, each node has an assigned time set during system initialization to send instructions or observations. The controller starts by broadcasting all of the instructions to the actuators in one codeword. This increases the blocklength of the actuators' data, which decreases the probability of decoding error. It also packs the data into the minimum number of OFDM symbols, which minimizes the overhead from the cyclic prefixes. Next, in a predetermined order each sensor sends its observations as well as an acknowledgment indicating whether it received the controller's data. If a node does not have an actuator, it can send a fixed value for the ACK; if it does not have a sensor, it can send only the ACK. The controller then broadcasts a block ACK to all of the sensors indicating which ones need to retransmit. The length of the first phase, $t_1 + t_2 + t_3$, is set by the total time needed to transmit the data and ACKs once from each node.

During phase 4, the controller retransmits data to any actuators that responded with a NAK or whose transmission the controller could not decode. Since only the controller is transmitting, it does not have to worry about collisions. It can simply repeat a single actuator's data in one codeword a given number of times based on the SNR to that actuator. The actuator can use an ARQ or HARQ approach to decoding. Note that there are no ACKs since it would waste time waiting for an ACK that would most likely not be received due to the poor channel (the channel is poor since the first transmission was not received). The length of the second phase, t_4, is fixed during system initialization and is equal to the time needed for the minimum number of controller to actuator retransmissions to guarantee the specified reliability. This allows the next phase to have a fixed start time.

In phase 5, the sensors retransmit data to the controller if they received a NAK in the block ACK or could not decode it. Since possibly multiple sensors may be retransmitting, they can be preallocated time-frequency resources. Alternatively, they can use CDMA, which allows the controller to increase the effective SNR of the received signal since the data is essentially repeated many times. Again, there are no ACKs since they would waste resources. The length of the fifth phase, t_5, is fixed during system initialization and can be calculated

in the same way as for t_4. Therefore, the entire cycle has a deterministic length. Note that if t_4 and t_5 were calculated together, the total cycle time would be smaller. However, this would make the border between the fourth and fifth phases random, which would be difficult to implement.

3.3.4 PHY Layer

This architecture uses OFDMA in the initial transmission phase to allow multiple sensors to be scheduled at the same time over different frequency subcarriers. Alternatively, OFDM can be used for a simpler time-multiplexed implementation, which is shown in Figure 3.8. Either OFDMA or CDMA can be used in the retransmission phase depending on the implementation chosen. Note that during retransmissions the SNR to the remaining nodes is small, so it is best to use a scheme that allows maximum ratio combining of retransmissions to boost the effective SNR. Similar to LTE, the controller broadcasts reference and synchronization signals to all sensor/actuator nodes. The synchronization and channel estimation will need to be more accurate than in LTE because of the fixed schedule, so longer sequences are needed. This requires additional time-frequency resources, which lowers the data rate. This overhead does not scale with the number of nodes since it is broadcast.

Coding is one of the most powerful tools to reduce the number of retransmissions, but it does this at the expense of adding deterministic overhead. Current wireless systems rely on retransmissions to clean up any errors that occur due to using an excessively high code rate or constellation order. Control systems cannot tolerate the number of retransmissions required for this, so they must be able to use lower rate codes with low-order constellations. The initial architecture uses the optimal code rates for the broadcast frame and individual frames to balance the deterministic overhead and the number of retransmissions. Note that this affects the choice of retransmission scheme chosen in the MAC layer, so the code rate and retransmission scheme should be chosen jointly. The code rate for the individual frame is usually much lower than 1/3. This optimization differs from that in [60] and [61] because the channel realization for the original transmission and for the retransmission are the same since the latency constraint is smaller than the coherence time.

The code rates are fixed during system initialization to avoid the communication needed to adapt the code rate to the channel and because having different code rates would make having a fixed schedule impossible during the initial transmission phase. Having fixed code rates works well if the average SNRs of the nodes are equal, which can be accomplished through CDMA-like power control feedback appended to the data. Since the same system may be used in different conditions, the hardware needs to be flexible so that any rate can be chosen during initialization. Since the information length is constant, two options that are promising are rateless codes and low-rate punctured codes. Similarly, a fixed constellation is used.

Diversity must be extracted in any way possible because reliability is of the utmost importance. Since the latencies are on the order of milliseconds and the coherence time for carrier frequencies and velocities of interest are on the same order or larger, time diversity,

achieved through interleaving in time, cannot be used. However, both frequency diversity from sending data across subcarriers separated by more than the coherence bandwidth and spatial diversity from multiple antennas are available to the system. Because the problem for control systems is meeting the reliability requirement, antennas should be used for diversity over multiplexing.

3.3.5 Evaluation Methodology

To evaluate the communication network as defined in Chapter 3.1.5, the minimum cycle time at a reliability of p, $t^*_{\text{cycle}}(p)$, can be found for a range of SNRs, and the smallest SNR that results in a $t^*_{\text{cycle}}(p) \leq t_{\text{cycle,req}}$ is the system's performance. Also, $t^*_{\text{cycle}}(p)$ is an interesting quantity on its own because, when evaluated across multiple system parameters, it shows how the system's latency scales with changing channel conditions, number of nodes, and reliability requirements.

Below are the steps to find $t^*_{\text{cycle}}(p)$ for a single SNR, each of which are described in detail in the subsequent sections (along with how it is implemented for the industrial printer with the initial system architecture):

1. Define the operating conditions and system parameters.
2. Define the protocol layers of the system architecture.
3. Write the equation for the architecture's cycle time.
4. Calculate distributions for the random terms in the cycle time equation.
5. Calculate $t^*_{\text{cycle}}(p)$.
6. Rerun the procedure to optimize the free variables.

To find a point that meets the condition $t^*_{\text{cycle}}(p) \leq t_{\text{cycle,req}}$, this procedure is run for multiple SNRs, which can be chosen according to any desired search strategy.

Define the operating conditions and system parameters

Define the SNR, channel type, the number of sensors and actuators in the system, the diversity on each link (which can be arbitrarily defined to see its effect on the system or calculated based on the number of antennas and the frequency selectivity of the channel), the amount of data each node needs to send and receive, the code rates, and the available bandwidth. All of these parameters, except the SNR and coding, are part of the system specifications for a given application and are known ahead of time. Here, the channel is assumed to be Rayleigh, p is 10^{-6}, there are 30 sensors and 30 actuators that each send/receive 20 bytes of information, and the system has 20MHz bandwidth available. For this analysis, the SNR, level of diversity on each link, and code rates are fixed. The analysis can be repeated

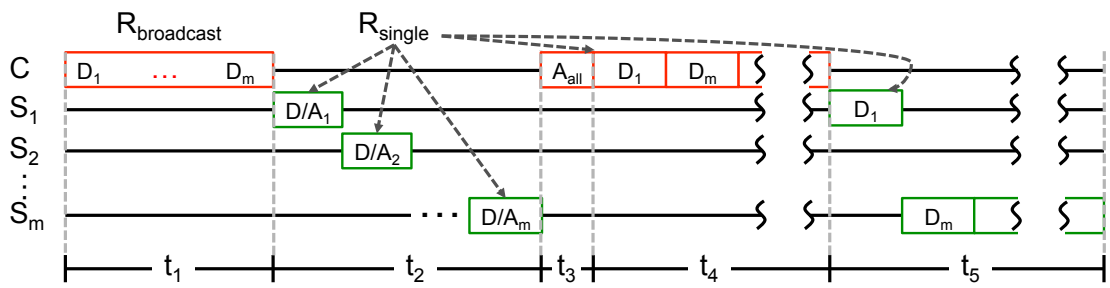


Figure 3.9: Timing diagram of the evaluated initial system architecture.

for other combinations of parameters in order to explore the design space or to optimize the system.

Define the protocol layers of the system architecture

In the case of the initial architecture, only the PHY and MAC layers are used. Essentially, all details that relate to the timing and duration of transmissions and idle periods must be well-defined. For this example, only time-division duplexing will be used. Also, an ideal TDMA-based preallocated transmission slot scheme is used for the controller to actuator and sensor to controller retransmission phases. It is ideal because the controller knows exactly how many retransmissions each actuator requires to receive the instructions correctly, and each sensor knows exactly how many retransmissions the controller requires to receive the observations correctly *and* knows when to transmits so as not to collide with each other. Also, the system does not use HARQ techniques to increase the effective SNR with each retransmission; instead ARQ is used, which has higher latency (since deep fades cannot be "overpowered") but is much simpler to implement and analyze. The timing diagram for the analyzed scheme is shown in Figure 3.9.

Write the equation for the architecture's cycle time

Based on Figure 3.8 and by using the ideal slotted TDMA with preallocated retransmission slot scheme, the cycle time for the architecture (here, a random variable) is:

$$\begin{aligned} T_{\text{cycle}} &= t_{data} \cdot ((m \cdot d_s + d_{CRC})/R_b + t_{CP}) \\ &\quad + t_{data} \cdot ((d_s + d_{CRC} + 1)/R_i + t_{CP}) \cdot m \\ &\quad + t_{data} \cdot ((d_a + d_{CRC})/R_i + t_{CP}) \\ &\quad + t_{data} \cdot ((d_s + d_{CRC})/R_i + t_{CP}) \cdot (N_c + N_s) \end{aligned} \tag{3.1}$$

where d_s is the number of bits of information each node transmits, d_{CRC} is the number of CRC bits, d_a is the number of bits in the block ACK, R_b is the broadcast code rate, R_i is the individual packet code rate, t_{CP} is the cyclic prefix length, N_c is the number of

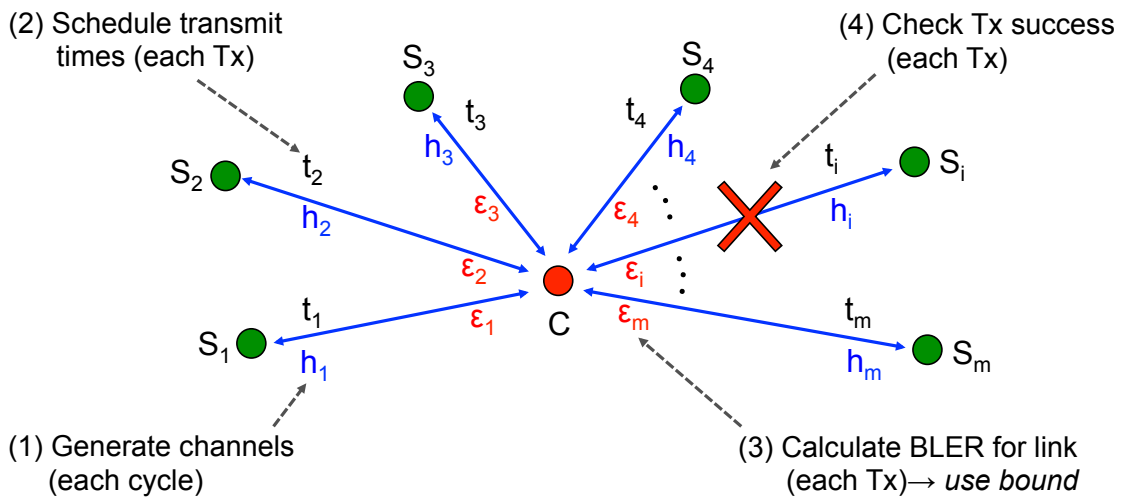


Figure 3.10: The network simulator used to evaluate the idealized initial system architecture. Since the network repeats every cycle, the simulator calculates the cycle time required for every cycle by first drawing the fading coefficients from the desired distribution that will stay constant for the cycle. Then, for every transmission, a transmit time is selected; the BLER is calculated based on the fading coefficient, the number of repetitions of the packet, and Equation (3.2); and a weighted coin is flipped (weighted based on the BLER) to check if the transmission is a success.

controller retransmissions, and N_s is the total number of sensor retransmissions. The first line corresponds to the controller sending the broadcast frame in phase 1, the second corresponds to the m sensor nodes sending their data plus ACKs in phase 2, the third corresponds to the controller sending the block ACK in phase 3, and the fourth corresponds to the controller and sensor TDMA retransmissions in phases 4 and 5.

Calculate distributions for the random terms in the cycle time equation

The distributions for the random terms in Equation (3.1) must be modeled, which in this case are N_c and N_s. This can be done analytically if possible or by network simulation, which is used here. The simulation performs iterations of generating channel realizations for each controller-sensor/actuator pair based on the parameters from step 1, models the links as switches where the probability of the switch being closed is the BLER, and then sends codewords across each link at the capacity of the link until it succeeds. In each iteration, the value of N_c and N_s is recorded, and those values are used at the end of the simulation to calculate an empirical joint probability mass function (PMF) for N_c and N_s. Figure 3.3.5 summarizes the network simulation procedure.

A key step in the simulation is calculating the BLER. Since the codes have shorter blocklengths, the effects of non-ideal codes must be considered. The BLER can be simulated

for every code under consideration, but this can be prohibitively slow if the behavior at low error rates is required and many codes are under consideration. Fortunately, there exist bounds on the performance of codes with finite blocklengths [31], and those bounds can yield the BLER over an arbitrary channel when rearranged into the following form:

$$\epsilon = Q\left(\left(C - R + \frac{1}{2}\frac{\log\left(k/R\right)}{(k/R)}\right)\sqrt{\frac{k}{RV}}\right) \tag{3.2}$$

where ϵ is the BLER, Q is the tail probability of the standard normal distribution, C is the capacity of the channel, R is the code rate (either R_i or R_b), k is the information length, and V is the dispersion of the channel. Both C and V are known for K parallel AWGN channels and are only a function of SNR [62], and the BLER can be found for fading channels by averaging over the fading statistics for a given noise variance. Note that if HARQ is used instead of ARQ, the SNR of the link will increase with each retransmission, which reduces the BLER and the number of retransmissions.

Calculate $t^*_{\mathbf{cycle}}(p)$

Using the distributions of the random variables, the PDF/PMF of T_{cycle} can be calculated. Then, $\overline{F}(T_{\text{cycle}})$ can be calculated empirically from the PDF/PMF of T_{cycle}, which is used as a lookup table to find $t^*_{\text{cycle}}(p)$ using p as the lookup argument (Figure 3.11). Alternatively, the same procedure can be performed on the controller and sensor/actuator retransmission phases separately to get their individual $t_{\text{phase}}(p)$ times (analogous to $t^*_{\text{cycle}}(p)$, but for phases). In other words, the PDF/PMF for a phase i is used to calculate the phase's $t^*_{\text{phase},i}(p)$, and $t^*_{\text{cycle}}(p) = \sum_i t^*_{\text{phase},i}(p)$. This allows each phase to have its own fixed duration, although the cycle time will be longer than if the overall PDF/PMF of T_{cycle} is used. The latter approach is used here so that all phases have fixed start and end times, particularly phases 4 and 5 since they are the only phase with random components.

Rerun the procedure to optimize free variables

This procedure can be rerun for different system parameters, such as R_b and R_i, to find the minimum possible value of $t^*_{\text{cycle}}(p)$ for the fixed parameters, which results in the optimal architecture.

3.3.6 Results for the Industrial Printer Example

The methodology for evaluating the initial architecture was implemented using Matlab. The finite blocklength BLER bound given by Equation (3.2) was used because the main concern was validating and optimizing the architecture. The bounds provide an effective tool to explore the effects of different levels of diversity and different code rates on the performance of the system. In fact, for each level of diversity, the simulation is run for a

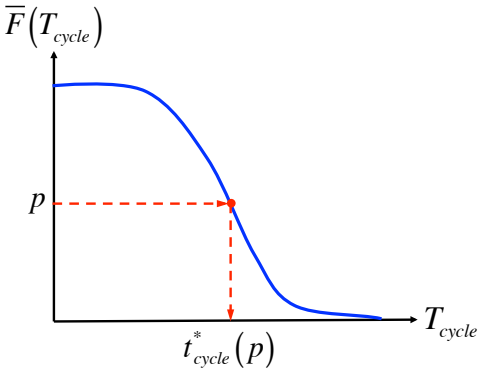


Figure 3.11: Using the complementary PDF/PMF of the overall cycle time distribution to find the overall $t^*_{\text{cycle}}(p)$. The same procedure can be done individually for each phase i. In the latter case, he overall $t^*_{\text{cycle}}(p)$ is the sum of the individual phases' $t^*_{\text{phase},i}(p)$.

set of parameters with many different pairs of broadcast and individual packet code rates to find the minimum $t^*_{\text{cycle}}(10^{-6})$ of the system under those parameters.

After running the methodology for a Rayleigh fading channel with diversity l (where l is fixed for each run and changed between runs) and $p = 10^{-6}$, it is found that $l \geq 4$ results in practical operating SNR on the links between nodes. If p is decreased, the diversity would need to be increased to keep the operating SNR in a practical range.

The resulting $t^*_{\text{cycle}}(10^{-6})$ as a function of SNR with $l = 4$ is shown in Figure 3.12 along with the optimal code rates for the lowest and highest SNRs. $t^*_{\text{cycle}}(10^{-6})$ is fairly insensitive to the exact value of the code rate pairs, so the simulation grid of the pairs is somewhat coarse. This causes the steps observed in the $t^*_{\text{cycle}}(10^{-6})$ curve. The latency specifications of the 30 node printer are met at SNRs above 7dB with a code rate of 0.6 for the broadcast codeword and 0.2 for the individual codewords. At lower SNRs, the required code rate decreases in order to reduce the number of retransmissions. Since Equation (3.2) was used to model the codes, the actual minimum SNR will be several dB larger than 7dB.

The minimum latency is limited by the time to transmit the controller data, the sensor data and ACKs, and the controller block ACK in the initial phase. The latency increases sharply for SNRs below 0dB due to one node being stuck in a deep fade, which occurs because the probability of failure being considered at is 10^{-6}, which is fairly small. Figure 3.12 also shows the $t^*_{\text{cycle}}(10^{-6})$ curves for a printer system with 100 and 500 sensors and actuators. Their $t^*_{\text{cycle}}(10^{-6})$ at high SNR is larger because the deterministic amount of data to transmit increases linearly with the number of nodes. The $t^*_{\text{cycle}}(10^{-6})$ goes to infinity sooner for a larger number of nodes because the probability of one node being in a deep fade increases. Figure 3.13 provides another view of the system's performance as the number of nodes increases at low, medium, and high SNR.

The key takeaway from this analysis is that the minimum latency achievable with the system is difficult to reduce since the architecture's optimizations on the MAC and PHY level

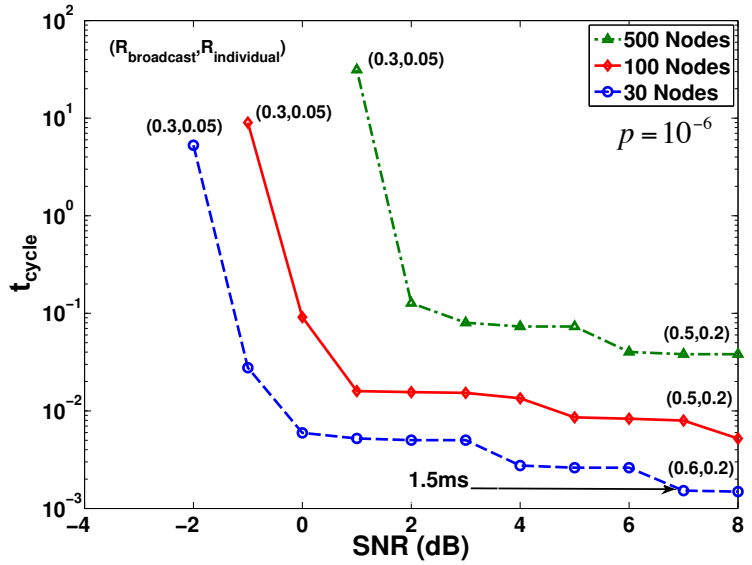


Figure 3.12: The initial architecture's $t^*_{\text{cycle}}(10^{-6})$ versus SNR for an industrial printer with 30, 100, and 500 sensor/actuator nodes.

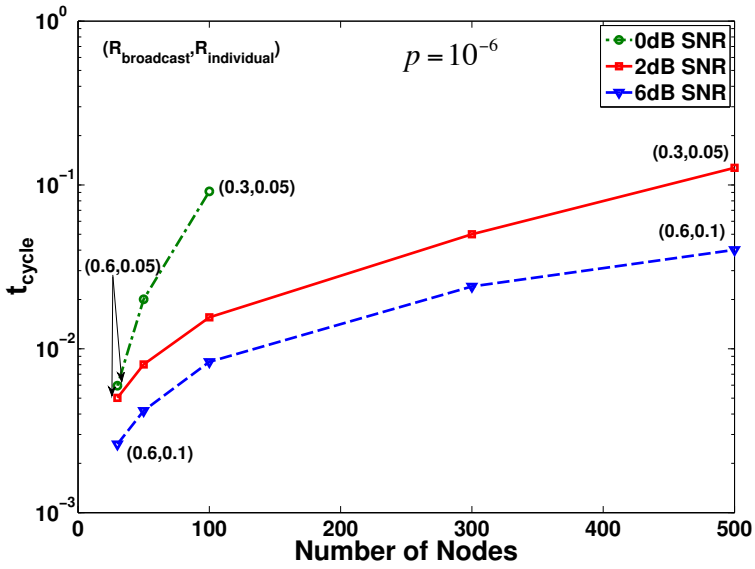


Figure 3.13: The initial architecture's $t^*_{\text{cycle}}(10^{-6})$ versus the number of nodes at 0dB, 2dB, and 6dB SNR.

have done that already. However, the level of diversity is the key to lowering the $t^*_{\text{cycle}}(p)$ of the system for a given p, number of nodes, and operating SNR (or alternatively the minimum required SNR for a given number of nodes, p, and $t^*_{\text{cycle}}(p)$). If HARQ is used to increase the effective SNR with each retransmission, the $t^*_{\text{cycle}}(p)$ would decrease, but the performance

would still be limited by the diversity since the probability for a deep fade decays much more quickly with increasing diversity than increasing SNR.

To achieve even lower $t^*_{\text{cycle}}(p)$ for systems with more nodes and stricter reliability specifications and tighter power constraints, more diversity is needed. Unfortunately, using multiple antennas on each node or transmitting data across a large bandwidth many not provide sufficient diversity because it relies heavily on the specific channel realizations. Instead, alternative schemes must be developed that can generate diversity in a more robust manner.

3.4 System Architectures Targeting Increased Diversity

The fundamental issue with the initial system architecture is that any one node being in deep fade results in a large increase in system latency or even failure, so diversity is required to lower the probability of a node being in deep fade. The amount of diversity provided by techniques based on channel variation across frequency and across antennas depends on the environment, which may not be enough because high diversity is required to meet the latency and reliability specifications. Cooperation either between multiple access points connected to the same controller or between the many sensor/actuator nodes in the system can solve this issue because diversity can be generated no matter the environment. This section introduces multi-user diversity, discusses a coordinated multipoint-based system architecture, develops a cooperative relaying system architecture and models for frequency diversity-based systems [63], and evaluates the cooperative relaying system and the frequency diversity-based systems based on a simplified link level analysis.

