CROSS-LAYER DESIGN FOR COMMUNICATION SYSTEMS

VINEET SRIVASTAVA

NATIONAL UNIVERSITY OF SINGAPORE

2004

CROSS-LAYER DESIGN FOR COMMUNICATION SYSTEMS

VINEET SRIVASTAVA

B.Eng(Hons.), NUS

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgements

First and foremost, a big thanks to my supervisor, Dr. Mehul Motani. He always allowed me the room and time to think and explore on my own and never ever forced his thoughts or ideas on me. At the same time, his comments and pointed questions kept me on course and inspired me to strive for clear logical thinking and expression. Thank you, Mehul—your friendly guiding influence has been a wonderful initiation into the world of research for me.

This work was conducted on a part-time basis, along with my full-time employment at Institute for Infocomm Research (I^2R). I thank my colleagues at I^2R for their understanding and support throughout my candidature.

I also thank my parents, friends and relatives for providing me the support and encouragement as I undertook my dip into the world of research. The somewhat philosophical discussions with my brother Puneet helped me stay focused amidst the inevitable uncertainty that a research pursuit offers. Words of thanks are also due to fellow students—in particular Hoang Anh Tuan, Lawrence Ong and Yap Kok Kiong—for the numerous stimulating and useful discussions and suggestions. I also acknowledge the anonymous reviewers of my publications, for their comments have helped me greatly to refine my research focus. Similarly, the comments from the examiners of the first draft of my thesis helped me to significantly improve the clarity of this thesis.

Finally, and most importantly, this work would not have been possible if not for the unflinching love and encouragement from my lovely wife, Nidhi. Her patience and understanding have meant a whole world to me.

Contents

List of Figures

List of Tables

List of Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
ARQ	Automatic Repeat Request
AWGN	Additive white Gaussian Noise
BCH	Bose-Chaudhary Hocquenghem
BPSK	Binary Phase-Shift Keying
BSC	Binary Symmetric Channel
CDMA	Code-Division Multiple Access
CLASS	Cross-Layer Signalling Shortcuts
CTS	Clear to Send
DARPA	Defense Advanced Research Project Agency
FCFS	First-Come First-Serve
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
GF	Galois Field
GSM	Global System for Mobile Communications
HDR	High Data Rate
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISO	International Standards Organization
JOCP	Joint Optimal Congestion Control and Power Control

LAN	Local Area Network
MAC	Medium Access Control
MPR	Multi-Packet Reception
MQAM	M-ary Quadrature Amplitude Modulation
NAK/NACK	Negative Acknowledgment
OAR	Opportunistic Auto Rate
OSI	Open Systems Interconnection
pdf	Probability Density Function
PHY (Layer)	Physical (Layer)
PK formula	Pollazcek-Khinchin formula
QPSK	Quadrature Phase-Shift Keying
RBAR	Receiver Based Auto Rate
RSSI	Received Signal Strength Indicator
RTS	Request to Send
SIR	Signal-to-Interference Ratio
SINR	Signal-to-Interference-and-Noise Ratio
SNR	Signal-to-Noise Ratio
STI	Service-Time Improving
TCP	Transmission Control Protocol
VG (bound)	Varshamov-Gilbert (bound)

Summary

Recent years have witnessed a widespread proliferation of wireless communication networks around the world. Wireless networks, and wireless communications in general, present several engineering challenges that were not present in their predecessor wired networks. At the same time, wireless networks—in particular ad-hoc wireless networks—offer certain modalities of communications that were just not possible in the wired networks. Such peculiarities of wireless communication networks are ushering in new paradigms for communication protocol design that better address the challenges and opportunities created by the wireless medium. This thesis looks at one such emerging paradigm termed as cross-layer design. The main idea behind cross-layer design is to allow enhanced dependence and information sharing between the different layers of the protocol stack. This is in contrast with the layered architectures that have been the cornerstone of data network design and development. In this thesis, we attempt to understand the cross-layer design methodology in more detail. We take stock of the existing work in this area, distill some key insights and spell out some of the open challenges.

After discussing in detail about the different aspects of cross-layer design, we present an instance of cross-layer design involving the link layer and the physical layer. In particular, we study the design of physical layer for a point-to-point communication system with the link layer average service time as our metric of interest. We come up with necessary and sufficient conditions on the parameters of specific physical layer processes like Forward Error Correction

and digital modulation such that the link layer average service time is favorably affected. We also study the impact of physical layer processing on the link layer average delay (sum of the average service time and the average queueing delay), assuming a Poisson arrival process.

Finally, we focus on forward error correction and merge the necessary and sufficient condition for improving the link layer average service time mentioned above with the Varshamov-Gilbert (VG) bound and the Sphere-packing bound, which are well-known coding theoretic results. Doing so enables us to study the existence of Service-Time Improving (STI) codes. By STI codes, we mean forward error correcting codes that reduce the average service time with respect to uncoded transmission for a fixed symbol rate and constellation size. We also explore the asymptotic case of large packet length and determine sufficient conditions for the existence of STI codes in this regime using the asymptotic form of the VG bound and the channel capacity theorem.

In summary, this thesis starts with a qualitative exploration of the various facets of cross-layer design. To the best of our knowledge, the methodology of cross-layer design has not been looked at so closely elsewhere. We then move on to apply some of the ideas to a specific scenario of a point-to-point communication system. Quantitative guidelines for physical layer processing with a higher layer metric in mind are developed for the system under consideration. These guidelines can be of practical importance. The application of coding theory ideas yields results that are more theoretical in nature but represent the application of ideas from two different disciplines—queueing theory and coding