3.4.1 Multi-User Diversity

When there are multiple spatially-distributed antennas in the system, cooperative and multi-user diversity can be harvested. Many researchers have studied these techniques in great detail, so the treatment here is limited. Laneman et al. [64] showed that cooperation amongst distributed antennas can provide full diversity without the need for physical arrays. [65] showed that even with a noisy inter-user channel, multi-user cooperation increases capacity and leads to achievable rates that are robust to channel variations. The prior work in cooperative communication tends to focus on the asymptotic regimes of high SNR and

Some or all of the work in Chapter 3.4 appeared in [33] and was done in collaboration with Vasuki Swamy, Paul Rigge, Gireeja Ranade, and Sahaana Suri. My contributions were primarily in the form of contributing to discussions leading to the development of the cooperative relaying architecture and to the diversity meter analysis and results. The other authors developed the cooperative relaying architecture, formulated the comparison schemes and robustness arguments, performed their analyses, and interpreted the results. The writing and making of figures was done primarily by the other authors, although I wrote the diversity meter section and helped edit the manuscript.

rely highly on the accuracy of the fading distribution in its analysis. By contrast, industrial control communication systems should operate in moderate SNR regimes and should not rely on the accuracy of fading models. The study in [66] found that reliable spectrum sensing by cognitive radios need not depend strongly on the details of fading distributions. It could instead rely upon the independence of fades across users. This is an important point that will be revisited later because it holds for the proposed cooperative relaying protocol.

Multi-antenna techniques have been widely implemented in commercial wireless protocols like IEEE 802.11ac. [67, 68] use relays and a TDMA-based scheme to bring sender-diversity techniques to industrial control. Unfortunately, TDMA can scale poorly with network size. To scale better with network size, simultaneous transmissions by many relays can be used instead, which requires distributed space-time codes such as those in [56–58], so that each receiver can harvest a large diversity gain. Recent work by [59] demonstrates the implementation of such space-time codes that harvest sender diversity using concurrent transmissions is possible.

3.4.2 Coordinated Multipoint System Architecture

In order to increase diversity in the system without mounting more antennas on each device or relying on frequency variations in each channel, additional "access points" wired to the central controller can be distributed around the system, which is analogous to coordinated multipoint (CoMP) in LTE-A. Each access point acts as a different user and adds additional channels to the controller.

In the context of wireless control systems, CoMP would mainly be used for joint processing on both the downlink and the uplink. For the downlink, this involves distributing the transmit data to all access points and jointly transmitting the data to the actuator(s). This provides the maximum diversity gain, but also requires the most data transfer and tightest synchronization. A simpler version of joint processing simply chooses the best access point to transmit to the actuator(s). For the uplink, joint processing involves the sensor transmitting and all access points receiving and combining their information for a final decision. This involves larger data transfers due to sharing soft information with tight latency constraints. Like the downlink, the simpler version of joint processing chooses the result from the access point with the most reliable received signal. These technique have been well-studied and analyzed in the literature, and they are currently under development for LTE Release 11 [69, 70].

Instructions for actuators cannot be distributed ahead of time to all access points since it is generated as it is needed, and observations from sensors cannot wait to be decoded due to the cycle time constraint. Since CoMP requires all access points to share transmit and receive information, the instructions must be shared between each base station before transmitting to the actuators. Likewise, the observations must be shared in order to jointly decode or just decide which result to use. Any time for sharing between the base stations reduces the time for transmissions between the controller and sensors/actuators. To minimize this overhead, the access points require high-bandwidth backhaul that has low-latency and high-reliability.

Likely, the backhaul will be implemented with wires to meet the tight CoMP specifications (tighter than those of the control system) and to avoid interfering with the control system.

In applications such as automobiles, aircraft, and HCI, having wired components is the exact problem that the wireless control system was meant to avoid, but it may be a viable option for industrial robotics. If this method is chosen, the initial system architecture described in Chapter 3.3 can be used as is.

3.4.3 Cooperative Relaying System Architecture

Instead of generating multi-user diversity by using multiple access points that are all connected to the central controller (C), cooperative relaying can be used to turn each of the n sensor/actuator nodes (S) into a potential relay. Relays transmit simultaneously in order to minimize latency. To accommodate this, the initial system architecture from Chapter 3.3 must be modified at both the MAC and PHY layer, although many of the key features remain, such as the scheduled transmissions, controller broadcast, controller block ACK, and the combined sensor data and ACKs.

Overview

Recall that communication is periodic with period t_{cycle}. Each period, or cycle, C sends a b-bit message to every S, and every S sends a b-bit message to C. Following cellular convention, the downlink (DL) consists of controller transmissions to sensors/actuators, and the uplink (UL) consists of sensor/actuator transmissions to the controller. It is assumed that the controller and all nodes are normally in-range of each other, but deep fading events can cause transmissions to fail.

The proposed cooperative relaying system uses idle S nodes as relays to overcome deep fades. On the DL, nodes that have received messages for other nodes from the controller act as *simultaneous* relays to deliver messages to their destination in a multi-hop fashion. A similar idea is applied on the UL. When nodes are not transmitting, they are listening to all other transmissions. Those that have successfully decoded other nodes' messages act as simultaneous relays for those messages. The protocol is implemented by dividing every communication cycle into multiple DL and UL phases with an additional short (but critical) scheduling and acknowledgment phase. First, the initial DL and UL phases occur, which are followed by the scheduling phase. Next, the additional DL phases occur that also transmit the schedule to all nodes. Finally, the remaining UL phases occur. The ordering of the DL and UL phases is critical for low-latency operation and UL performance. Figure 3.14 shows a timing diagram for the protocol with 3 DL and UL phases, and Figure 3.15 shows a more detailed example of the protocol's operation. Each part of the protocol will be discussed in more detail in the following sections.

Simultaneous transmissions are possible using distributed space time codes, such as cyclic delay diversity (CDD). In CDD, users with the same data can transmit at the same time as long as they use different cyclic shifts of their OFDM symbols with appropriately updated

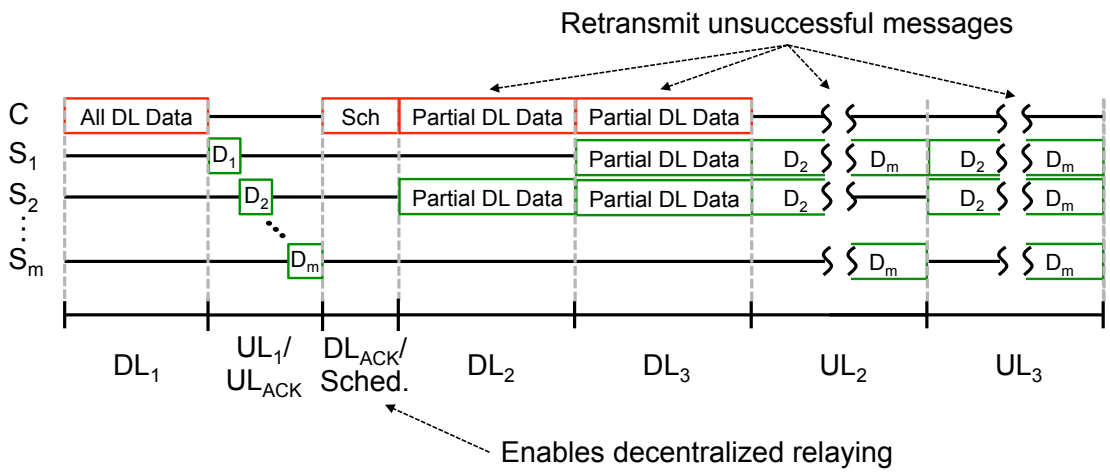


Figure 3.14: Timing diagram of the proposed cooperative relaying protocol. This differs from the timing diagram for the initial system architecture since the DL and UL retransmission phases are broken up into multiple parts to enable relaying.

cyclic prefixes (Figure 3.16). Since time delays are phase shifts in the frequency domain, the receiver sees the sum of each terminal's transmissions in the same way as multipath, which causes fading and makes the channel frequency selective. Essentially, CDD converts spatial/antenna diversity into frequency diversity [58]. Other space-time schemes that enable simultaneous transmissions are possible.

Resource Assumptions

A few assumptions are made regarding the hardware and environment to focus on the conceptual framework of the protocol. All the nodes share a universal addressing scheme and order, and messages contain their destination address.

Fundamentally, errors are caused by bad fades. Since the short cycle time puts the system in the non-ergodic flat-fading regime, time diversity cannot be used. All nodes are assumed to be capable of instantly decoding variable-rate transmissions [71]. 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 [72]. Thus time slots can be scheduled for specific nodes without any overhead. The protocol relies on time/frequency synchronization to achieve simultaneous retransmission of messages by multiple relays. Also, if k relays simultaneously transmit (such as with consciously introduced jitter to transform spatial diversity into frequency-diversity as in CDD [58]), then all receivers can extract signal diversity k.

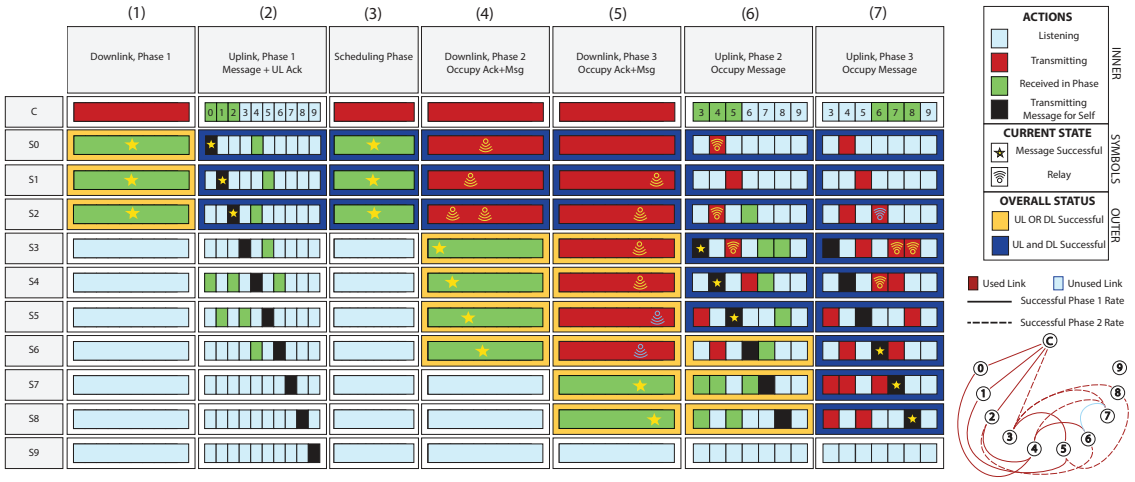


Figure 3.15: A representative example of the proposed cooperative relaying protocol when there are three DL and UL phases each. The table shows a variety of successful DL and DL transmissions using 0, 1 or 2 relays. S9 is unsuccessful for both DL and UL. The graph on the right shows the underlying link-strengths for the network.

System Architecture Description

The cooperative relaying system architecture consists of $2N + 1$ phases, where there are N used for the DL, N used for the UL, and one used for scheduling. The ordering a short description of the phases are given below, and detailed descriptions are given in the subsequent sections:

1. DL phase 1: initial broadcast of actuator information from the controller.
2. UL phase 1: initial single sensor data and ACK transmissions to the controller.
3. Scheduling phase: initial broadcast of block-ACK/schedule from the controller.
4. DL phases 2 to N: controller and relays rebroadcast remaining actuator information and schedule.
5. UL phases 2 to N: sensors and relays rebroadcast remaining sensor information.

DL and UL Phase 1 DL phase 1 (length T_{D_1}) is used by the controller to broadcast all messages to all sensor/actuator nodes at rate

$$R_{D1} = \frac{b \cdot n}{T_{D1}}$$

This corresponds to the first column in Figure 3.15, where nodes S0-S2 successfully receive the broadcast DL packet.

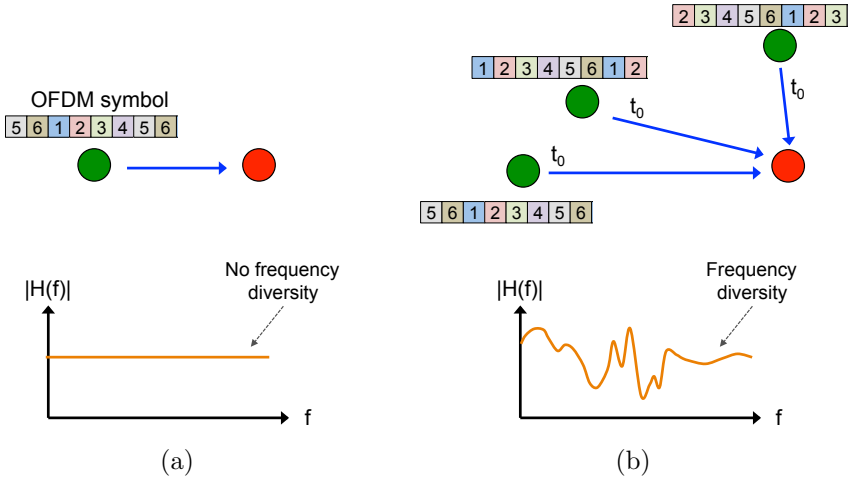


Figure 3.16: The difference between (a) a standard point-to-point OFDM transmission and (b) a simultaneous OFDM transmission from multiple users using cyclic delay diversity (the OFDM symbols are cyclic shifts of the original data).

UL Phase 1 (length T_{U1}) follows, in which the individual nodes transmit their messages (including one bit for an ACK/NAK for the DL message) to the controller one by one according to a predetermined schedule set at system initialization. Each transmission evenly divides the time slots among all sensor/actuator nodes, giving a common rate of

$$R_{U1} = \frac{b+1}{T_{U1}/n} = \frac{(b+1)\cdot n}{T_{U1}}$$

This corresponds to column 2 of Figure 3.15, where nodes S0-S2 successfully transmit their UL packets to the controller. Since all idle nodes are listening, S2 and S0 receive and decode the message from S4, even though it does not reach the controller.

Sensor/actuator nodes successful in both the first phases of the DL and UL are called strong users; otherwise they are weak users.

Scheduling Information The scheduling phase (length T_S) is used by the controller to transmit a block ACK containing the status of messages for all users, which is received by the strong users. Each sensor/actuator node adds 2 bits of information in the block ACK, one for the DL status and one for the UL status for that node. Sharing this information about the system's state enables the controller and other nodes to create a common schedule for relaying messages for the weak nodes. The strong nodes that are able to help must receive the block ACK so that they can relay in the subsequent DL phases. Although the weak nodes nodes have not received the block ACK yet, it does not matter since they do not have anything to transmit until they receive the DL information. The block ACK is passed on to the remaining nodes in the subsequent DL phases along with the DL data.

The scheduling phase corresponds to column 3 of Figure 3.15, where the strong nodes S0, S1, and S2 receive the block ACK.

DL Phases 2 Through N In DL phase 2 (length T_{D2}) the controller alters its broadcast message to remove already successful messages for the strong nodes; so, the packet is sent at an adapted rate

$$R_{D,2} = \frac{x_{D1} \cdot (b+2)}{T_{D2}}$$

where x_D is the number of nodes that were not successful in DL phase 1. The strong users compute the same adapted packet and broadcast the same information as the controller *simultaneously*. This procedure is illustrated in column 4 of Figure 3.15. S3 receives the broadcast message and block ACK from the controller due to the lower adapted rate. S4 receives the DL broadcast and schedule through the strong users S0 and S2, S5 receives successfully through S1, and S6 receives successfully through and S2.

DL Phases 3 through N (length T_{Di}, where $3 \leq i \leq n$) follow the same structure as DL phase 2, where transmissions use the rate

$$R_{Di} = \frac{x_D \cdot (b+2)}{T_{Di}}$$

which allows for N-hop relay paths. At the end of the last DL phase, all successful nodes have also received the block ACK. This allows any of them to participate as relays in the UL phases since they can calculate the UL transmission schedule. For the example in Figure 3.15, there are 3 DL phases, so there are up to 3-hop relay paths. Column 5 represents the third DL phase, where S7 receives the DL broadcast and ACK through S3 (which only received this information in DL phase 2).

Note that the strong nodes that received the information from the controller in DL phase 1 are the bottleneck for successful relay paths to other nodes during subsequent DL phases.

UL Phases 2 Through N The schedule (calculated from the block ACK) allocates a slot in UL phases 2 through N for each unsuccessful sensor/actuator node from UL phase 1. The total length of each UL phase is T_{Ui}, where $2 \leq i \leq n$. Time slots are divided evenly among all x_D unsuccessful nodes. In the slot for a given weak node j, node j and every other node who heard node j in an earlier UL phase simultaneously transmits the node j's message at the new rate

$$R_{Ui} = \frac{x_D \cdot b}{T_{Ui}}$$

where i is the UL phase number. In UL phase i, there is the potential for i-hop relay paths.

Column 6 of Figure 3.15 represents UL phase 2, where S2 and S0 transmit S4's message they overhead in UL phase 1 to the controller. Column 7 represents UL phase 3, where the S6 $\rightarrow$ S4 $\rightarrow$ S2 $\rightarrow$ C three-hop relay path allows S6's message to reach the controller successfully. Note that this relies on S4 hearing S6 in UL phase 1 and S2 hearing S4 in UL

phase 2. It is also possible to have new, shorter relay paths emerge due to using a lower rate in UL phases 2 through N. (e.g. S7 to S3 in UL phase 2 and S3 to controller in UL phase 3).

The UL phases are similar to their DL counterparts, but are in a sense inverted. The bottleneck to the controller now occurs on the last-hop, meaning UL phase N. As a final note, the exact transmission rates for each of the DL and UL phases depend on the time allocated and number of weak nodes.

3.4.4 Evaluation Methodology for Cooperative Relaying Architecture

For the initial system architecture, the minimum SNR was found by setting the desired reliability, finding the cycle time for a given SNR using network simulation, and sweeping the SNR until the cycle time equaled the required cycle time. This procedure required long simulation times for even moderate values of p, so the cooperative relaying system has been analytically modeled using a simplified link model so that the latency for smaller values of p can be found.

Assumptions

The following behavioral assumptions are made in addition to the resource assumptions in Chapter 3.4.3. As with the initial system architecture, all links have a a fixed nominal SNR with independent Rayleigh fading on each link, and each channel has a single tap (hence flat-fading). Channel reciprocity is assumed, which is consistent with a TDD system. Because the cycle time is very short, the delay-limited capacity framework is used [15, 73].

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

$$p_{link} = P(R > C) = 1 - \exp\left(-\frac{2^{R/W} - 1}{SNR}\right) \tag{3.3}$$

Multiple hops use the same idea where either the direct path succeeds as just described or any of the other multi-link paths succeed if all the links in it succeed. Figure 3.17 shows this idea for one-hop and two-hop paths.

If there are k simultaneous transmissions, 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. This ignores a subtle effect: the CDD space-time coding schemes that are envisioned for use in this system effectively make the channel response longer. This pushes the PHY into the "wideband regime" in wireless communication theory, and a full analysis must account for the required increase in channel sounding by pilots to learn this channel [74]. Preliminary results suggest that it will only add $2-3$dB to the SNRs required at reasonable network sizes.

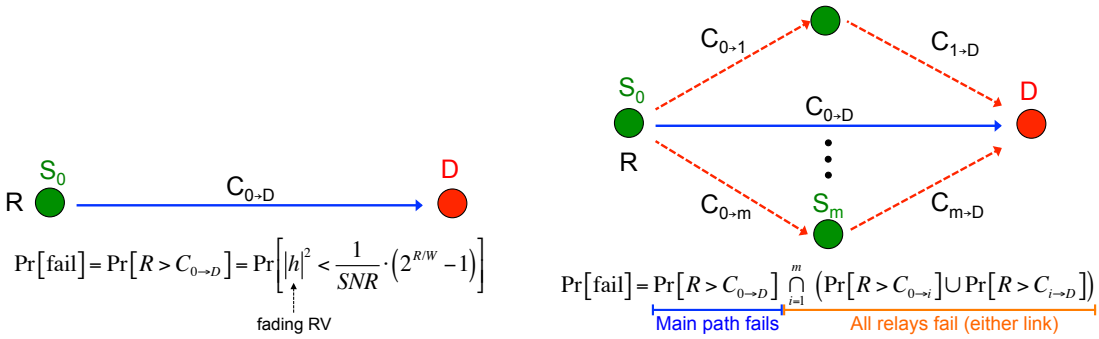


Figure 3.17: The link analysis procedure for a link with (a) one relaying phase and (b) two relaying phases.

The dispersion-style finite-block-length effects on decoding are not considered in this analysis (justified in spirit by [75]). A related assumption is that no transmission or decoding errors are undetected [76] — a corrupted packet can be identified and completely discarded. This is justified by considering all messages to include a 40-bit CRC code that can be added to the underlying message size.

Notation

1. A binomial distribution with r experiments, probability of *failure* p_e and number of successes s is denoted $B(r, s, p_e) = \binom{r}{s}(1-p_e)^s p_e^{r-s}$.

2. The probability of at least one experiment (out of r) failing is $F(r, p_e) = 1 - (1-p_e)^r$.

Reliability of 1-Hop Links

In the 1-hop version of the proposed cooperative relaying scheme, the controller transmits the information to the actuators in one broadcast, and the sensors send their information to the controller using TDMA. This section derives equations for the overall probability of failure of the DL and UL for the 1-hop scheme.

Downlink The rate of transmission is R_{D1}, which corresponds to a probability of failure of a single link p_{D1} as given by Equation (3.3). The DL succeeds if all actuators get their information from the single broadcast, so the probability of failure is

$$P(\text{fail}) = F(N, p_{D1}) \tag{3.4}$$

Uplink Similar to the DL, the rate of transmission is R_{U1}, which corresponds to a probability of failure of a single link p_{U1} as given by Equation (3.3). The UL succeeds if the controller

successfully decodes all sensor transmissions on the first attempt, so the probability of failure is

$$P(\text{fail}) = F(N, p_{U1}) \tag{3.5}$$

Reliability of 2-Hop Links

In the 2-hop version of the proposed cooperative relaying scheme, both the controller and sensors have two attempts to get their messages to their destination as described in Chapter 3.4.3. The sensors/actuators can relay for the controller and other sensors in their second attempts in order to help get the messages to their destination. The total time for DL in the 1-hop scheme is divided between the two phases in the 2-hop scheme, and the same is done for the UL. This section derives equations for the overall probability of failure of the DL and UL for the 2-hop scheme.

Downlink In the 2-hop DL, failure is the event in which at least one of the n actuators has not received its message by the end of the second DL phase.

The rate of transmission in phase 1 (R_{D1}) is dictated by the time allocated for this phase (T_{D1}) as described in 3.4.3. The rate of transmission in phase 2 (R_{D2}) depends on the number of successful actuators in phase 1 (x_D), which make up the set X_D, and the time allocated for phase 2 (T_{D2}) according to

$$R_{D2} = R_{D1} \cdot \frac{(n - x_D) \cdot T_{D2}}{n \cdot T_{D1}}$$

Let the probability of failure corresponding to R_{D1} and R_{D2} be p_{D1} and p_{D2}, respectively. The probability that the controller-to-actuator link fails in phase 2 given it failed in phase 1 is given by

$$p_{D2|D1} = \min\left(\frac{p_{D2}}{p_{D1}}, 1\right)$$

Then, the overall probability of failure of the 2-hop DL is given by

$$P(\text{fail}) = \sum_{x_D=0}^{n-1} \left\{ B(n, x_D, p_{D1}) \cdot F(n - x_D, (p_{D2})^{x_D} \cdot p_{D2|D1}) \right\} \tag{3.6}$$

Uplink For the 2-hop UL, failure is the event in which the controller does not successfully decode at least one of the n sensor's transmissions by the end of the second UL phase.

The rate of transmission in phase 1 (R_{U1}) is dictated by the time allocated for this phase (T_{U1}) as described in 3.4.3. The rate in phase 2 (R_{U2}) depends on the number of successful sensors in phase 1 (x_U) and the time allocated for this phase (T_{U2}) according to

$$R_{U2} = R_{U1} \cdot \frac{(n - x_U) \cdot T_{U2}}{n \cdot T_{U1}}$$

Let the probability of failure corresponding to R_{U1} and R_{U2} be p_{U1} and p_{U2}, respectively. There are two cases to consider:

1. **Case 1:** $R_{U2} > R_{U1}$**.** Some links which were "good" might become "bad" because the now higher rate is larger than the capacity of the faded link. Let Y_U (with cardinality y_U) be the set of nodes in X_U that still have "good" links. The probability of a link becoming "bad" is $q = p_{U2}/(1 - p_{U1})$

2. **Case 2:** $R_{U2} < R_{U1}$**.** Some links that were "bad" might become "good" because the now lower rate is smaller than the capacity of the faded link. This occurs with probability $\tilde{q} = (1 - p_{U2})/p_{U1}$. Let S be the set of all sensors/actuators, and then let $\tilde{Y}_U$ be the nodes in $S \setminus X_U$ that get added in phase 2. Let the minimum number of nodes that need to succeed in phase 1 to enable newer links to the controller be x_o.

The probability of failure of the 2-phase UL is then given by

$$\begin{aligned} P(\text{fail}) = &\sum_{x_U=0}^{x_o-1} \sum_{y_U=0}^{x_U} \left\{ B(N, x_U, p_{U1}) \cdot B(x_U, y_U, q) \cdot F(n - x_U, p_{U1}^{y_U}) \right\} \\ &+ \sum_{x_U=x_o}^{n-1} \sum_{\tilde{y}_U=0}^{n-x_U-1} \left\{ B(N, x_U, p_{U1}) \cdot B(n - x_U, \tilde{y}_U, 1 - \tilde{q}) \cdot F(n - x_U - \tilde{y}_U, p_{U1}^{x_U+\tilde{y}_U}) \right\} \end{aligned} \tag{3.7}$$

Reliability of N-Hop Links

Using the method for deriving the reliability equations for the 2-hop DL and UL, the reliability equations for the 3-hop and, in general, the N-hop scheme can be obtained. The key is finding the probability of failing in the first phase, then the conditional probability of failing in the second phase given the number of failures in the first phase, then continuing this processes until finding the conditional probability of failure in the last phase conditioned on all previous phase results.

3.4.5 Results for Cooperative Relaying Architecture

Comparison with Baseline Schemes

Following guidelines of Chapter 3.1.5 and the communication theoretic convention, the minimum SNR required to achieve a reliability of p is used as a metric to compare the proposed cooperative relaying protocol to other baseline schemes. The analysis in this section differs somewhat from that used for the initial system architecture in Chapter 3.3.5. Instead of setting the reliability and finding the cycle time at different SNRs, the cycle time is fixed by setting the overall spectral efficiency (using the total number of information bits to be transmitted $n \cdot b$) and the reliability is found at different SNRs. All results in this section

are for a reliability of 10^{-9} and where individual links are modeled as diversity 1 Rayleigh fading channels. Where spectral efficiency is not a parameter, each node transmits 160 bits of information on the UL and DL, channels have a bandwidth of $W = 20$MHz, and the $t_{\text{cycle,req}} = 2$ms.

Figure 3.18 shows how the performance of the proposed cooperative relaying scheme and two baseline schemes scales with the number of nodes n in the system. The two solid lines correspond to the proposed cooperative relaying scheme using 2 and 3 hops. Even though the throughput increases as $b \cdot n$, the minimum required SNR for the proposed cooperative relaying scheme initially decreases with increasing n and flattens for $n > 10$ because of the increased reliability from multi-user diversity. Eventually, the gains from multi-user diversity give way and the required SNR starts to increase for very large n, although this regime is not shown in Figure 3.18.

The topmost curve in Figure 3.18 corresponds to the first baseline scheme that restricts the UL and DL to a single hop, which is similar to the initial system architecture in Chapter 3.3. The required SNR is extremely large because the system has no means to increase diversity with an increasing number of nodes, but it must still transmit at a proportionately higher spectral efficiency due to the increasing $n \cdot b$ and constant $t_{\text{cycle,req}}$.

The curve directly below the topmost is for the second baseline scheme, which is a purely hypothetical scheme that allows each node to use the entire $t_{\text{cycle,req}}$ for its own UL and DL message but without any relaying and thus also no diversity. This bounds what could possibly be achieved by using adaptive HARQ techniques. As expected, it has (marginally) better performance than the first baseline scheme, but it has the same scaling trend with increasing n.

The final curve represents a hypothetical (non-adaptive) frequency-hopping scheme that divides the bandwidth W into k sub-channels that are assumed to be independently faded. This can be viewed as an idealized version of WSN systems, such as WirelessHART. As k (and thus the number of frequency hops) increases, the available diversity increases, but the added message repetitions force the instantaneous link data rate higher. This leads to a tradeoff between spectral efficiency and diversity, so the optimal value of k is used, which is labeled on the performance curve. For low n, the optimal version of the scheme uses more frequency hops because of the diversity benefits. The SNR cost of doing this is not high because the throughput is low enough that it is still in the linear regime of channel capacity. According to Figure 3.18, frequency-hopping is preferred for $n < 7$ — as long as there are 20 or more independently faded sub-channels to repeat across.

It turns out that the aggregate throughput required (overall spectral efficiency considering all users) is the most important parameter for choosing the number of relay hops in the proposed cooperative relaying scheme. This is illustrated in Figure 3.19, which shows the SNR required and the optimal number of hops to use for a given n. With one node, a 1 phase scheme is all that is possible. As the number of nodes increases, the number of possible hops increases as well. However, a larger number of hops increases the required data rate since an N-hop scheme has effectively $N\times$ less time to transmit the same amount of information. This increase in instantaneous rate dominates the the choice of optimal the number of hops.

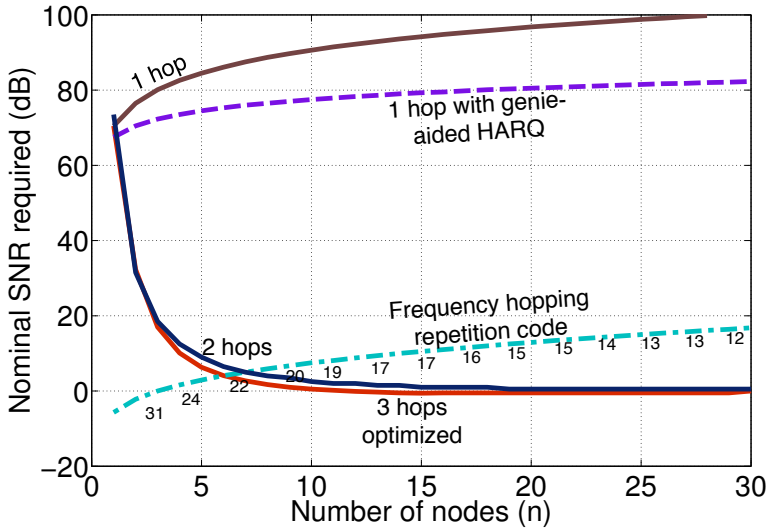


Figure 3.18: The performance of the proposed cooperative relaying scheme and several baseline schemes for $n = 30$ nodes transmitting $b = 160$ bit messages on the UL and DL over a 20MHz bandwidth in a 2ms cycle time. The reliability is set at 10^{-9}. The numbers next to the frequency-hopping scheme represent the amount of frequency diversity needed.

In principle, at high enough aggregate rates, the one-hop scheme will be best even with more users. When the target reliability is 10^{-9}, this occurs is at very high and impractical aggregate rates (around 40b/s/Hz). In the practical regime, diversity is better.

Phase Length Optimization

It may seem natural to divide time evenly between phases so that links in different phases fail with the same probability, but Figure 3.20 shows that performance can be improved with an optimized allocation. Each phase of the protocol has a different level of importance in reaching the target reliability, which is seen by looking at the optimal phase lengths.

To understand why it is better to allocate more time to phase 1 of the DL, consider the difference between a link that fails in phase 1 and a link that fails in a later phase. A link between node i and the controller that fails in phase 1 is equivalent to all of the other $n-1$ links at node i failing in phase 2. A link connected to node i that fails in phase 2 does not prevent other nodes from using node i as a relay from the controller. Therefore, a link between node i and the controller is on many more paths from the controller than a link connected to node i in phase 2. As a result, the qualities of the links between the controller and each node are the bottleneck of the system. Giving more time to phase 1 during DL improves these critical links at the expense of less important links in later phases. This explains why the DL performs better with a longer phase 1. A similar argument can be used

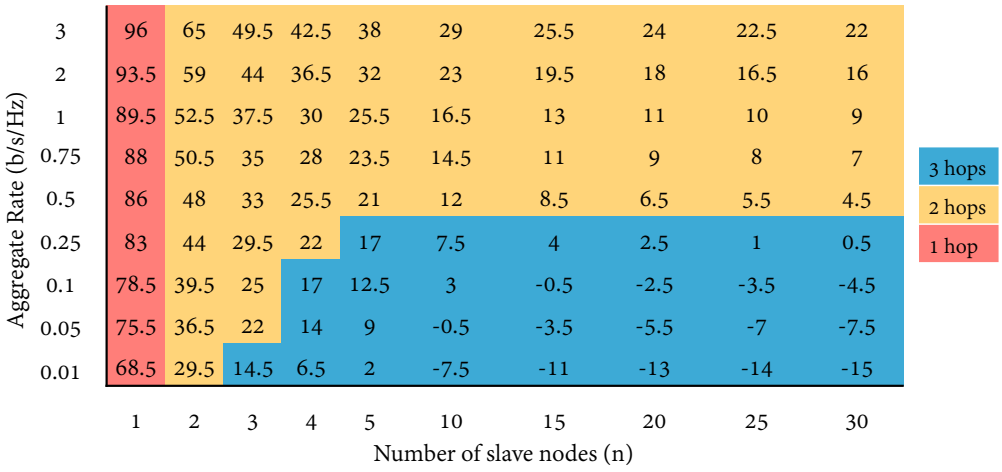


Figure 3.19: The "DUI" chart lists the number of hops and minimum SNR required to achieve a reliability of 10^{-9} for various combinations of aggregate rate and number of users. Here, the division of time between phases is unoptimized. The UL and DL are allocated equal time and all DL (UL) phases are the same length.

for UL, except that the bottleneck links to the controller are in the last phase. As a result, the UL sees the opposite trend and performs better with a shorter phase 1 and longer last phase.

There is one further important optimization to consider: the times allocated for UL and DL. For the 2-hop scheme, the DL phase's probability of error is slightly larger than UL's, so giving the DL more time at the expense of the UL may improve the overall system performance. Similarly, the UL in the 3-hop scheme dominates the overall system performance, so giving the UL more time improves overall system performance, as shown in Figure 3.20.

A "Diversity Meter"

Traditional formulations of diversity as a concept are asymptotic: they look at high SNR and very low error probabilities. While the proposed cooperative relaying scheme is also designed for the low probability of error regime, the targeted SNR is not asymptotically high. Studies of the finite-SNR diversity-multiplexing tradeoff tend to find that diversity is lower than expected in the non-asymptotic regime [77]. However, it is still unclear how the non-asymptotic diversity numbers should be interpreted. To resolve this question, a "diversity meter" is proposed as another baseline scheme that considers "frequency diversity" as *the* paradigmatic form of diversity, which can be used convert the diversity a scheme provides to a frequency diversity equivalent.

To get a fair bound, the diversity meter is not restricted to non-adaptive repetition codes over frequency as was done for the frequency hopping scheme in Chapter 3.4.5, but is

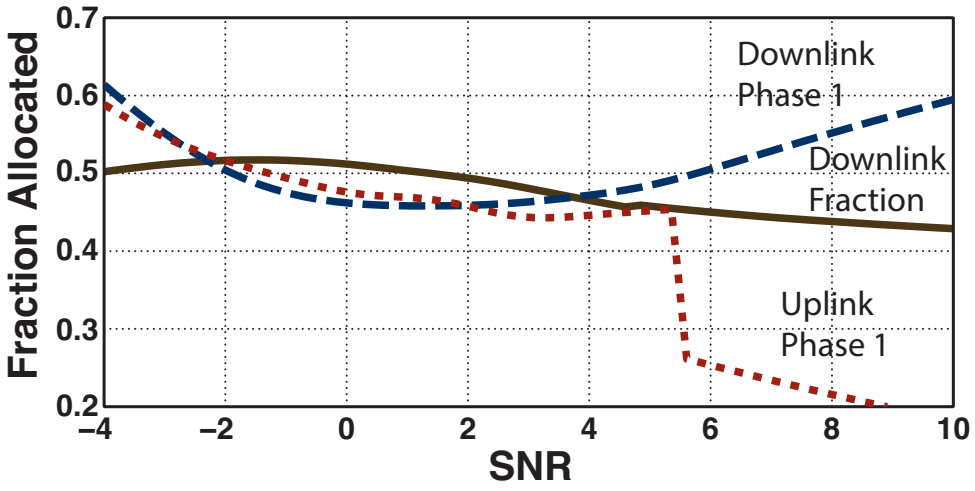


Figure 3.20: Optimal fraction of time allocated for phase 1 for the UL and DL and the optimal fraction of time spent on DL vs. UL in the 3-hop scheme at various SNRs. Both the DL and UL have better performance with more time allocated for the bottleneck links to the controller; in DL this means a longer phase 1, and in UL this means a longer final phase. Parameters used were 160 bit messages, 30 users, 2×10^4 total bits.

allowed to intelligently code across all frequency blocks. The frequency diversity is modeled as dividing the bandwidth into k independently faded blocks. An outer erasure code of rate $R_o = k'/k$ codes over all the blocks so the overall transmission succeeds if k' blocks are decoded. Each block uses a capacity-achieving inner code of rate R_i that succeeds if the block's capacity is greater than R_i. The aggregate information rate is then given by $R_o \cdot R_i \cdot W$. The diversity required is defined as the smallest k, optimized over R_o and R_i to meet the desired performance. This is illustrated in Figure 3.21 where one can look up the minimal frequency diversity required to achieve a probability of error of 10^{-9} at different combinations of aggregate rate and SNRs.

This turns out to be equivalent to schemes that know which of their diverse sub-channels are not too deeply faded and only transmit in those sub-channels (provided the transmitter is not allowed to boost the local SNR in this channel to compensate for other deeply faded sub-channels). In other words, the transmitter must comply with a flat spectral mask, not an average power-constraint that permits frequency-shaping.

The proposed cooperative relaying scheme with 30 users and aggregate rate of 0.25 bits/s/Hz achieves a probability of error of 10^{-9} at 0.42dB. With identical constraints on SNR, bandwidth, and probability of error, Figure 3.21 shows that the diversity meter requires a diversity of 201 with $R_o = 1/3$ and $R_i = 3/4$. Contrary to popular belief that the ideal diversity is approached from below, this shows that at finite-SNR and for this intuitive sense of diversity, the multiuser diversity obtained with 30 users is comparable to having 201 independently faded sub-channels for every point-to-point link between controller and

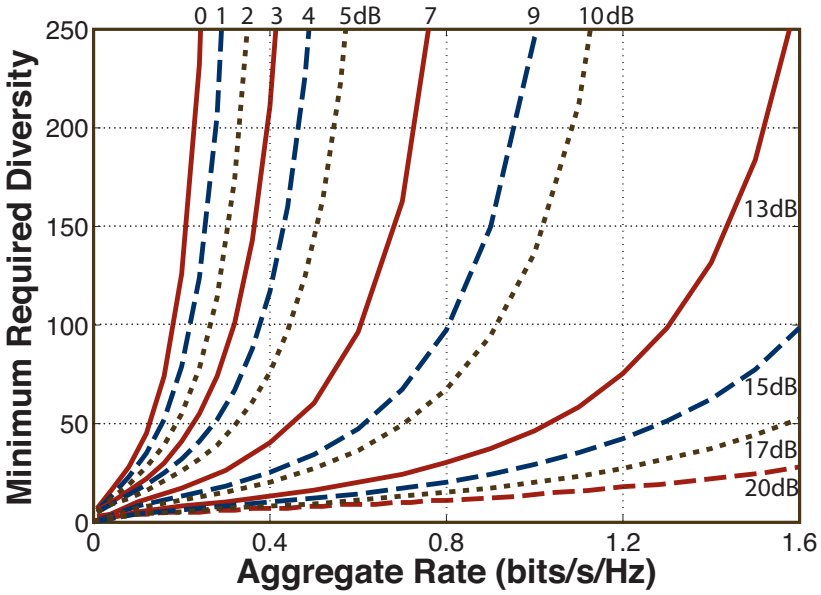


Figure 3.21: Diversity Meter: given an SNR, follow the appropriate line to the desired aggregate rate. The y-axis gives the minimal frequency diversity that a genie-aided frequency-hopping scheme would need in order to get $p = 10^{-9}$ at this SNR.

the sensors/actuators! This drastic difference is because the proposed cooperative relaying scheme gains diversity in one time slot instead of over many time slots, which allows for a lower effective R_i. Multiuser diversity is better than frequency diversity in our context.

Robustness to Modeling Error

Given the extremely low error probabilities required in a wireless setting, it is natural to question the impact of modeling error and uncertainty. Should anyone really trust the fading distribution down to 10^{-9}?

To better understand this, consider the DL of the proposed cooperative relaying scheme with 2-hops, time evenly divided between both phases, and no rate adaptation (i.e. all messages are repeated in both phases: $R_{D_1} = R_{D_2}$). The probability of failure of the scheme p is calculated by (3.6). For this case, the top curve in Figure 3.22 shows the maximum probability of single link failure the system can tolerate for different numbers of nodes n, while keeping p constant at 10^{-9}. The tolerable probability of link failure in the cases of interest ($n \geq 20$) is fairly high (above 15%), and fading models are quite good in this regime.

What if modeling error or the local industrial environment's interference introduced an extra probability of failure at each link, p_{env}, on top of the probability of error due to nominal fading, p_{fade}? Then the probability of link failure, p_{link} is the combined effect of both these parts: $p_{link} = p_{fade} + p_{env}$.

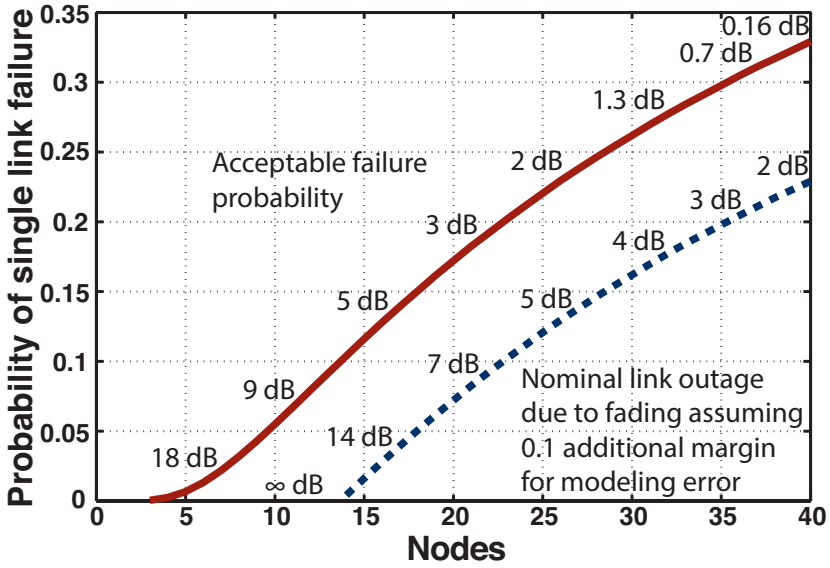


Figure 3.22: The probability of link failure that can be tolerated in the DL of a 2-hop scheme as a function of the number of nodes. The aggregate rate is held constant at 0.25 bits/sec/Hz. The lower curve is a shifted down of the upper curve (by 0.1), and the SNR numbers represent the nominal SNR required to reach that particular p_{link} for Rayleigh fading.

The bound on the tolerable p_{fade} can be obtained by shifting the p_{link} curve down by p_{env}, which is done in Figure 3.22) for $p_{env} = 0.1$. To attain the same p_{link} for a given number of nodes, the SNR needs to increase. Figure 3.23 shows the required SNR penalty increases for larger levels of robustness. For small network sizes ($n < 15$), achieving the probability of error requirement robustly can become impossible (when the desired tolerance to modeling error is itself greater than the maximum tolerable p_{link}). For moderate to large network sizes ($n \geq 25$), the protocol only has a small SNR penalty ($\approx 3dB$).

From this, it is concluded that the proposed cooperative relaying scheme does not rely on perfect knowledge of deep fading distributions and achieves high-reliability by relying on the independence of link failures. It is natural to wonder this assumption of independence holds. Preliminary results based on [78] indicate that even with a limited number of scatterers, multi-path will be essentially independent (or better) for the diversity perspective here. It is interesting to see that diversity contrasts with multiplexing gain, which is capped by the richness of the scattering environment.

A final consideration is whether or not the actual optimization of the scheme parameters (phase lengths) should change greatly in the presence of modeling errors. Figure 3.24 considers three different phase length allocations for the DL in the 2-hop scheme: the naïve 50/50 allocation, an allocation optimized to account for both p_{fade} and p_{env} as part of p_{link}, and one that only chooses phase lengths based on p_{fade} (even though p_{link} contains p_{env}). In

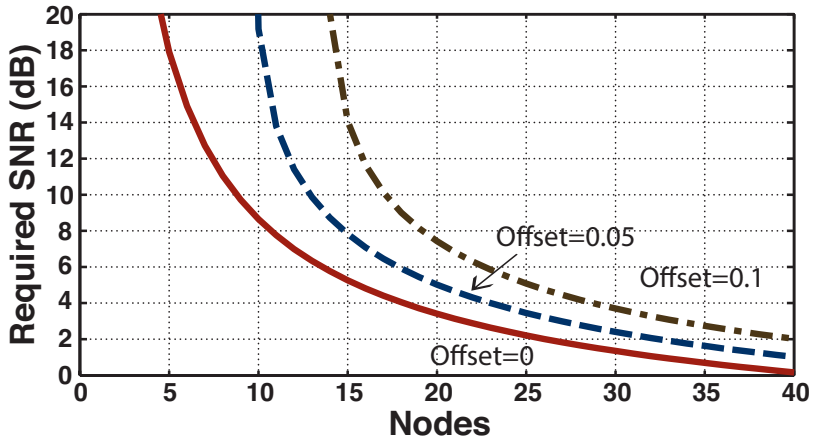


Figure 3.23: The SNR penalty to achieve performance robustly. For an increasing amount of robustness, a higher SNR is required as well as more nodes.

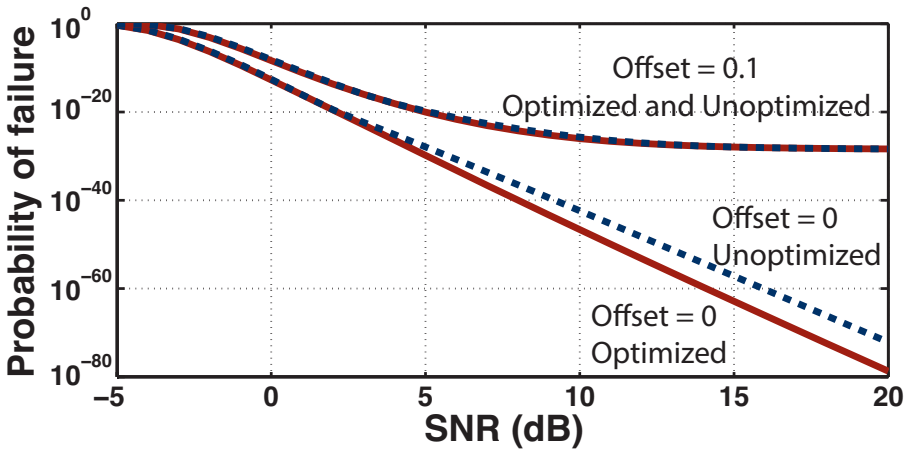


Figure 3.24: Performance of the DL in the 2-hop scheme in the presence and absence of a worst-case modeling error that causes links to fail with a 0.1 higher probability regardless of SNR.

the absence of modeling errors (i.e. accurate knowledge of deep fading distributions), the optimizing phase lengths provide a significant performance advantage over the naïve allocation. However, accounting for the modeling errors reduces the impact of optimization. This suggests that for phase length optimization to be effective in yielding a robust strategy, the fading model need not be highly accurate.

3.5 Conclusions

Industrial control systems currently use wired networks because no wireless solutions exist that can meet the required specifications, but wired systems have many mechanical problems that a wireless solution would solve. Additionally, the IoT of the future has similar structure and specifications as current industrial control applications, and a wireless solution is a must for the IoT. Therefore, a high-performance wireless network geared towards these types of systems must be developed.

A model of a representative industrial control application was developed along with a set of specifications for the communication network. Systems such as IEEE 802.11 and LTE cannot meet the specifications because they cannot handle multiple access and retransmissions efficiently when there are many nodes in the system, so an initial system architecture was developed to address these issues. It was found that the system's latency depends strongly on the amount of diversity available to each user since a single node being in deep fade requires many retransmissions (and hence higher latency).

Unfortunately, nature cannot be relied upon to provide enough diversity such that the probability of any single sensor/actuator to controller link being in a deep fade is smaller than the reliability requirement, especially when the number of nodes in the network increases. Coordinated multipoint is one option to generate diversity, but it requires the placement of many access points that likely must be wired together. As a more flexible alternative, a cooperative relaying system was developed that uses the links between non-controller nodes to route around deep fades and manufacture diversity when nature does not provide any. To reduce the overhead from determining routing through the network and multiple transmissions, all relays transmit simultaneously using a distributed space-time code, such as cyclic delay diversity. Initial transmissions are scheduled during system initialization, and retransmission scheduling is determined as needed using a decentralized algorithm. The proposed cooperative relaying system and other baseline schemes are analyzed using a simplified link level analysis, and the proposed system significantly outperforms the other schemes as long as there are a modest number of nodes in the network. Also, the proposed system is robust to interference and fading model accuracy in the extreme tails of the fading distribution.

Chapter 4

PHY Implementation Aspects

The cooperative relaying system architecture described in Chapter 3.4.3 consists of mainly of the MAC and PHY layers, so these are the main components that must eventually be implemented. The MAC layer does not require compute intensive algorithms and does not have highly latency-sensitive functions, so it can easily be implemented on a dedicated micro-controller, an FPGA, or an ASIC. The latter two options are most attractive since it provides controllable, predictable latency. On the other hand, the PHY layer contains many complex algorithms that are highly latency sensitive, so it must be completely implemented in hardware. Ideally, the system would reuse as many blocks as possible from one or many current wireless PHYs in order to minimize the development and verification costs. This would also enable integration of the proposed architecture with a current wireless PHY into a single chipset drives volume and makes the chipset cheaper and more versatile for control applications. In particular, if the blocks come from an existing wireless PHY used for high-performance communication, the high-performance mode can be used to transfer large amounts of data between the controller and sensors/actuators (say for buffering the locations and colors to print for the industrial printer), and the control mode can be used during operation of the control system. The rest of this chapter explores aspects of a possible PHY implementation of the cooperative relaying system architecture from Chapter 3.4.3 that reuses as many blocks as possible from existing wireless PHYs while also meeting the architecture's unique requirements.

4.1 Background

4.1.1 Wireless Transceivers

The PHY reliably transports a data frame from the source across the physical medium to each destination. Since no other layer interfaces with the physical medium, the PHY contains all specifications for the digital baseband and analog front end (AFE), such as error

Some or all of the material in this chapter was the product of joint work with Paul Rigge.

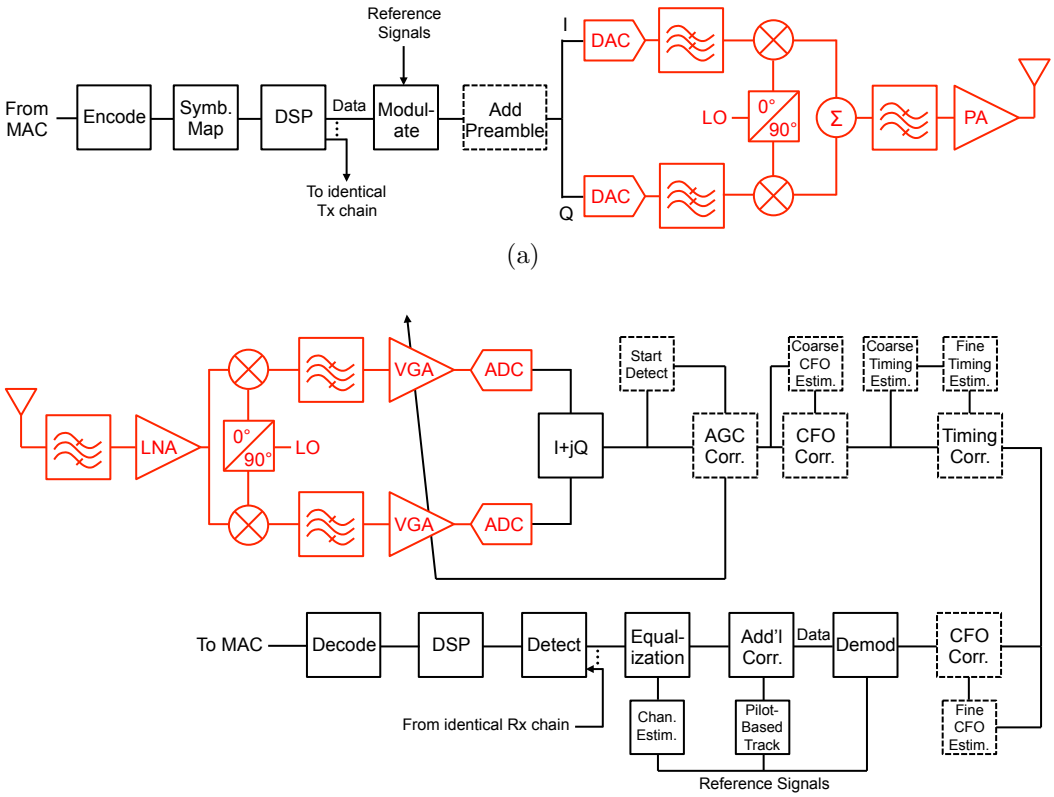


Figure 4.1: Digital baseband and analog front end for (a) the transmitter and (b) the receiver. The analog front end blocks in each chain are in red, and the digital baseband blocks are in black. Continuously connected systems, such as LTE, only used the dashed blocks when connecting to the network (otherwise it uses the reference signals embedded in the modulated signal to adjust CFO, timing, and channel estimates). In a multiple antenna system, each antenna has a separate stream that is processed by identical Tx/Rx chains. Only one of the chains are shown on the Tx and Rx side, but the streams for the other chains are shown at the DSP block on the Tx side and the detect block on the Rx side.

control coding, antenna configurations, sampling rate, carrier frequency, and the spectral mask. Figure 4.1 shows a general PHY transceiver, which is a more general version of the IEEE 802.11ac transceiver presented in Figure 2.20.

On the Tx side, the digital baseband encodes each data frame, groups and maps the digital bits into complex symbols from a constellation, performs DSP operations that depend on the physical medium and number of antennas/streams, and digitally modulates the symbols including adding in pilot/reference and preamble information to form a complex digital baseband signal with symbol rate f_s.

The AFE for each antenna converts the complex baseband digital signal into a passband analog waveform centered at carrier frequency f_c using digital-to-analog converters (DAC),

filters, a local oscillator (LO), mixers, and a power amplifier (PA). The filters shape the output spectrum so that it complies with the spectral mask that specifies bandwidth and tolerable adjacent channel leakage. Mixers using an LO tuned to f_c upconvert the I and Q signals to RF frequencies separately (the two components use LOs that are 90° out of phase). The I and Q signals are then added together and amplified to form the output waveform that goes to the antenna.

On the Rx side, the AFE for each antenna filters the output from the antenna based on f_c and the bandwidth to eliminate signals from adjacent channels and then amplifies the received waveform using a low-noise amplifier (LNA). It downconverts the I and Q components to baseband separately using mixers and the destination's LO tuned to f_c (and offset from each other by 90°). Sharp channel-selection filters reduce out-of-channel content, a variable gain amplifier (VGA) controls the dynamic range of the signal, and an analog-to-digital converter (ADC) sampling at $N \cdot 2f_s$ brings the analog information back into the digital domain ($N > 1$ is the oversampling factor). The VGA is necessary to keep the ADC from clipping since ADCs have a limited dynamic range, and a feedback loop involving the digital baseband controls the VGA's gain.

If the system is not scheduled and continuously synchronized (or it is and is establishing the initial connection), the digital baseband begins by detecting the start of a packet and adjusting the VGA gain using an automatic gain correction (AGC) feedback loop, usually by correlating the periodic preamble sequence. It corrects carrier frequency offset (CFO) that arises from the Tx and Rx having slightly different LO frequencies due to static manufacturing differences, different temperatures, and phase noise. Uncorrected CFO results in the received constellation rotating over time since there is a linearly changing phase error with time. It corrects the symbol timing either by adjusting the ADC's sampling clock or by interpolating the sampled signal and choosing the correct output from a matched filter. Next, for either burst or continuously connected systems, the system demodulates the non-preamble portion of the signal and separates the data from pilots. The pilots are used for further phase and timing offset correction and for channel estimation. The data enters a channel equalizer that uses the channel estimates, detects the constellation symbols from each chain, and undoes any DSP and coding that was introduced by the Tx digital baseband either within a single stream or across multiple streams. The resulting bits are reframed and passed to the MAC layer.

Duplexers separate the Rx and Tx chains on a single transceiver chip (not shown in Figure 4.1). In a time division duplexed (TDD) system where only one of the Tx and Rx is used at a time, the duplexer is simply a switch that selects between using either the Tx or Rx chain, and it requires several microseconds to switch between the two modes. In an frequency division duplexed (FDD) system where the Tx and Rx chains operate simultaneously, the duplexer isolates the two chains so that the high power Tx signal does not reduce the sensitivity of the Rx chain. The isolation filters usually must be implemented off-chip due to their stringent roll-off requirements.

The MAC layer can configure many of the blocks in the PHY in order to optimize the throughput or reliability of the network. For example, it can change the coding rate, the

symbol constellation, and the channel within the frequency band.

4.1.2 Cooperative Relaying System for Wireless Control

The cooperative relaying system for wireless control proposed in Chapter 3.4.3 has four distinct message types (modes) that must be considered when designing the physical layer:

1. The controller broadcasts a large frame containing information for all actuators.
2. The controller and relays broadcast a large frame with information for all actuators.
3. The sensors each transmit a small frame to the controller by themselves.
4. The sensors each transmit a small frame to the controller with help from relays.

Each mode has different performance requirements, such as synchronization in time and frequency, channel estimation, and coding. In addition, switching between modes cannot add excessive latency.

4.2 Duplexing

There are two forms of duplexing in wireless communication systems: FDD and TDD. FDD separates UL and DL transmissions onto different frequency bands. In current systems, this requires that the UL and DL must statically split the total available bandwidth, but it enables full-duplex operation. Even the problem of sharing the same spectrum for full-duplex operation is solved for point-to-point systems, relays will not be able to decode the overlapping transmissions if they do not have extra antennas. TDD requires that the UL and DL occur at different times, but each uses the entire frequency band when transmitting. This only allows half-duplex operation, and terminals require several microseconds to switch between transmit and receive modes [79].

FDD may seem an attractive option since it allows simultaneous transmission and reception, but TDD is the more appropriate choice for the cooperative relaying system for three reasons. First, a TDD system uses the full available bandwidth of the system, which enables the maximum amount of frequency diversity to be harvested from the system. FDD systems typically use contiguous parts of the spectrum for the UL and DL that are separated from each other by a guard band, so they cannot use interleaved segments of the entire band to get the full frequency diversity. Emerging FDD systems aggregate non-contiguous bands, but each band must be separated by additional guard bands, lowering the spectral usage. Second, the proposed cooperative relaying system's MAC layer does not benefit greatly from full-duplex operation because of the system's scheduled operation. Terminals are either transmitting as the source or as a relay or listening in order to receive their data or so that they can act as a relay in another phase. The only benefit of full-duplex comes from eliminating the switching time between Tx and Rx modes. However, the switching time in

TDD systems is so short that a node would only miss the transmission directly before or after its own transmission, which would not harm the overall reliability of the cooperative relaying system with a moderate to large number of sensors/actuators. Third, FDD requires additional channel sounding since there is no reciprocity between the UL and DL channels. With all of this in mind, the cooperative relaying system should use a TDD-based PHY.

4.3 Analog Front End

The design, implementation, and testing of an AFE requires significant effort because all components are custom and require careful layout and extensive testing. Fortunately, an AFE for a particular frequency band can be reused for another design that uses the same band as long as the RF requirements for the original application are similar or better than those for the intended application. Since the cooperative relaying architecture of Chapter 3.4.3 does not necessarily have strict RF requirements aside from those imposed by the FCC and each of the message types described in Chapter 4.1.2 do not depend on the RF characteristics, any existing AFE can potentially be used as long as it can support the modulation and bandwidth assumed by the digital baseband and does not require additional processing in the baseband that causes excessive amounts of latency. If the AFE is from a widely-used standard, the standard's baseband and the wireless control baseband can be integrated on the same die and share the AFE. The chipset could then support both high-data rates for traditional user-centric systems or low-latency, high-reliability operation for machine-centric and control systems. This would also allow the control systems to leverage the volume of user-centric chipsets to reduce the cost of wireless control systems.

The AFE for IEEE 802.15.4-based systems (ZigBee and WirelessHART) is designed for small bandwidths that cannot support the data rates required by most high-performance industrial control applications.

The IEEE 802.11ad AFE supports very large bandwidths (2.16GHz), but it requires maintaining a beamformed signal between terminals and cannot support omnidirectional transmissions. Reestablishing a beamforming lock after it is lost can take hundreds of microseconds per pair of terminals, which adds excessive latency when many terminals must do this. Omnidirectional transmissions, used extensively in the DL and relaying phases of the proposed cooperative relaying scheme, are approximated by repeating the same transmission in each sector, which reduces the effective data rate significantly.

The LTE AFE supports moderate bandwidths, omnidirectional transmissions, both TDD and FDD, and has moderate power consumption. Unfortunately, it exclusively uses licensed bands. This means that LTE AFEs used in wireless control systems can only be used where they would not interfere with cellular operation or where the user has licensed spectrum. In general, this is too strict of an assumption, so it is not appropriate in most cases. In the future, LTE may be used in the unlicensed bands, and a TDD-LTE AFE designed for those bands may be acceptable for the cooperative relaying system.

The IEEE 802.11ac AFE transmits in the unlicensed ISM band, supports omnidirectional transmissions, uses TDD, and supports moderate bandwidths ($\leq$ 160MHz) that can support moderate to high data rates depending on the antenna configuration. There is much effort from both industry and academia to reduce the power of IEEE 802.11ac AFEs, and the volume of these AFEs will be very large since they are used in virtually all mobile devices.

Overall, the AFE for IEEE 802.11ac is a good candidate for the AFE for wireless control system radios, and the future AFE for TDD-LTE in the unlicensed bands may be acceptable.

4.4 Modulation

Since the AFE is (mostly) agnostic to the type of modulation, the cooperative relaying system is free to choose one that best suits its requirements. There are many candidate modulation techniques that have been used in past and current standards and that have been proposed for 5G wireless communication, such as single carrier (SC), direct sequence spread spectrum (DSSS), frequency hopping spread spectrum (FHSS), orthogonal frequency division multiplexing (OFDM), ultra-wideband (UWB) [80], generalized frequency division multiplexing (GFDM) [5], faster-than-Nyquist (FTN) [4], filter-bank multicarrier (FBMC) [6], and universal filtered multicarrier (UFMC) [7].

The chosen modulation cannot introduce excessive delays, particularly from using large blocks of data to form modulation symbols. Any of the multicarrier schemes modulate blocks of data, but FBMC and UFMC use the largest size because data must be arranged into a sizable 2-D frame whose dimensions represent time and frequency. They require "ramp up" time for their filters, which increases the delay, especially when the system ideally transmits many smaller frames. GFDM has a moderate size since it uses smaller 2-D frames.

The modulation should not introduce excessive transmitter or receiver complexity, which is required for FTN and FBMC (and to a lesser extent UFMC) due to their complex filtering or equalization. The Doppler immunity of the modulation is not of particular importance because objects in the manufacturing environment and the machines themselves move on the order of meters per second.

The cooperative relaying system requires a distributed space-time scheme that enables the receiver to harvest multi-user diversity and that adds minimal latency. Cyclic delay diversity is one option for multicarrier systems since it creates frequency diversity from spatial/multi-user diversity. DSSS systems can use rake receivers to separate each path or different codes to separate users. SC and UWB do not have options readily available to them.

The modulation would ideally have immunity to interference from other systems in the same band beyond the robustness provided by the cooperative relaying system. Only the spread-spectrum schemes can do this in a decentralized manner. DSSS can have neighboring systems use different spreading codes, and FHSS can assign different hopping patterns to neighboring systems.

Finally, the modulation should have a good tradeoff for performance and synchronization accuracy. The single carrier schemes are more immune to frequency offsets than multicarrier schemes since there is no inter-carrier interference. However, multicarrier schemes can be configured to reduce the requirements of frequency and timing synchronization at the expense of data rate. OFDM has the worst tradeoff of the multi-carrier schemes.

Overall, OFDM is a good choice for the modulation because it does not introduce excessive latency, has low transmitter and receiver complexity, and has a viable distributed space-time scheme that converts multi-user diversity to frequency diversity and can trade off its performance and synchronization requirements easily (although not optimally). Also, it has many current implementations in both IEEE 802.11 and LTE that can be reused. Since each message does not contain much data, shorter OFDM symbols lengths are desired, so IEEE 802.11 would be preferable. The parameters for OFDM (or any other multicarrier modulation) will be discussed in Chapter 4.6.

4.5 Multiple Access

As with modulation techniques, many multiple access protocols have been used in standards and proposed for 5G including time division multiple access (TDMA), frequency division multiple access (FDMA), code division multiple access (CDMA), orthogonal frequency division multiple access (OFDMA), the extensions of OFDMA to the other multicarrier modulations, and non-orthogonal multiple access (NOMA) [81]. Out of these options, TDMA is the most promising multiple access scheme since the latency constraint is too strict to schedule users by their (subchannel) SNR, the cooperative relaying scheme works best when all users listen to each transmission, and TDD will be used. TDMA allows each user to use the full bandwidth to obtain maximum frequency diversity, and it requires less accurate frequency synchronization. Only if there are too many users to allocate each their own time slot should static OFDMA be used (where static means the time/frequency resource allocation for each user is predetermined). These observations are essentially suggested by the setup of the cooperative relaying protocol. Many current standards can support this form of multiple access, so it does not place any restrictions on the hardware.

A related issue is interference between other systems using the same frequency bands. If the system is another wireless control system, the controllers for each system can coordinate so that each has orthogonal time slots if the specifications allow. Otherwise, they can use TDMA for intra-system multiple access, and CDMA for inter-system multiple access. Using CDMA would also help with interference from other types of interfering systems, although the system design would need to take the extra interference into account to make sure the system still meets specifications. An alternative is to require all systems within range of a wireless control system to be managed by the control system's controller. In the case of multiple control systems, a leader will be selected to coordinate all networks. This would allow the control systems to guarantee their resources and the other systems can use whatever remains. Note that any sort of carrier sense protocol would not be viable since the control

systems would never be able to guarantee their specifications.

4.6 Parameter Estimation and Correction

4.6.1 Considerations

Non-ideal effects such as AGC, CFO, timing offset, and channel variations harm the performance of the system because they cause inter-carrier interference (ICI), inter-symbol interference (ISI), and reduced SNR at the receiver. In order to maintain the required reliability at the desired spectral efficiency, these non-idealities must be estimated and then corrected. When the transmitter and receiver have no estimate of the non-ideal effects, the transmitter prepends a periodic sequence of symbols known to both the transmitter and receiver called a preamble. The receiver compares the received signal and what it knows the signal should be in order to estimate each non-ideal effect in turn. In general, a longer sequence that is repeated more times in the preamble enables a better estimate of each effect. Unfortunately, preambles carry no information, so the spectral efficiency to transmit the same amount of data must be higher to account for the preamble's overhead.

The non-ideal effects vary with time, so the estimates obtained from the preamble become out of date, so updated estimates are required to maintain the desired performance. For this reason the transmitter embeds reference/pilot symbols throughout the transmitted data that are known to the receiver. For multicarrier modulations, reference symbols placed in time and frequency give better estimate updates. As with preambles, the reference symbols carry no information, so more reference symbols require a higher spectral efficiency for the same amount of information (and thus a higher required SNR). Reference symbols generally use less resources than preambles since they are used to update estimates instead of calculate estimates from scratch.

The amount of preamble and reference symbols must be balanced against the required SNR increase to meet the system's spectral efficiency requirements. If communication is continuous, preambles may not be needed after system initialization.

Other trade-offs can be made between spectral efficiency and the accuracy of parameter estimation by adjusting the modulation's parameters. For example, the ICI caused by CFO can be mitigated by increasing the subcarrier spacing, which results from a smaller FFT size for the same bandwidth. Increasing the cyclic prefix length relaxes the timing synchronization requirements of each terminal since transmitting at slightly different times looks like multipath.

4.6.2 Cooperative Relaying System

For the cooperative relaying system, communication is periodic and occurs on a timescale shorter than the coherence time of the channel, but relaying causes random variations in the

channel since the set of relays is not known and changes from cycle to cycle. For this reason, parameter estimation and correction for each mode must be considered separately.

Mode 1 has "normal" synchronization and channel estimation requirements because the controller sends a large broadcast frame to all actuators. A standard preamble and reference symbols (i.e. those used in IEEE 802.11ac) can be used since the overhead is amortized over all actuators.

Mode 3 has many short individual transmissions from each sensor to the controller, so a preamble for each transmission would have too much total overhead. Instead, the controller must track all sensors' channels after system initialization using embedded reference symbols in each sensor's transmissions. This will not perform as well as using both the preamble and reference symbols, so more reference symbols are required in order to have tight enough synchronization for mode 3's narrow time slots.

For mode 2, where the controller and some number of relays simultaneously transmit a large frame, a preamble can be used since it is again amortized over many actuators' messages, but the presence of relays requires tighter time and frequency synchronization. For CDD to operate correctly, the time synchronization must be precise enough (and the cyclic prefix large enough) to have a valid portion of all received messages overlap [59]. Since all terminals relaying have different CFOs, frequency synchronization must be precise enough (and the subcarrier spacing large enough) to avoid excessive ICI. A longer preamble or more reference symbols may be needed.

Mode 4, in which remaining sensors and some number of relays simultaneously transmit small frames, cannot use a preamble, but also requires tight time and frequency synchronization because it has the constraints of both modes 2 and 3. A large number of reference symbols may be the only option. A possible alternative is using the channel estimates from modes 1 and 3, which are channel estimates of the individual terminals, to estimate the channels in modes 2 and 4. Rather than having to estimate the long relay channel "from scratch", a receiver may instead be able to estimate which transmitters are acting as relays. This may enable a reduction in the number of reference symbols, which lowers the signaling overhead.

The preamble transmitted in modes 1 and 2 enable all sensors/actuators to synchronize to the controller. The terminals that are not reached by the controller in mode 1 will have poorer timing estimates since they derive their estimates from the relays' estimates.

The estimation and correction blocks will likely have a custom design, although parts can be borrowed from either IEEE 802.11 or LTE.

4.7 Other Baseband Processing

Other important blocks in the baseband that have not been discussed yet include the symbol mapper, channel coding, retransmission handling, the equalizer, and the DSP related to multiple antennas. The symbol mapper groups multiple bits into a symbol from a complex (QAM) constellation at the transmitter, and the detector estimates which symbol was sent

at the receiver. A constellation with fewer points has a lower probability of error, but more points allows more bits to be sent per symbol. Also, lower order constellations allow looser timing and CFO estimation, which allow smaller preambles and fewer reference signals. Since CDD affects the reliability of each subcarrier in the cooperative modes, the system may need higher order constellations to have large enough data rate on the "good" subcarriers to make up for the information that cannot be decoded on the "poor" subcarriers. For this reason, using different constellations in the individual modes (1 and 3) and in the cooperative modes (2 and 4) may be beneficial. This is a complex tradeoff that must be analyzed in more detail, but is outside of the scope of this section.

Channel coding enables forward error correction, which trades off the static overhead of having redundant bits with the random overhead of retransmissions. The individual transmissions (modes 1 and 3) should have predetermined code rates so that the selected code rate does not need to be communicated before every transmission. The code rates need to be low enough to have very few nodes require retransmissions in the relaying phases. The simultaneous transmissions (modes 2 and 4) can either use an adapted code rate that allocates each remaining user equal time in the relaying phase, which changes based on the number of users requiring retransmissions, or they can use the same code rates as used in the initial transmissions. The former option leads to better performance and full utilization of the cycle time, but it requires a coding scheme that can adjust its code rate on the fly with the same information length. In other words, it must adjust its code rate by adding or removing only parity information, which changes its blocklength. Either puncturing a low-rate Turbo or LDPC code or using rateless codes can accomplish this. The latter option has a much simpler implementation because the same encoder/decoder can be used for the individual transmissions and the relayed transmissions; however, different codes must be used for modes 1/2 and 3/4 since they have different blocklengths and likely different code rates. Any class of ECC can be used, and there are many implementations of flexible encoders/decoders available that can support multiple codes with different rates and/or blocklengths.

Many codewords are often interleaved in order to make the input symbols to the decoder statistically independent and to avoid burst errors. Unfortunately, this increases the latency of the system because multiple transmissions from the same source must be received before they can be decoded. Also, with the cooperative relaying system, successful nodes do not send multiple transmissions in a cycle time, so interleaving is not practical or useful. Therefore, interleaving will not be used in the cooperative relaying system.

Retransmission handling is important for both latency and reliability. Ideally, retransmissions use hybrid automatic-repeat-request (HARQ) since it uses information from all retransmission attempts. One version of HARQ combines the soft estimates of each (re)transmission together in order to average out bad estimates. Another version requires that each retransmission have new parity symbols that are used to help decode the information symbols that were transmitted as part of the first transmission. Each relay would know which parity symbols to send in a given phase of a cycle because the system is completely scheduled. The second version has better performance than the first, but it is more complicated to implement. It works very well with both puncturing a low-rate code and with rateless codes.

When using puncturing, the parity symbols left out in the first transmission can be included in subsequent transmissions. When using rateless codes, additional parity information can be generated on the fly. LTE implements a version of this with punctured Turbo codes, but the decisions for retransmission are made in software. The time required for receiver's software to detect and notify the transmitter's software that a retransmission is required ($\geq$ 4ms) and for the scheduler to allocate a retransmission slot ($\geq$ 4ms), which occurs every time a retransmission is required, dominates LTE's latency. The cooperative relaying protocol eliminates the scheduling latency, but the error detection and notification latency remains. Therefore, all components relating to HARQ must be implemented in hardware to minimize latency, which would require custom design of the decoder, error detector, and retransmission controller. The first version could use an existing decoder, but the error detector and retransmission controller would still need to be custom. The block that combines the soft information from multiple retransmissions may need to be custom as well.

OFDM equalization is usually implemented as a simple one tap filter, but larger subcarrier spacing and the additional frequency selectivity from CDD may require more complex channel estimation and equalization for the system modes that use relays. Still, the frequency domain approach is preferable to time domain equalization because the number of taps would be large for a CDD-based system. If only one tap is required, existing basebands can support the equalization, but more complex estimation and equalization would require a custom solution.

If the controller and sensors/actuators can support multiple antennas, they should be used for diversity instead of spatial multiplexing in all system modes because reliability is the most important parameter of the system and data rates can be increased in other ways first before spatial multiplexing is required. Also, spatial multiplexing requires far more sophisticated reference signaling and channel estimation to operate correctly, especially with a distributed space-time code like CDD. Diversity can be achieved using CDD across each transmit antenna, and each receive antenna can calculate their own channel estimates based on the same preambles and reference signals that were present before. Also, the additional antennas used will not introduce more variability in the CFO or arrival times because they are controlled by the same source as an antenna that was already present in the system. Existing IEEE 802.11n/ac and LTE basebands support CDD for multiple antennas and receive diversity.

4.8 Conclusions

The PHY will need to be flexible to support the different processing choices for each communication mode and different control system requirements (information per node, latency, reliability). Depending on the implementation choices and the level of flexibility desired, many blocks can be borrowed from IEEE 802.11ac and some from LTE (AFE, symbol mapper, equalizer, MIMO DSP, parts of the estimation/correction, parts of the OFDM modulator). Custom implementations will be required for several blocks (ECC encoder/decoder,

MAC, parts of the estimation/correction), but they can easily be integrated with a standard baseband so that a single chipset can implement both standard high-data-rate transmission or low-latency, high-reliability transmissions.

Chapter 5

Low-Latency Hardware Case Study: LDPC Decoder

When implementing the system architectures discussed in Chapter 3, as many functions as possible will be implemented in hardware because handing off calculations to software can add significant latency and power, but poorly designed hardware may not be much better. The error control coding (ECC) block provides a good platform for a case study of low-latency, low-power hardware design. They are a necessary component in any low-latency, high-reliability system because they allow correction of transmission errors without the need for retransmission, and their decoders take a significant amount of the baseband processing time and power due to their complexity.

This chapter focuses on the implementation of a decoder for low-density parity-check (LDPC) codes, which are a powerful class of ECC that are widely used in modern communication systems and applicable to the system architectures of interest. In particular, a low-power decoder for the IEEE 802.11ad standard will be implemented because that standard specifies multiple short blocklength codes and a high throughput requirement. The former is of interest because of the short message lengths and variable rates used in wireless control systems. The latter is of interest because it will result in low-latency architectures due to the iterative nature of the decoding algorithm. In other words, the next codeword cannot be decoded until the decoder is finished completely with the current codeword, so a high throughput is equivalent to a low latency. In the rest of the chapter, throughput will be a focus instead of latency, but all techniques to improve throughput will result in lower latency. Lessons learned from this design can be applied to many other modern ECC decoders since they share features with LDPC decoders.

Some or all of the work in this chapter appeared in [34] and [35]. Sergey Skotnikov helped with writing the decoder RTL and with analyzing decoder performance when using approximate marginalization. Milovan Blagojević helped with the physical implementation of the decoder chip in 28nm FDSOI. Andreas Burg and Zhengya Zhang contributed to discussions on the decoder architecture, and Philippe Flatresse contributed to discussions on the decoder's physical implementation.

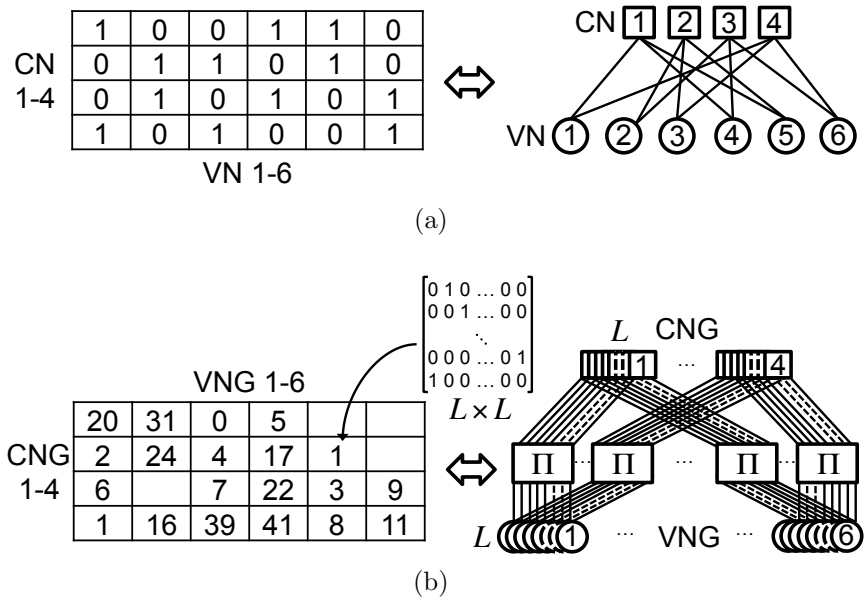


Figure 5.1: The conversion between **H** and its Tanner graph for (a) a random LDPC code and (b) for a quasi-cyclic LDPC code where the Π blocks represent a cyclic shift.

5.1 Background

5.1.1 LDPC Codes

Binary LDPC codes are defined by a sparse binary $M \times N$ parity check matrix $\mathbf{H}$, which is illustrated graphically using a Tanner graph where each of the N columns (bits) is represented by a variable node (VN) and each of the M rows (parity checks) is represented by a check node (CN). An edge connects VN i and CN j if and only if $\mathbf{H}(j,i) = 1$. Figure 5.1a illustrates this relationship. Each code has a rate R, defined as $R = (N - M)/N$.

A quasi-cyclic LDPC code has an $\mathbf{H}$ matrix composed of smaller $L \times L$ submatrices. Each submatrix is either a cyclic shift of the $L \times L$ identity matrix or an $L \times L$ all-zeros matrix [82, 83]. For this reason, $\mathbf{H}$ is often specified as an $M/L \times N/L$ matrix of the shift amounts denoted $\mathbf{H}_{\text{sub}}$ (an all-zeros matrix is denoted by a blank entry). In the Tanner graph, the L VNs corresponding to bits in the same submatrix form a VN group (VNG), and similarly the L CNs form a CN group (CNG). Each VN within VNG i has a one-to-one connection to a CN in CNG j if and only if submatrix $\mathbf{H}_{\text{sub}}(j,i)$ is not blank. If the cyclic shift for the submatrix is S, VN k in VNG i connects to CN $k + S \pmod{L}$ in CNG j. Figure 5.1b shows the relationship between a quasi-cyclic $\mathbf{H}$ and its Tanner graph.

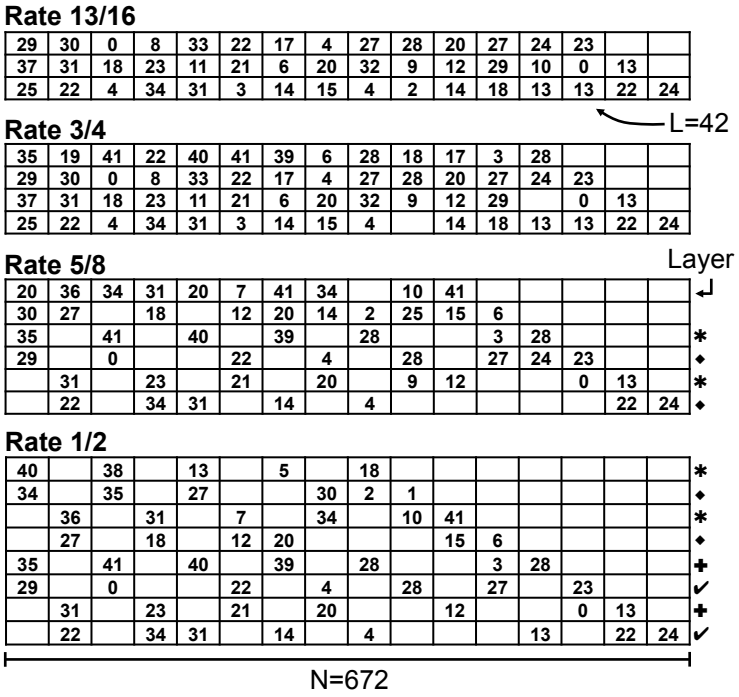


Figure 5.2: The four matrices defined in the IEEE 802.11ad standard where each numerical entry in the matrix represents a cyclically shifted 42×42 identity matrix with the shift amount given by the number and where each blank entry represents a 42×42 all-zero matrix. The symbols on the right indicate pairs of non-overlapping layers.

5.1.2 IEEE 802.11ad (WiGig) Wireless Standard

The IEEE 802.11ad standard, also known as WiGig, operates in the wide unlicensed band around 60GHz and targets very high data rate communication. To enable low power dissipation, it uses single carrier modulation with complex equalization and LDPC decoding employed in the baseband along with beamforming. IEEE 802.11ad has three classes of operation with coded throughputs of 1.5, 3, and 6Gb/s. It defines 4 irregular quasi-cyclic LDPC codes of rates 1/2, 5/8, 3/4, and 13/16 with a common blocklength of 672 bits and submatrix size of 42, as shown in Figure 5.2 [84]. A quasi-cyclic matrix $\mathbf{H}_{\text{sub}}$ is irregular if it does not have the same weight for every row and the same weight for every column, where weight is the number of non-blank submatrices in a row or column.

The matrices have another structural feature that can be exploited in the architectural design; the rate 5/8 and 1/2 codes have 2 and 4 pairs of non-overlapping submatrix layers, respectively. A highly parallelized decoder can process these layers simultaneously, reducing the number of sequentially processed layers for both codes to 4. This decreases the number of sub-iterations, lowering the decoder's operating frequency and increasing hardware utilization [34].

5.1.3 Decoding Algorithm

LDPC codes are iteratively decoded using the message-passing algorithm. This algorithm operates on the Tanner graph of the LDPC code where soft messages are passed between the VNs and CNs. There are several implementations of message-passing, but typically decoders use the min-sum variant of the sum-product algorithm in the log-likelihood domain due to its low complexity implementation [85]. To begin, each VN v_i is initialized with a prior log-likelihood ratio (LLR) $L(p_i)$ based on the corresponding channel output y_i according to

$$L(p_i) = \log \frac{\Pr(v_i = 0|y_i)}{\Pr(v_i = 1|y_i)} \tag{5.1}$$

Each VN i sends its initial value as a variable-to-check (V2C) message to each neighboring CN j.

Each CN j computes an LLR $L(c_{j\to i})$ for each neighboring VN i based on the newest V2C messages that it has received from its neighboring VNs and its parity constraint according to

$$L(c_{j\to i}) = \max\left\{\min_{i'\in\mathcal{N}(j)\setminus\{i\}} |L(v_{i'\to j}) - \beta|, 0\right\} \times \prod_{i'\in\mathcal{N}(j)\setminus\{i\}} \operatorname{sgn}\left(L(v_{i'\to j})\right) \tag{5.2}$$

where $L(v_{i'\to j})$ is the most recent V2C message that VN i' sent to CN j, $\mathcal{N}(j)\setminus\{i\}$ is the set of VNs that share an edge with CN j except VN i, and β is an empirical correction factor. These LLRs are returned to the corresponding VN as a check-to-variable (C2V) message.

Each VN i updates its decision on v_i based on $L(p_i)$ and the the most recent C2V messages according to

$$v_i = \begin{cases} 0, & \sum_{j\in\mathcal{N}(i)} L(c_{j\to i}) + L(p_i) \geq 0 \\ 1, & \text{otherwise} \end{cases} \tag{5.3}$$

where $L(c_{j'\to i})$ is the most recent C2V message that CN j' sent to VN i and $\mathcal{N}(i)$ is the set of CNs that share an edge with VN i.

The marginalized posterior information of VN i for CN j $L(v_{i\to j})$ is returned to the CNs as a V2C message for use in the next iteration and is calculated as

$$L(v_{i\to j}) = \sum_{j'\in\mathcal{N}(i)\setminus\{j\}} L(c_{j'\to i}) + L(p_i) \tag{5.4}$$

Passing C2V and V2C messages continues until the decoder reaches a stopping condition, such as the hard decisions satisfying parity or the decoder exceeding a specified number of iterations. If the former is true, decoding succeeds.

Alternative schedules for passing messages between VNs and CNs exist. The flooding schedule divides every iteration into 2 parts. First, the CNs receive updated V2C messages from all VNs, and then the VNs receive updated C2V messages from all CNs. The layered schedule divides an iteration into $2l$ parts, where l is the number of groups of CNs, or layers,

H is divided into. In sub-iteration k, all VNs send updated V2C messages to the CNs in layer k, and in sub-iteration $k+1$ those CNs send updated C2V messages to all VNs. Compared to layered decoding, flooding requires twice as many iterations to converge. It also has fewer, but more complex, computations in the CN [86, 87]. Other schedules exist, such as informed dynamic scheduling [88], but have not been commonly used so far.

5.1.4 Decoder Architectures

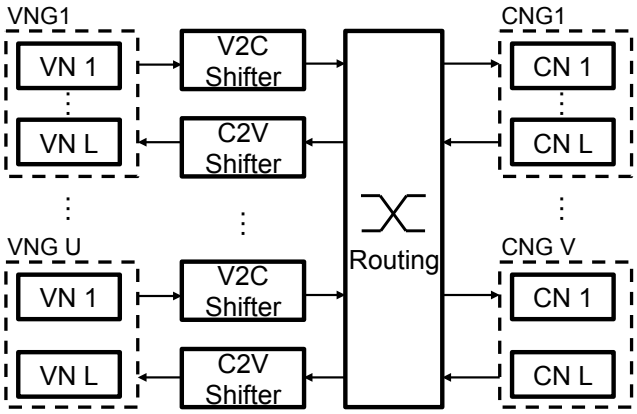
LDPC decoders that implement the min-sum algorithm have three architectural classes: fully-parallel [89], serial [90], and partially-parallel (for a quasi-cyclic **H**) [91]. For multi-Gb/s data rates, the energy-area optimal architecture is deeply-pipelined and highly-parallel because those features support a wide range of voltage-frequency scaling (VFS) [92]. Fully-parallelized architectures are not generally used because they have excessive routing overhead, which decreases their maximum clock frequency (and increases latency). Serial architectures require a large number of cycles per iteration, and cannot be run fast enough to meet the clock frequencies needed to achieve high-throughput (and low-latency). Therefore, partially-parallel architectures are used instead.

The partially-parallel architecture in [92] groups L VNs into a VNG and L CNs into a CNG, and it is parallelized by having multiple VNGs and CNGs (Figure 5.3a). The shifters implement the submatrix shifts required by the quasi-cyclic **H**, and the routing is the fixed global routing between the VNGs and CNGs. The two primary blocks are the VNs and CNs. VNs compute V2C messages by accumulating the channel prior and incoming serialized C2V messages according to (5.4). All marginalizations are performed in the VNs, which have memories for prior V2C and C2V messages (Figure 5.3b). Since the CNs do not perform marginalization, they compute C2V messages using a tree whose compare-select blocks find the first and second minima of the inputs' magnitudes and the product of the signs as required by (5.2) (Figure 5.3c).

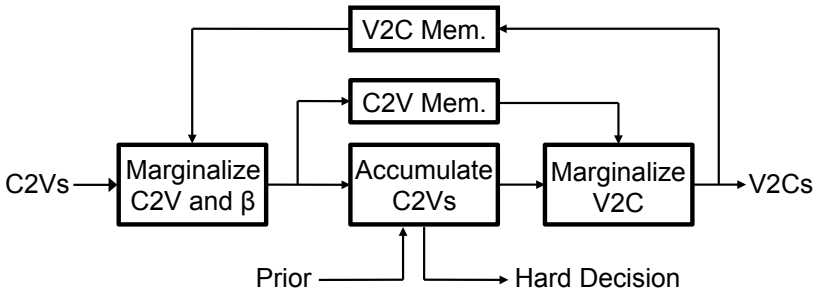
The architecture described in this Chapter uses the flooding schedule instead of the layered schedule because it requires fewer total cycles to converge for highly-parallel, deeply-pipelined decoders despite requiring more iterations to converge. This stems from the layered schedule having an inter-layer dependency instead of an inter-iteration dependency that results in a larger pipeline bubble (and more cycles per iteration) for the former (Figure 5.4).

5.2 Throughput Enhancement

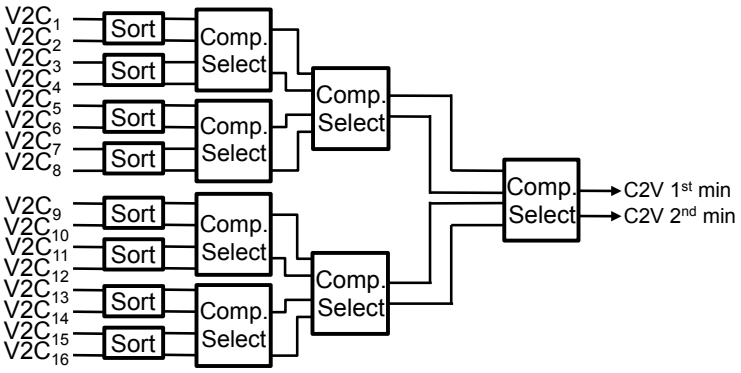
The architecture in Chapter 5.1.4 needs modifications to achieve the required throughput, flexibility, and power specifications of IEEE 802.11ad. Two techniques to accomplish this are proposed that take advantage of the **H** matrices' structure and how the architecture processes the matrix: (1) layer-merging and (2) multiple codeword processing [34].



(a)



(b)



(c)

Figure 5.3: (a) General architecture of the decoder, (b) the design of a VN, and (c) the design of a 16-input CN.

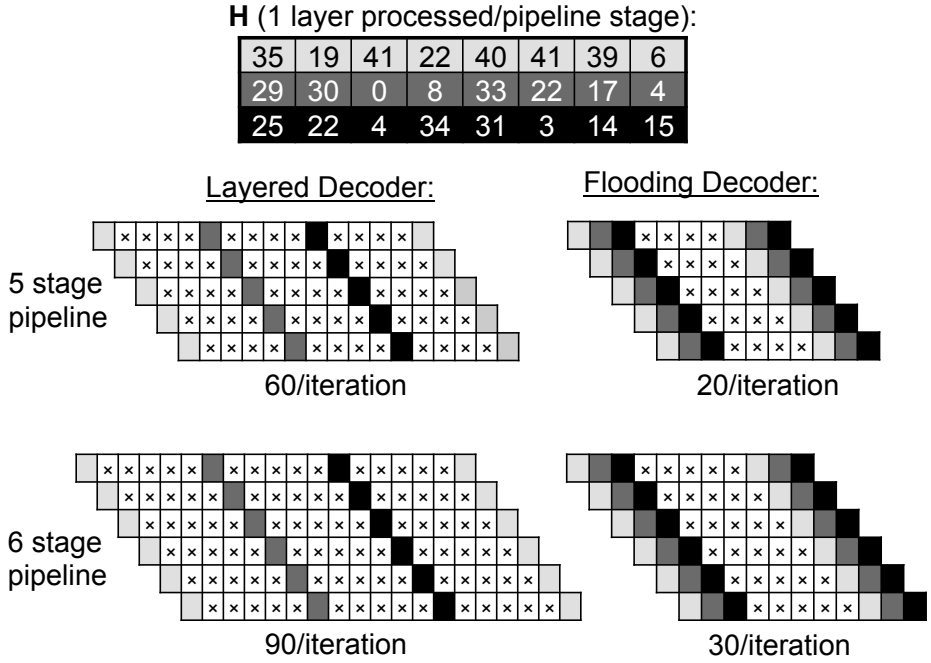


Figure 5.4: Pipeline bubbles created by the inter-iteration dependency of the flooding schedule and inter-layer dependency of the layered schedule for a decoder with a 5 or 6 stage pipeline that can process one layer of **H** in a pipeline stage. The number of wasted cycles (indicated with an $\times$) per iteration is shown below each example.

5.2.1 Layer-Merging

Reconfigurable decoders can require more cycles to decode lower rate codes because the matrix has more layers and since they have longer critical paths due to the extra hardware needed for flexible routing. A low-overhead method for reconfiguration is a key requirement to minimize power. Efficient reconfiguration for the IEEE 802.11ad standard is possible because the rate 5/8 and 1/2 **H** matrices have 2 and 4 pairs of non-overlapping layers, respectively, as indicated in Figure 5.2. Additionally, each layer in the pair has at most a weight of 8, which is half their maximum possible weight.

Instead of processing the pair of layers sequentially, they can be processed simultaneously using layer-merging. In terms of the decoder's operation, this is best visualized as combining the pairs of non-overlapping layers into a single merged layer, as shown in Figure 5.5 for the rate 1/2 code. This merging reduces the number of layers from 8 to 4 for the rate 1/2 code, and from 6 to 4 for the rate 5/8 code. This gives the rate 1/2, 5/8, and 3/4 codes the same number of layers, reducing the worst-case number of cycles per iteration by half. In addition, layer-merging better utilizes the hardware since there are fewer columns of weight zero in each layer.

To implement layer-merging, the CNs require modification because the V2C messages

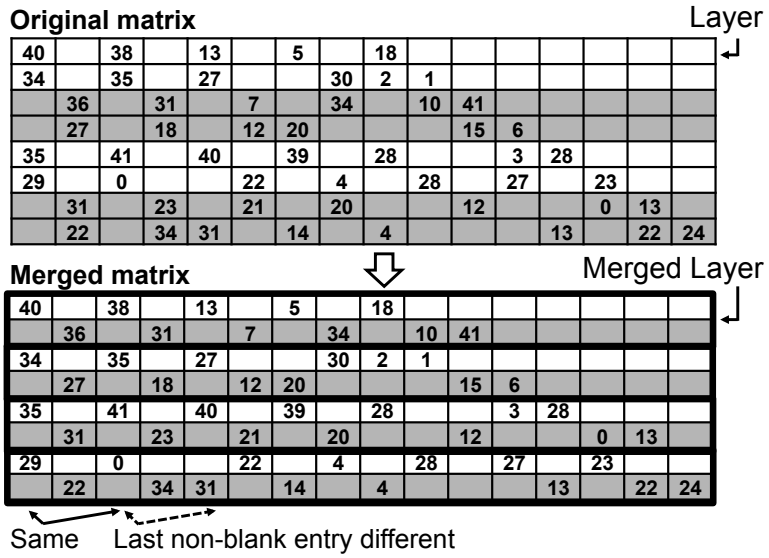


Figure 5.5: The original and merged IEEE 802.11ad rate 1/2 matrix. The rate 5/8 matrix has two pairs of non-overlapping layers and two near full-weight layers, so its merged matrix has both non-merged and merged layers. A given column of the merged matrix does not have the same pattern of non-blank entries, indicating that the routing must be flexible between sub-iterations. The rate 3/4 and 13/16 codes do not have non-overlapping layers, so their merged matrices are the same as their original matrices.

from the two non-overlapping layers must be kept separate when finding their minima. Since the CN is a tree and the weight of each non-overlapping layer is at most half of the maximum, the V2C messages can be processed separately by routing those from the first (second) layer to the top-half (bottom-half) of the tree, taking the output of the top-half (bottom-half) of the tree at the penultimate stage of the tree, and selecting the top-half (bottom-half) result for the VNGs belonging to the first (second) layer. When processing a non-merged layer with weight greater than half the maximum, the ordering of the inputs does not matter and the output from the entire tree is used. Figure 5.6 shows an example of this process, and Figure 5.7 shows the reconfigurable CN that acts as either a single full-size CN or two half-size CNs. Since the pairs of non-overlapping layers do not share any VNs, no read-before-write conflicts occur, so no changes to the VNs are required.

5.2.2 Multiple Codeword Processing

Due to the inter-iteration dependency of the flooding schedule, a pipeline bubble exists between accumulating all C2V messages from the current iteration in the last stage of the pipeline and computing new V2C messages for the next iteration in the first stage of the pipeline (Figure 5.4). During this time, the other stages of the pipeline are flushed and do

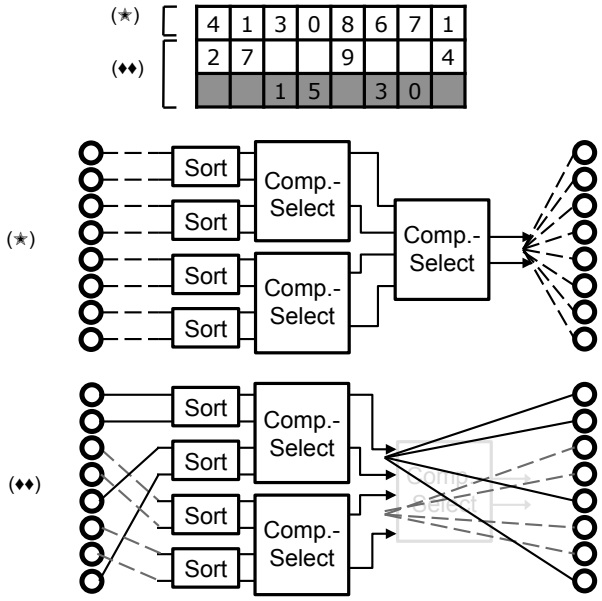


Figure 5.6: Routing required for processing both non-merged and merged layers for an example matrix. When processing the weight-8 layer (★), the CN takes the V2C messages in any order, uses the entire tree to find the overall first and second minima, and sends all VNGs the same C2V message. When processing the two weight-4 layers (♦♦), the CN takes the V2C messages from the first layer into its top four inputs and those from the second layer into its bottom four inputs, takes the outputs one stage early to keep the first and second layer minima separate, and sends the VNGs corresponding to the first layer (1,2,5,8) the results from the top-half of the tree and sends the VNGs corresponding to the second layer (3,4,6,7) the results from the bottom-half of the tree. If an **H** matrix has both non-merged and merged layers, the CN must be able to shift between the two modes on the fly.

not contribute useful computations. However, up to $\lfloor (p-1)/s \rfloor$ additional codewords can be processed during this time since there are no feedback or feedforward paths, where p is the number of pipeline stages and s is the number of sub-iterations for the worst-case matrix. This increases throughput by a factor equal to the number of additional codewords processed, but it requires additional memory for channel priors and for C2V message accumulation. No extra current or prior message memories are required since the pipeline registers hold current messages and the prior message memories in the VNs can be overwritten after use.

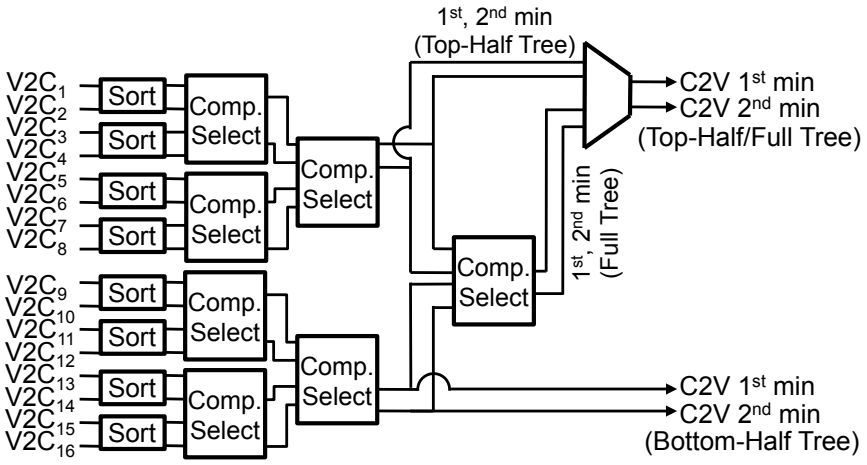


Figure 5.7: Design of granular check nodes. The XOR tree is included in the compare-select blocks.

5.3 LDPC Decoder Design

5.3.1 Architecture Optimization

Since the IEEE 802.11ad **H** matrices are quasi-cyclic, the partially-parallel architecture from Chapter 5.1 can be used with the optimizations from Chapter 5.2, but the optimal parallelization factors of the VNGs and CNGs, pipeline depth, and number of simultaneously processed codewords must be selected. An exhaustive search for the optimal architecture is unrealistic, but a sensitivity-based heuristic search can yield a near-optimal design [34]. The steps are as follows:

1. List the combinations of design parameters, or architectures, to explore, but leave those out that will obviously require high operating frequencies f_{op} to achieve the specified throughput T, such as low parallelizations or very deep pipelines.

2. For each architecture and **H** combination, find the number of cycles C that the pipeline takes to complete all calculations for an iteration of a single codeword. This is insensitive to the logic in each stage, but architectural optimizations must be considered, such as layer-merging.

3. For each architecture/**H** combination, compute $f_{\text{op}} = (T \cdot I_{\text{avg}} \cdot C)/(N \cdot F)$, where I_{avg} is the average number of iterations when decoding **H** at the E_b/N_0 of interest, N is the code's blocklength, and F is the number of codewords processed simultaneously. Select the largest f_{op} for each architecture.

4. Eliminate architectures with a large f_{op} since they cannot use significant VFS.

5. Design and synthesize the blocks of the remaining architectures to find their power and area.

6. Choose an architecture with balanced area and power since it has a near optimal tradeoff between wiring overhead and logic power (initial synthesis does not take wiring overhead into account).

For the IEEE 802.11ad specifications, the heuristic search returned an architecture that has fully parallelized VNGs, serialized CNGs, and a pipeline depth of 5 that simultaneously processes 2 codewords.

5.3.2 Architecture

Figure 5.8 shows a block diagram of the decoder. There are 672 VNs arranged into 16 VNGs and 42 CNs grouped into 1 CNG, which means each stage of the decoder operates on a full layer of the merged **H** matrix. The C2V and V2C shifters implement the forward and reverse cyclic shifts required by the quasi-cyclic structure of the **H**s, the shuffler implements global routing between the VNGs and CNGs and, along with the post-CN MUXs, enable layer-merging.

Since the decoder uses a serialized CNG and the matrices have at most 4 layers after layer-merging, the CNs in the CNG are time-multiplexed to act as 4 different CNs in each iteration. For example, in the first cycle of an iteration, each VN sends a single message that is marginalized for the CNs in the first layer. When the CNG receives these V2C messages a cycle later, it computes the C2V messages from the first layer of CNs and sends them back to the VNs in the third cycle. In the second cycle, each VN sends another single message that is marginalized for the CNs in the second layer, so the CNG can compute the C2V messages from the CNs of the second layer in the third cycle and send it to the VNs in the fourth cycle. This continues for all the layers in the merged matrix, with serial messages being passed back and forth between VNs and CNs. The shifters, shuffler, and post-CN MUXs ensure that messages are routed correctly between nodes.

5.3.3 Functional Blocks

Variable Node

The VN implements (5.4) and performs both the CN and VN marginalizations, which keeps all memory within the VNs. At the start of decoding, the VN loads its $L(p_i)$ into the appropriate prior and accumulation registers. Over the next 4 cycles (3 for the rate 13/16 code), it sends V2C messages to its neighboring CNs and locally stores them in a shift register. When unmarginalized C2V messages return serially, the VN uses the stored V2C message to marginalize the magnitude and sign of the C2V message. After β correction, the VN serially accumulates the marginalized C2V messages and the prior and locally stores them in a second shift register. After accumulation, the VN calculates the new V2C messages

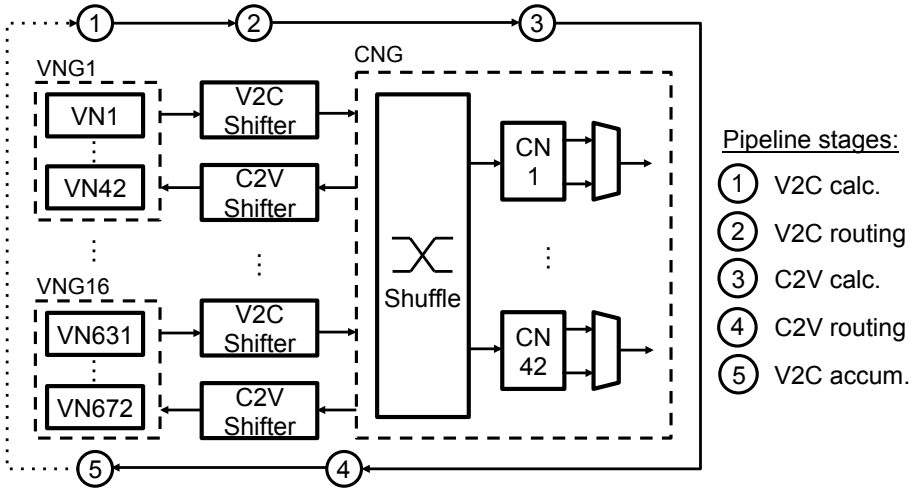


Figure 5.8: Block diagram of the decoder architecture with pipeline stage locations and descriptions.

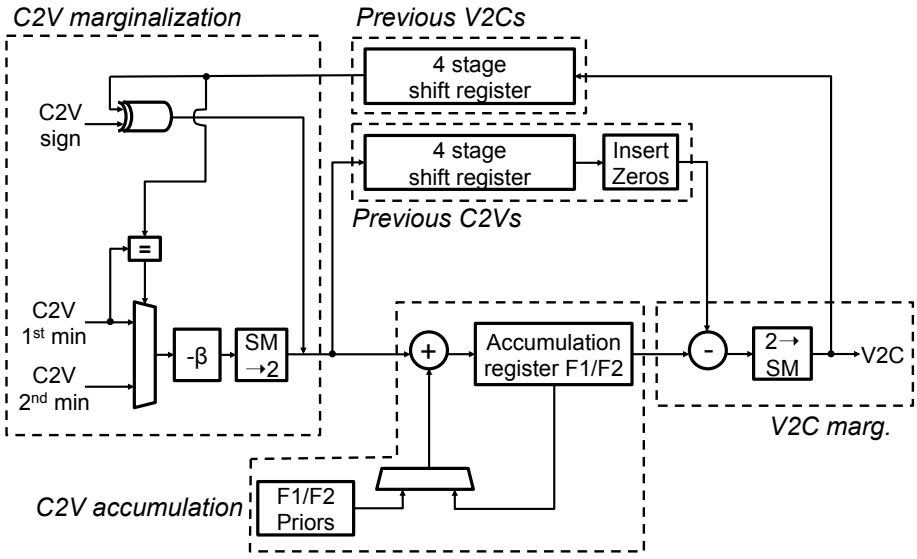


Figure 5.9: Design of the variable node (SM→2 is a sign-magnitude to 2's complement conversion and 2→SM is the reverse).

by subtracting the individually stored C2V messages from the accumulated value. The sign bit of the accumulated value is used as the hard decision and is sent to the CN to detect early convergence. Figure 5.9 shows the design of the VN.

Shuffler

The first half of the shuffler performs global routing by connecting the 42 outputs of each VNG's shifter to the appropriate nominal input of a CN in the CNG (output i of shifter j goes to input j of CN i). The second half selects which VNG's V2C message goes to the top-half or bottom-half of each CN (rearranging the nominal ordering of inputs to the CN). The second half can be simplified significantly (as compared to e.g. [34]) for the $\mathbf{H}$ matrices of interest by observing that a CN does not care about the order of its inputs if it is acting as a 16-input CN and that the interleaving pattern of the merged layers is nearly the same for each merged layer for all code rates (Figure 5.2) [93]. Using these insights, much of the routing is accomplished by shuffling fixed wires, eliminating many of the long, complex global routes and multiplexers required in [34]. Since the path from the output of the VNs to the input of the CNs is the critical path, this optimization allows additional VFS to reduce overall power consumption and mitigates the overhead of reconfiguration. Figure 5.10a illustrates how the second half of the shuffler fits into the overall CN architecture and shows the partially fixed shuffling of wires.

Check Node

Since the VN performs the marginalization in (5.2) and the shuffler orders the input V2C messages, the CN only needs to find the first and second minima of the incoming V2C message magnitudes and the product of the V2C message signs. As described in Chapter 5.1, the CN computes the former with a compare-select tree. A sorting block at the beginning arranges pairs of inputs in ascending order, and subsequent stages of the tree select the minimum 2 out of 4 inputs. A separate XOR tree finds the product of the signs. When processing non-merged layers, the output of the entire tree is used. When processing merged layers, the outputs of the top- and bottom-half of the tree are taken at the penultimate stage. Figure 5.10b shows how this function is implemented. The CN contains an additional XOR tree to compute the syndrome based on the VNs' hard decisions. This is used for early termination, and detects convergence with a delay of one iteration.

Post-CN MUXs

For each VNG, the post-CN MUXs select whether to send the top-half/full tree's or bottom-half tree's result. This function is implemented with 16 multiplexers after each CN (one for each VNG), which are controlled by a small memory. Figure 5.10c shows where the post-CN MUXs fit into the overall CN architecture and their implementation.

5.3.4 Pipeline

Each iteration takes 3 sub-iterations for the rate 13/16 code and 4 for the rest, one for each time-multiplexed use of the CNs. The pipeline stages are as follows: (1) the VNs output V2C messages and shifters reorder them; (2) V2C messages enter the global wiring to the

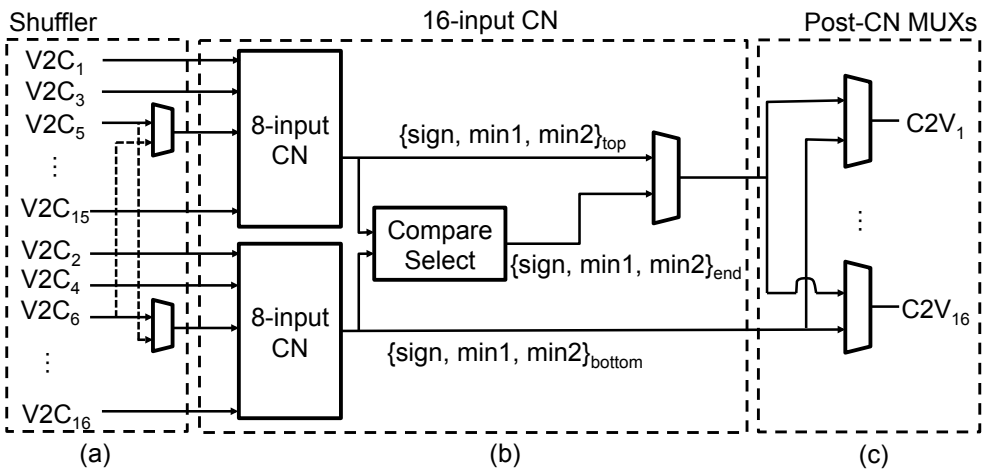


Figure 5.10: Design and connection between the (a) shuffler, (b) check node, and (c) post-CN MUXs.

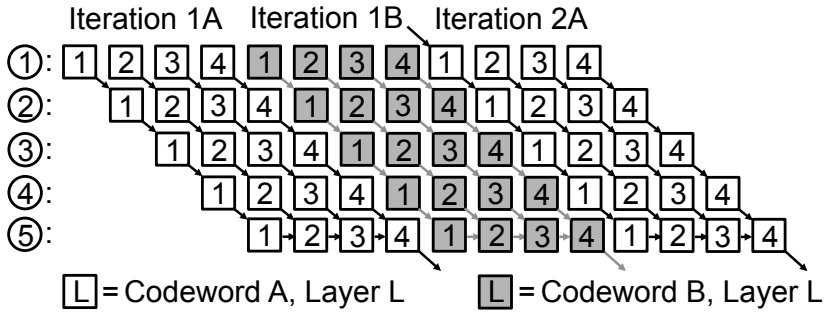


Figure 5.11: Pipeline diagram showing the simultaneous decoding of two codewords (in the case of the rate 13/16 code, there is a 1 cycle bubble since there are only 3 layers in its $\mathbf{H}$).

CN and the shuffler; (3) CNs compute C2V messages and post-CN MUXs select the correct outputs for each VNG; (4) C2V messages are routed to the VNGs across global wires; (5) VNs accumulate the serial C2V messages over 3 or 4 cycles.

The accumulation in stage 5 would normally cause a 4 cycle bubble in the pipeline due to the flooding schedule's dependency between accumulating all C2V messages and computing the next V2C messages. With 5 pipeline stages, the bubble is large enough to process a second codeword, which reduces the bubble to 1 cycle for the rate 13/16 code and removes it for the other rates. Figure 5.11 shows the pipeline diagram of the decoder.

5.4 Power Optimization

To meet power consumption requirements for wireless applications, the quantization throughout the decoder have been optimized to reduce wiring overhead and memory. These optimizations trade off power and reliability and must be jointly chosen with the requirements of the application. In addition, forward-biasing has been used to optimally trade off active and leakage power. None of these techniques affect the throughput significantly.

5.4.1 Message Quantization Optimization

The quantization of the V2C and C2V messages significantly impacts the power and area of the decoder not only because of the extra hardware required for each extra bit of precision, but also because of the additional global routing overhead. Simulations are used to determine the minimum V2C and C2V message wordlengths and the corresponding optimal β factor from (5.2) that results in acceptable bit-error rate (BER) performance for all codes. A floating-point and fixed-point min-sum LDPC simulator was implemented in C++, and each code in the IEEE 802.11ad standard was simulated around its waterfall region with a maximum of 15 iterations for a range of quantizations and βs. The results of the simulations for quantizations between 4 and 6 bits with their optimal β are shown in Figure 5.12 for the rate 13/16 and rate 1/2 codes. Based on the simulation results for each code, quantizations with four integer bits or less and any number of fractional bits cause a significant rise in the error floor, and quantizations with at least five integer bits and one fractional bit have negligible loss. Q5.0 (five total bits with zero fractional bits) with $\beta = 1$ has negligible implementation loss for the rate 13/16 and rate 3/4 codes, a 0.1dB loss for the rate 5/8 code, and a 0.2dB loss for the rate 1/2 code. Q5.0 was chosen since it has acceptable performance for all code rates and has significantly reduced power and complexity compared to Q6.1. Figure 5.13 shows the BER and average number of iterations of each code using Q5.0.

5.4.2 Reduced Marginalization Memory Precision

Memory dominates the power consumption of LDPC decoders, and the VNs and pipeline registers contain a majority of this decoder's memory. The number of pipeline stages and the number of registers per stage have already been optimized during the architecture optimization and wordlength optimization. Therefore, any further optimizations to the memory size must be done within the VNs. The largest number of flip-flops within the VN is contained in the shift registers that store C2V and V2C messages for marginalization of incoming or outgoing messages. Reducing the number of bits in these two blocks has a considerable effect on the total power since this architecture has 672 VNs and each shift register has a depth of 4 and width of 5 bits.

To ensure that this does not significantly affect the error-correcting performance of the decoder and to find the bits that should be removed, extensive simulations were performed

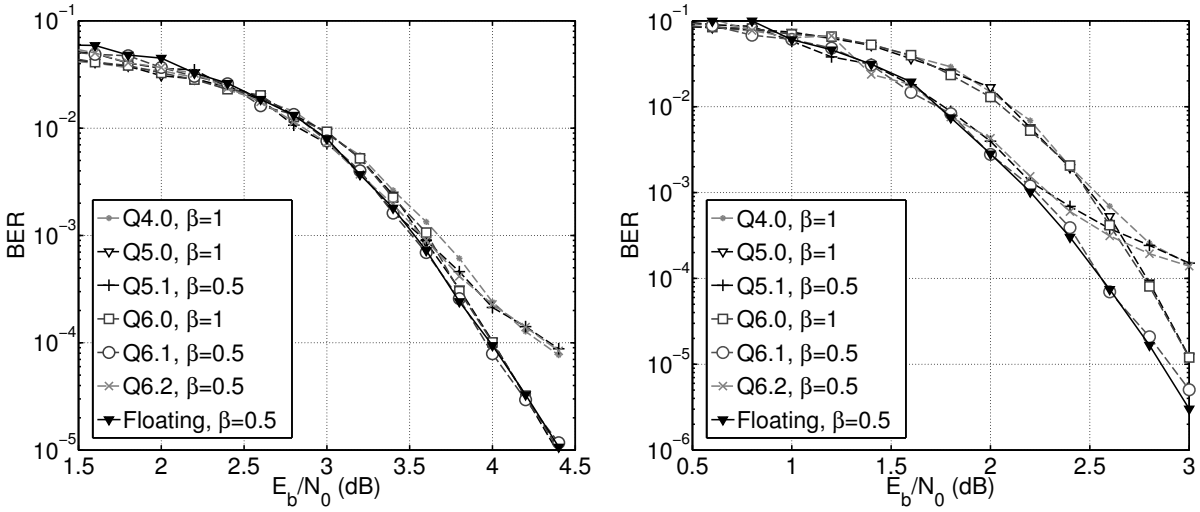


Figure 5.12: Min-sum algorithm performance for different quantizations, including floating point, with their optimal β values for the (a) (best-case) rate 13/16 code and (b) (worst-case) rate 1/2 code. Here, Q$X.Y$ is interpreted as X total bits with Y fractional bits and $X - Y$ integer bits.

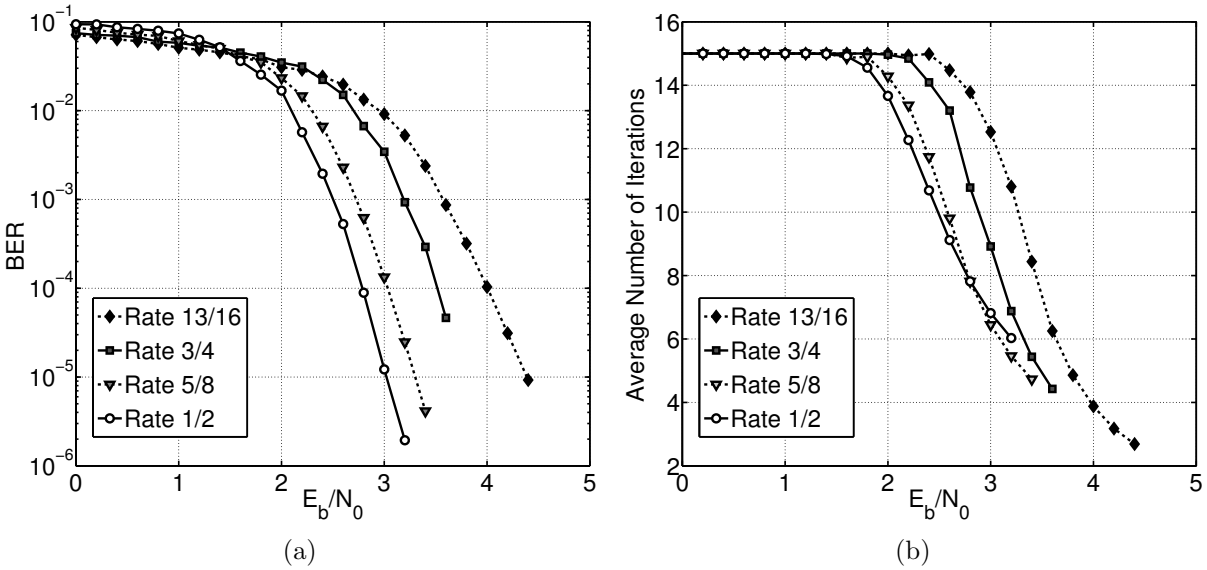


Figure 5.13: Performance of the min-sum decoding algorithm for all code rates with a 5-bit quantization and with $\beta = 1$: (a) BER, (b) average number of iterations.

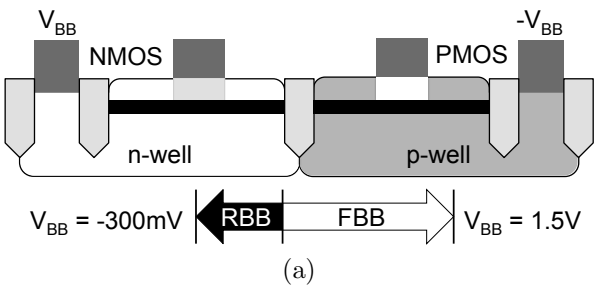
where subsets of the bits of stored messages were used for marginalization. The simulations showed that 2 bits of precision could be removed from both the V2C and C2V shift registers with less than a 0.1dB loss in BER performance. From the stored C2V messages, the 2 least significant bits are removed, and from the stored V2C messages, the least significant and the most significant magnitude bits are removed. The loss in performance can be recovered by increasing the maximum number of iterations by 5, which increases the average number of iterations by less than 1% at E_b/N_0 values of interest. This decreases the decoder's power by 15% based on place-and-route results, and this technique can be applied to any LDPC decoder architecture and code structure [35]. However, the error floor rises by an order of magnitude in the worst-case from 10^{-8} to 10^{-7}, but this is not enough to affect performance for the targeted wireless applications. The increase in error floor is much less sensitive to reducing marginalization precision than reducing wordlength, and the power savings are similar.

5.4.3 Back-Biasing Using FDSOI Devices

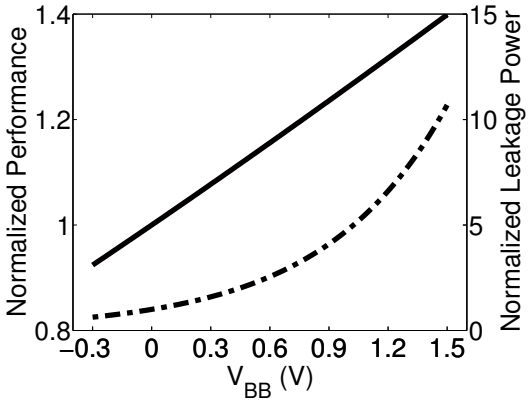
Back-biasing allows the tradeoff between active and leakage power to be optimized by tuning the n-/p-well bias voltages and the supply voltage [94]. One possibility for implementing back-biasing is using bulk CMOS devices with a triple-well process option, but the body effect in bulk CMOS has been steadily diminishing in each process node [95] and the forward-biasing range is limited. FDSOI devices offer a better alternative since they inherently allow a large range of forward- or reverse-biasing, depending on the orientation of their wells, and they have a strong body effect.

At high throughputs, the percentage of the total decoder power due to leakage is lower than optimal. Therefore, forward-biasing paired with voltage scaling would help to decrease total power. A standard well orientation, which has an p-well under NMOS devices and a n-well under PMOS devices, allows forward bias voltages up to 300mV before forward-biasing PN junctions between the wells. Flip-well FDSOI devices, which have an n-well under the NMOS devices and a p-well under the PMOS devices [96], allows forward bias voltages up to 1.5V on each well (Figure 5.14a). For this reason, flip-well devices are used to get the largest benefit from the back-biasing at high throughputs. Figure 5.14b shows the significant body effect in these devices for a 28nm process.

Back-biasing requires two extra supply rails for the n- and p-well bias voltages, but their overhead is minimal because the devices do not draw significant current through their wells, so their power straps can be sparse and thin. Figure 5.15 shows the floorplan of the decoder's power grid. Since there are no macros and a single voltage domain is used in this architecture, implementing back-biasing requires only simple changes in the back-end flow, and verification is not complicated.

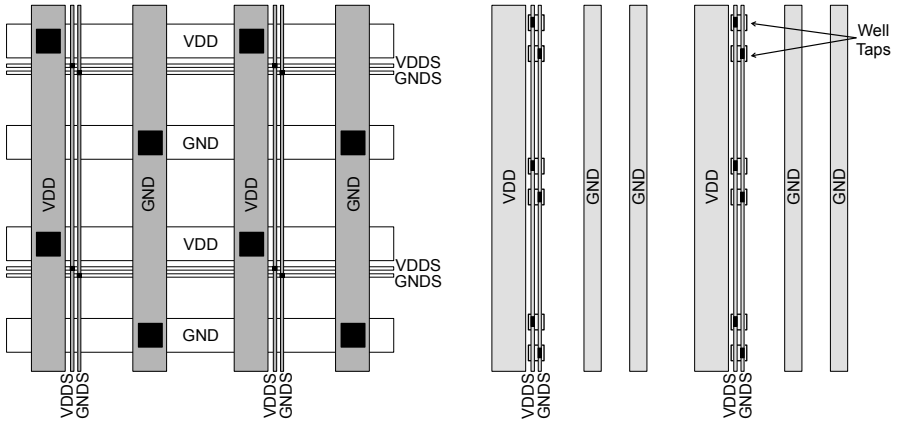


(a)



(b)

Figure 5.14: Details on the 28nm flip-well FDSOI devices: (a) a cross-sectional view of the devices where the body-biasing terminals are V_{BB} and $-V_{BB}$; (b) the power-performance tradeoff for back-biasing, highlighting the strong body effect in these devices.



(a) (b)

Figure 5.15: Floorplan of the decoder's four supply rails (V_{DD} and GND for the core and V_{DDS} and V_{GNDS} for back-biasing) (a) for the upper metal layers and (b) for the active layers through M3 (standard cell straps not shown).

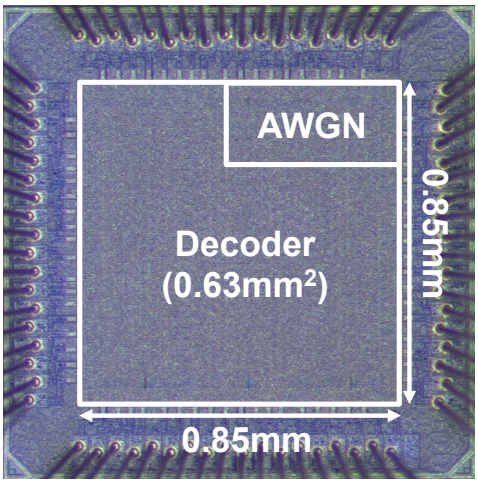


Figure 5.16: Die micrograph. The core area and test structures are labeled.

Table 5.1: Technology and chip summary.

Technology	28nm UTBB FDSOI
Transistors	Flip-well (LVT), $L = 24$nm
Die Area (mm^2)	2.56 (1.6mm×1.6mm)
Pad Count	72
V_{DD} Range (V)	0.41 – 1.07
V_{BB} Range (V)	−0.3 – 0.8

5.5 Chip Implementation

The LDPC decoder is implemented in a 28nm UTBB FDSOI technology and occupies 0.72mm^2. The decoder core area is 0.63mm^2, and the remaining 0.09mm^2 is dedicated to on-chip AWGN noise generators and decoding statistics collectors. Figure 5.16 shows a micrograph of the decoder with the core and testing areas annotated, and Table 5.1 gives a summary of the decoder ASIC's key characteristics.

5.5.1 Chip Testing Setup

The decoder supports two testing modes: manual and automated. In the manual mode, a received vector is scanned in to the chip, decoding starts, and the hard decision output is scanned out. This mode is most useful for confirming functionality and debugging the chip. In the automated mode, the decoder takes inputs from the on-chip AWGN generators at full-speed, and the decoding outputs are checked on-chip for errors. Each AWGN generator creates a $\mathcal{N}(0, 1)$ random variable using the Box-Muller algorithm, and a cascaded multiplier sets the variance according to the desired signal-to-noise ratio (SNR) assuming the signal

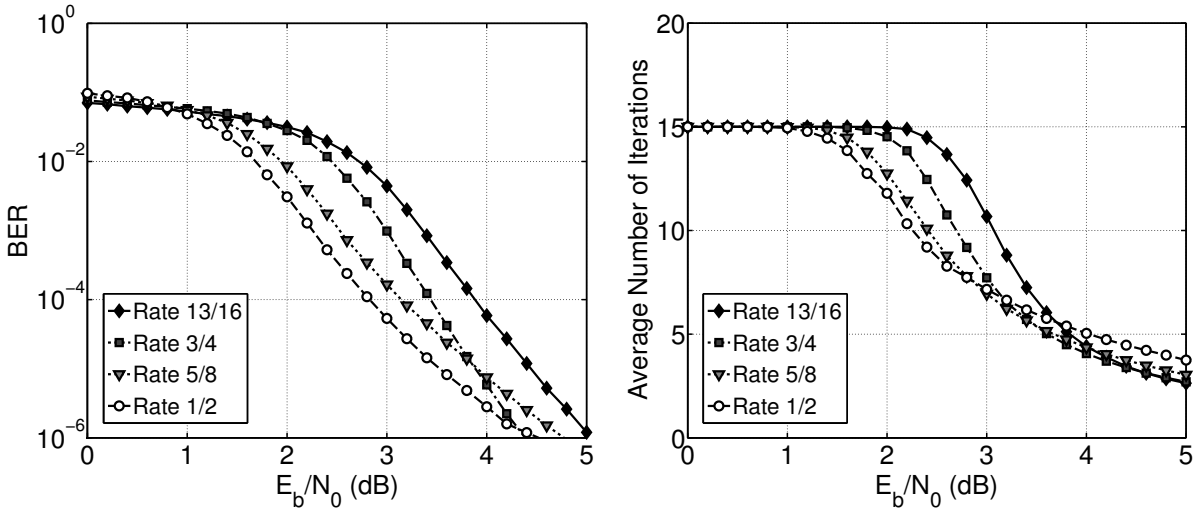
power is 1. The all-zeros codeword is assumed, so any ones that appear in the decoded output are errors. In addition, an early termination option is available that terminates decoding when all parity constraints are satisfied. The number of bit errors, codeword errors, iterations, and codewords processed are accumulated so that the BER, block error rate, and average number of iterations can be calculated. The SNR is swept across multiple points and sufficient error statistics are collected to create BER vs. SNR waterfall curves. Each point is run several times to ensure that the error statistics do not change across repeated trials.

The chip connects via a socket to a daughter card, which in turn connects to the Opal Kelly Shuttle-LX1 via an FMC connector. The Opal Kelly has a Xilinx Spartan-6 FPGA that both provides a digital interface to the decoder and interfaces to a host computer over a USB3 connection. The host computer sends configuration information and commands through the FPGA to the chip, and the chip returns decoding statistics or hard decisions to the host through the FPGA. An external, single-ended clock provides the reference for both the chip and the Opal Kelly. An external reference is used for the core supply, and a separate reference is used for each of the back-bias supplies. This allows individual tuning of the back-biasing supplies, although the same magnitude voltage (but opposite in sign) was used in testing.

5.5.2 Measurement Results

Figure 5.17 shows the BER performance and the average number of decoding iterations as a function of the channel conditions for all codes. The decoder achieves a BER of less than 10^{-6} in the waterfall region for all code rates defined in the standard. For a given BER level, the rate 1/2 code takes the most iterations to converge, which means the decoder will consume the most power when using that code.

To determine the worst-case power consumption of the decoder in a typical operating environment, the decoder was configured to use the rate 1/2 code at an E_b/N_0 of 5.0dB with early termination enabled. For these conditions, the decoder requires an average of 3.75 iterations to converge and meets the BER specifications of IEEE 802.11ad. Using the optimal core supply voltage, the decoder consumes 6.3mW, 14.4mW, and 41mW for throughputs of 1.5Gb/s, 3Gb/s, and 6Gb/s, respectively. The power can be further decreased by using the back-biasing capability of the flip-well FDSOI devices [97], where the back-bias supplies are generated off-chip. Using the optimal back-bias and core supply voltages, the power can be decreased by 5.3% and 11% for the 3Gb/s and 6Gb/s throughputs, respectively (Figure 5.18). The reason for the moderate decrease in power when using back-biasing is that the thresholds are close to optimal for this application, but, if the throughput required is higher, the savings would be larger. The decoder has efficiencies of 8.2 pJ/bit, 9.1 pJ/bit, and 12.7pJ/bit for the 1.5Gb/s, 3Gb/s, and 6Gb/s throughputs, which corresponds to normalized efficiencies of 1.1pJ/bit/iteration, 1.3pJ/bit/iteration, and 1.8pJ/bit/iteration. Additionally, the decoder consumes 3.0mW at a throughput of 0.7Gb/s at 15MHz with a core supply of 0.48V and no back-bias and consumes 179.9mW at a throughput of 12Gb/s at 260MHz with a core supply



(a) (b)

Figure 5.17: (a) Measured BER for all code rates with 5b quantization and reduced marginalization memory size of 3b. The error floor occurs after $E_b/N_0 = 5.0$dB, and is between 10^{-7} and 10^{-8} depending on the code rate. (b) The corresponding average number of iterations.

of 1.07V and a back-bias voltage of 0.8V. When operating at a throughput of 6Gb/s, the decoder has a maximum latency under 1μs. The latency scales linearly with throughput, but can be reduced by using fewer iterations or using the higher code rates.

5.5.3 Comparison with State-of-the-Art

Table 5.2 compares this decoder to other state-of-the-art high-throughput decoder designs. The power and energy efficiency in terms of pJ/b/iteration is reported for each decoder over a range of operating conditions to show each decoder's throughput vs. power and efficiency tradeoff. The energy efficiency gives a more fair comparison because it normalizes to the number of iterations each decoder uses per codeword. Area was not normalized for technology because the decoders are routing limited, but the area of the proposed decoder is compared to [98] and [99] when implemented in 65nm CMOS. The power has not been normalized for technology since the memory power for the other decoders does not scale in the same way as logic since the supply voltage would stay near the same value. Also, back-biasing could not be applied to the entire design for those decoders since the memories could not use it. For a given throughput, all the decoders will have a similar latency, but the proposed decoder can achieve the same latency at lower clock frequencies and lower power.

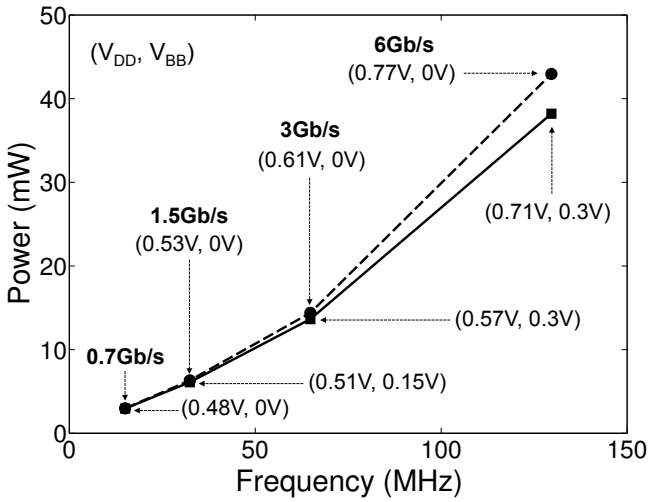


Figure 5.18: Measured power versus frequency at $E_b/N_0 = 5.0$dB with no back-biasing (dashed line) and at optimal back-biasing (solid line) of the flip-well (LVT) FDSOI devices when decoding the rate 1/2 code. The power consumption is significantly smaller when decoding the higher code rates due to fewer decoding iterations required to converge at the same BER level.

5.6 Conclusions

A low-power, high-throughput (and low-latency) LDPC decoder that is compatible with all LDPC codes and throughputs specified by the IEEE 802.11ad standard is designed. A near-optimal highly parallel architecture with a deep, efficient pipeline is used to achieve a high-throughput at the nominal supply voltage. Processing two non-overlapping layers and decoding two codewords simultaneously increases throughput by up to a factor of four for the worst case code. The high throughput is traded for power savings using voltage-frequency scaling. To reduce power further, the marginalization memory precision is reduced to lower the memory size and back-biasing is applied to optimize the active and leakage power tradeoff. All memory is implemented in flip-flops, which keeps design complexity low and allows the entire design to be back-biased. With voltage-frequency scaling and optimal back-biasing, the decoder only consumes 6.2mW at 1.5Gb/s and 38.1mW at 6Gb/s, which corresponds to energy efficiencies of 1.1pJ/bit/iteration and 1.8pJ/bit/iteration, respectively. The decoder has a maximum latency below 1μs at a throughput of 6Gb/s, which is sufficient for industrial control systems requiring cycle times on the order of milliseconds. Since the decoder can run at 12Gb/s, the latency can be cut in half if needed.

Reduced marginalization precision can be used in any LDPC decoder to reduce memory size and power, regardless of the chosen architecture or code structure. It enables a softer tradeoff between both power and area versus performance than reducing the decoder wordlength, and it retains almost all of the power savings.

Table 5.2: Comparison table for the LDPC decoder.

	This Work				**JSSC'14 [98]**				**JSSC'12 [99]**		**ASSCC'11 [100]**
Technology	28nm FDSOI				65nm bulk				65nm bulk		65nm bulk
Standard	IEEE 802.11ad				IEEE 802.11ad				IEEE 802.15.3c		IEEE 802.16e
Blocklength	672				672				672		576-2304
Code rates	1/2, 5/8, 3/4, 13/16				1/2				1/2, 5/8, 3/4, 7/8		1/2, 2/3, 3/4, 5/6
Decoding schedule	Flooding				Flooding				Layered		Layered
Core area (mm^2)*	0.63				1.6				1.56		3.36
Throughput (Gb/s)	1.5	3	6	12	1.5	3	6	9	3.3	5.79	1.056
Core supply (V)	0.5	0.6	0.7	1.1	0.5	0.6	0.9	1.2	1.0	1.0	1.2
Memory supply (V)	-	-	-	-	0.8	0.9	1.1	1.3	-	-	-
Back bias supply (V)	0.2	0.3	0.3	0.8	-	-	-	-	-	-	-
Avg. Iterations	3.75				10				5		10
Frequency (MHz)	32	65	130	260	90	180	360	540	197	197	110
Power (mW)†	6.2	14	38	180	38	106	374	783	320	400	115
Energy Efficiency (pJ/bit/iteration)	1.1	1.3	1.8	4.1	2.5	3.5	6.2	8.6	N/A	12.5‡	10.9

* Area not normalized because decoders are routing limited.

† Power consumption is for the rate 1/2 code at a BER of 1e-6 to 1e-7.

‡ For the rate 7/8 code, which will have higher energy efficiency than the rate 1/2 code.

Chapter 6

Conclusion

A paradigm shift is approaching in the world of wireless communication. The number of wireless devices will soon surpass the number of people on the planet, and this rate of growth shows no signs of slowing. With the advent of the smart home, the smart grid, and the IoT, most devices will communicate with each other instead of with people and will form a closed-loop system that both senses and acts on the environment. This will require low-latency and high-reliability communication at moderate data rates as opposed to the high data rates that current people-centric networks provide.

Currently, high-performance industrial control is one of the few applications that has similar requirements to the IoT of the future, so we can use it to judge the state-of-the-art and any solutions developed for the IoT. Unfortunately, all current high-performance control systems use wired networks because no current wireless networks can meet their stringent latency and reliability specifications. In order to support current control systems and IoT applications of the future, a wireless system architecture must be developed that can meet control system specifications and that can scale to networks with even larger number of nodes in order to support the closed-loop IoT applications of the future.

6.1 Key Contributions

This work has contributed to developing such a wireless system architecture in three main ways: (1) defining and analyzing a low-latency, high-reliability wireless communication system using cooperative relaying, (2) discussing a PHY layer implementation that differs minimally from that for current standards, and (3) providing a case study of low-latency hardware in the form of an LDPC decoder.

For the system architecture design, this work developed a model for low-latency high-reliability closed-loop systems. An initial wireless system architecture was developed that addresses the issues current WLAN and cellular systems have with supporting low-latency and high-reliability operation for a large number of nodes, particularly those involving multiple access and retransmissions. The initial architecture's achievable latency is a strong

function of the available diversity, so it requires a scheme to generate the diversity in a low-latency manner without relying heavily on the channels provided by nature. To generate this diversity without significantly increasing latency, two options are proposed. First, a coordinated multipoint-based approach can be used, but this requires multiple access points be distributed around the system and likely be wired together in order to increase reliability. Second, a cooperative relaying system architecture was developed that manufactures its own diversity in a low-latency manner by using a scheduled, decentralized relaying protocol. The proposed system outperforms other communication techniques when enough users/relays are in the system based on a simplified link level analysis, and it is robust to interference and the fading distribution.

At the implementation level, this work explored aspects of a possible PHY implementation of the cooperative relaying system architecture that reuses as many blocks as possible from existing wireless PHYs while also meeting the architecture's unique requirements. Many trade-offs in the system design were discussed, but the analysis and final choice is still an open question.

A case study for low-power, low-latency, high-speed hardware was given in the form of an LDPC decoder for the IEEE 802.11ad standard. This was chosen because LDPC decoders are iterative and can add significant latency compared to the rest of the baseband depending on their design, and the IEEE 802.11ad standard has specifications close to what is needed for wireless control (i.e.short blocklength, multiple code rates). An efficient, highly-parallel, deeply-pipelined, flooding decoder was designed that balances latency from serial processing and power from deep voltage-frequency scaling. To decrease latency and power, the architecture was enhanced to processes multiple low-weight layers and multiple codewords simultaneously. To support decoding multiple codes, a simplified switching network based on the common matrix structure and the architectural enhancements was designed that mainly routes fixed wires instead of using multiplexers. After the global wordlength was selected, the precision of the marginalization memory, which dominates the decoder's power, was reduced. This provides a shallower tradeoff of error floor and power compared to further reduction of the decoder's wordlength. These architectural techniques can be applied to many standards (any that use the row-splitting method to derive lower-rate codes), and the reduced marginalization memory precision can be applied to any decoder. The decoder was implemented in 28nm FDSOI, which allows the design to be forward-biased in order to optimize active and leakage power. The decoder achieves sub-microsecond level latency with a power consumption of 6.2mW at 1.5Gb/s and 38.1mW at 6Gb/s. The power is reduced if lower throughput and higher latency is acceptable.

6.2 Future Work

Moving forward, the system designed in this work and the area of low-latency, high-reliability wireless system design have many open questions. The MAC layer analysis performed in this work uses simple fading channel models and idealized link models that are

either perfect if the rate is above capacity or fail if the rate is below capacity. More realistic models that take into account the effects of real-world channels, finite blocklength codes, synchronization errors, interference from simultaneously transmitting relays that cannot be decoded, non-zero Tx/Rx switching delays, and interference from other systems are required to analyze the true performance and scaling characteristics of the system. Next, the system must be fully implemented and tested to verify its performance. This includes making design decisions outlined in Chapter 4, including the signal processing algorithms (i.e. synchronization, channel estimation, error correction, CDD diversity extraction or diversity extraction using a different space-time code), modulation and symbol size, and reference signaling. Also, new algorithms must be designed that use information from a previous cycle to aid in the current cycle since the channel conditions can be highly correlated because there are multiple cycle times in a coherence time. In particular, this can be done for the channel estimator. Afterwards, the performance must be verified using bit-level simulations and then in hardware, which will require the design of the custom blocks for those components of the system that do not exist in current standards.

Beyond this work, the area of low-latency, high-reliability wireless system design has many open questions. Primarily, there is no theoretical framework for assessing the tradeoff between latency, reliability, and spectral efficiency for networks that can be used to judge the optimality of proposed systems. This is essentially information theory in the finite blocklength regime. The next step would be to co-design the communication system and the control algorithm in order to relax the communication specifications as much as possible and enable even lower-latency, higher-reliability, larger systems. In other words, bringing in the error model of the communication system into the control algorithm design and allowing for more errors in the communication system could relax the network's required specifications. Third, the issue of coexistence with other systems has not been considered except by looking at the robustness of the system to a static error probability. Methods to deal with interfering systems are key to future IoT applications since there will be many such systems in close proximity. Finally, the proposed cooperative relaying system does not consider power consumption, so a methodology to evaluate power consumption and more power-efficient system architectures are required because sensors and actuators will have limited battery life or power delivery capability.

Bibliography

[1] G. Fettweis and S. Alamouti, "5G: Personal mobile Internet beyond what cellular did to telephony," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 140–145, Feb. 2014.

[2] F. Boccardi, R. Heath, A. Lozano, T. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[3] H. Viswanathan, "Low latency wireless: a cellular network perspective," NSF Workshop on Achieving Ultra-Low Latencies in Wireless Networks, Tempe, Arizona, 2015.

[4] J. Anderson, F. Rusek, and V. Owall, "Faster-than-nyquist signaling," *Proc. of the IEEE*, vol. 101, no. 8, pp. 1817–1830, Aug. 2013.

[5] G. Fettweis, M. Krondorf, and S. Bittner, "GFDM - generalized frequency division multiplexing," in *Proc. of IEEE Vehicular Technology Conference*, Apr. 2009, pp. 1–4.

[6] B. Farhang-Boroujeny, "OFDM versus filter bank multicarrier," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 92–112, May 2011.

[7] F. Schaich and T. Wild, "Waveform contenders for 5G - OFDM vs. FBMC vs. UFMC," in *Proc. of Int. Symp. on Commun., Control and Signal Processing*, May 2014, pp. 457–460.

[8] T. Wild, F. Schaich, and Y. Chen, "5G air interface design based on universal filtered (UF-)OFDM," in *Proc. of International Conference on Digital Signal Processing*, Aug. 2014, pp. 699–704.

[9] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July, Oct. 1948.

[10] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[11] R. G. Gallager, "Low density parity check codes," Ph.D. dissertation, M.I.T., 1963.

[12] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

[13] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[14] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks," in *Proc. IEEE Int. Conference on Comput. Commun.*, vol. 1, Mar. 2004.

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

[16] D. Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 630–643, 2003.

[17] J. Nilsson, "Real-time control systems with delays," Ph.D. dissertation, Lund Institute of Technology, 1998.

[18] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Trans. on Automat. Control*, vol. 55, no. 9, pp. 2030–2042, Sept. 2010.

[19] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *IEEE Trans. on Control Syst. Technology*, vol. 10, no. 3, pp. 438–446, May 2002.

[20] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Trans. on Automat. Control*, vol. 56, no. 10, pp. 2456–2461, Oct. 2011.

[21] *WirelessHART*, HART Communications Foundation , 2007.

[22] N. Ploplys, P. Kawka, and A. Alleyne, "Closed-loop control over wireless networks," *IEEE Control Syst.*, vol. 24, no. 3, pp. 58–71, 2004.

[23] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, and M. Nixon, "WirelessHART: Applying wireless technology in real-time industrial process control," in *Proc. IEEE Real-Time and Embedded Technology and Applicat. Symp.*, Apr. 2008, pp. 377–386.

[24] N. Nikaein and S. Krea, "Latency for real-time machine-to-machine communication in LTE-based system architecture," in *Proc. IEEE European Wireless Conference*, Apr. 2011, pp. 1–6.

[25] V. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. on Ind. Informatics*, vol. 7, no. 4, pp. 529–539, Nov. 2011.

[26] P. Kumar, "New technological vistas for systems and control: The example of wireless networks," *IEEE Control Syst.*, vol. 21, no. 1, pp. 24–37, Feb. 2001.

[27] D. Jiang, H. Wang, E. Malkamaki, and E. Tuomaala, "Principle and performance of semi-persistent scheduling for VoIP in LTE system," in *Proc. IEEE Int. Conference on Networking and Mobile Computing Wireless Commun.*, Sept. 2007, pp. 2861–2864.

[28] K. Zhou, N. Nikaein, R. Knopp, and C. Bonnet, "Contention based access for machine-type communications over LTE," in *Proc. IEEE Veh. Technology Conference*, May 2012, pp. 1–5.

[29] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. Johnson, "M2M: From mobile to embedded Internet," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 36–43, Apr. 2011.

[30] P. Suriyachai, U. Roedig, and A. Scott, "A survey of MAC protocols for mission-critical applications in wireless sensor networks," *IEEE. Commun. Surveys and Tutorials*, vol. 14, no. 2, pp. 240–264, Mar. 2011.

[31] Y. Polyanskiy, H. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[32] M. Weiner, M. Jorgovanović, A. Sahai, and B. Nikolić, "Design of a low-latency, high-reliability wireless communication system for control applications," in *Proc. of IEEE International Conference on Communications*, Jun. 2014, pp. 3829–3835.

[33] V. N. Swamy, S. Suri, P. Rigge, M. Weiner, G. Ranade, S. Anant, and B. Nikolić, "Cooperative communication for high-reliability low-latency wireless control," in *Proc. of IEEE International Conference on Communications*, 2015.

[34] M. Weiner, B. Nikolić, and Z. Zhang, "LDPC decoder architecture for high-data rate personal-area networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 2011, pp. 1784–1787.

[35] M. Weiner, M. Blagojević, S. Skotnikov, A. Burg, P. Flatresse, and B. Nikolić, "A scalable 1.5-to-6Gb/s 6.2-to-38.1mW LDPC decoder for 60GHz wireless networks in 28nm UTBB FDSOI," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2014, pp. 464–465.

[36] U. Madhow, *Fundamentals of Digital Communication.* Cambridge University Press, 2008.

[37] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communication*, 3rd ed. Springer, 2003.

[38] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, 2012.

[39] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[40] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, 1995.

[41] *IEEE Standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical Link Control*, IEEE Std. 802.2, 2013.

[42] *IEEE Standard for Ethernet*, IEEE Std. 802.3, 2014.

[43] Z. Wang, Y.-Q. Song, J.-m. Chen, and Y.-X. Sun, "Real time characteristics of Ethernet and its improvement," in *Proc. IEEE World Congress on Intelligent Control and Automation*, vol. 2, 2002, pp. 1311–1318.

[44] C. Borrelli, "IEEE 802.3 cyclic redundancy check," *Application note: Virtex Series and Virtex-II Family*, vol. 23, 2001.

[45] R. Zurawski, Ed., *The Industrial Communication Technology Handbook*. CRC Press, 2014.

[46] (2013, Aug.) Introduction to SERCOS III with industrial Ethernet. [Online]. Available: http://www.sercos.com/technology/sercos3.htm

[47] H. Holma and A. Toskala, *LTE for UMTS-OFDMA and SC-FDMA based radio access*. John Wiley & Sons, 2009.

[48] F. Khan, *LTE for 4G Mobile Broadband: Air Interface Technologies and Performance*. Cambridge University Press, 2009.

[49] A. Technologies, *N9080A LTE FDD and N9082A LTE TDD Measurement Applications Technical Overview with Self-Guided Demonstration*, 2010. [Online]. Available: http://www.keysight.com/upload/cmc_upload/All/5989-6537EN_1.pdf?&cc=US&lc=eng

[50] *ZigBee 2012*, ZigBee Alliance , 2012.

[51] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*. New Jersey: Prentice Hall, 2001.

[52] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. New York: Wiley, 2007.

[53] K. Åström and B. Wittenmark, *Computer Controlled Systems: Theory and Design*. Prentice Hall, 1997.

[54] F. Xia, X. Kong, and Z. Xu, "Cyber-physical control over wireless sensor and actuator networks with packet loss," in *Wireless Networking Based Control*. Springer, 2011, pp. 85–102.

[55] 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.

[56] 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.

[57] 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.

[58] G. Wu, Z. Li, H. Wang, and W. Zou, "Selective random cyclic delay diversity for HARQ in cooperative relay," in *Proc. IEEE Wireless Communications and Networking Conference*, Apr. 2010, pp. 1–6.

[59] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: A distributed wireless architecture for exploiting sender diversity," in *Proc. ACM SIGCOMM*, 2010, pp. 171–182.

[60] P. Wu and N. Jindal, "Coding versus ARQ in fading channels: How reliable should the PHY be?" *IEEE Trans. on Commun.*, vol. 59, no. 12, pp. 3363–3374, Dec. 2011.

[61] D. Baron, S. Sarvotham, and R. Baraniuk, "Coding vs. packet retransmission over noisy channels," in *Proc. IEEE Conference on Inform. Sci. and Syst.*, Mar. 2006, pp. 537–541.

[62] Y. Polyanskiy, H. Poor, and S. Verdu, "Dispersion of Gaussian channels," in *Proc. IEEE Int. Symp. on Inform. Theory*, June 2009, pp. 2204–2208.

[63] T. Watteyne, S. Lanzisera, A. Mehta, and K. S. Pister, "Mitigating multipath fading through channel hopping in wireless sensor networks," in *Proc. of IEEE International Conference on Communications*, May 2010, pp. 1–5.

[64] J. Laneman, "Cooperative diversity in wireless networks: Algorithms and architectures," Ph.D. dissertation, MIT, 2002.

[65] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. Part I. system description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.

[66] S. Mishra, A. Sahai, and R. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. IEEE International Conference on Communications*, vol. 4, June 2006, pp. 1658–1663.

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

[68] S. Girs, E. Uhlemann, and M. Bjorkman, "Increased reliability or reduced delay in wireless industrial networks using relaying and Luby codes," in *Proc. IEEE Conference on Emerging Technologies Factory Automation*, Sept. 2013, pp. 1–9.

[69] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Communications Magazine*, vol. 49, no. 2, pp. 102–111, 2011.

[70] *Coordinated multi-point operation for LTE - physical layer aspects - Release 11*, 3GPP TR 36.819, 2011.

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

[72] 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, July 2010.

[73] L. Ozarow, S. Shamai, and A. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 2, pp. 359–378, May 1994.

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

[75] 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.

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

[77] R. Narasimhan, A. Ekbal, and J. Cioffi, "Finite-SNR diversity-multiplexing tradeoff of space-time codes," in *Proc. of IEEE International Conference on Communications*, vol. 1, May 2005, pp. 458–462.

[78] A. S. Y. Poon, R. W. Brodersen, and D. N. C. Tse, "Degrees of freedom in multiple-antenna channels: a signal space approach," *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 523–536, Feb. 2005.

[79] "MAX2830 data sheet," Maxim Integrated, San Jose, CA.

[80] G. Aiello and G. Rogerson, "Ultra-wideband wireless systems," *IEEE Microwave Magazine*, vol. 4, no. 2, pp. 36–47, Jun. 2003.

[81] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *Proc. of IEEE Vehicular Technology Conference*, Jun. 2013, pp. 1–5.

[82] Y. Kou, S. Lin, and M. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711 –2736, Nov. 2001.

[83] S. Lin, L. Chen, J. Xu, and I. Djurdjevic, "Near Shannon limit quasi-cyclic low-density parity-check codes," in *Proc. of IEEE Global Telecommunications Conference*, vol. 4, Dec. 2003, pp. 2030 – 2035.

[84] *IEEE Standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Amendment 3: Enhancements for Very High Throughput in the 60GHz Band*, IEEE Std. 802.11ad, 2012.

[85] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.

[86] D. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Workshop on Signal Process. Syst.*, Oct. 2004, pp. 107–112.

[87] M. Mansour and N. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," *IEEE J. of Solid-State Circuits*, vol. 41, no. 3, pp. 684–698, Mar. 2006.

[88] A. Vila Casado, M. Griot, and R. Wesel, "Informed dynamic scheduling for belief-propagation decoding of LDPC codes," in *Proc. IEEE Int. Conf. on Commun.*, June 2007, pp. 932–937.

[89] A. J. Blanksby and C. J. Howland, "A 690mW 1Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE J. of Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, Mar. 2002.

[90] E. Yeo, P. Pakzad, B. Nikolić, and V. Anantharam, "High throughput low-density parity-check decoder architectures," in *Proc. IEEE Global Telecommunications Conference*, vol. 5, Nov. 2001, pp. 3019–3024.

[91] M. M. Mansour and N. R. Shanbhag, "Low-power VLSI decoder architectures for LDPC codes," in *Proc. Int. Symp. on Low Power Electron. and Design*, Aug. 2002, pp. 284–289.

[92] Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolić, "An efficient 10GBASE-T Ethernet LDPC decoder design with low error floors," *IEEE J. Solid State Circuits*, vol. 45, no. 4, pp. 843–855, Apr. 2010.

[93] S. Skotnikov, "Low power LDPC decoder design for 802.11ad standard," Master's thesis, École Polytechnique Fédérale de Lausanne, 2012.

[94] R. Gonzalez, B. Gordon, and M. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE J. of Solid State Circuits*, vol. 32, no. 8, pp. 1210–1216, Aug. 1997.

[95] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De, "Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS ICs," in *Proc. IEEE Int. Symp. on Low Power Electron. and Design*, Aug. 2001, pp. 207–212.

[96] J.-P. Noel, O. Thomas, M. Jaud, O. Weber, T. Poiroux, C. Fenouillet-Beranger, P. Rivallin, P. Scheiblin, F. Andrieu, M. Vinet *et al.*, "Multi-UTBB FDSOI device architectures for low-power CMOS circuit," *IEEE Trans. on Electron Devices*, vol. 58, no. 8, pp. 2473–2482, Aug. 2011.

[97] P. Flatresse, B. Giraud, J. Noel, B. Pelloux-Prayer, F. Giner, D. Arora, F. Arnaud, N. Planes, J. Le Coz, O. Thomas, S. Engels, G. Cesana, R. Wilson, and P. Urard, "Ultra-wide body-bias range LDPC decoder in 28nm UTBB FDSOI technology," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2013, pp. 424–425.

[98] Y. S. Park, D. Blaauw, D. Sylvester, and Z. Zhang, "Low-power high-throughput LDPC decoder using non-refresh embedded DRAM," *IEEE J. Solid State Circuits*, vol. 49, no. 3, pp. 783–794, Mar. 2014.

[99] S.-W. Yen, S.-Y. Hung, C.-L. Chen, H.-C. Chang, S.-J. Jou, and C.-Y. Lee, "A 5.79-Gb/s energy-efficient multirate LDPC codec chip for IEEE 802.15.3c applications," *IEEE J. Solid-State Circuits*, vol. 47, no. 9, pp. 2246–2257, Sept. 2012.

[100] X. Peng, Z. Chen, X. Zhao, D. Zhou, and S. Goto, "A 115mW 1Gbps QC-LDPC decoder ASIC for WiMAX in 65nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2011, pp. 317–320.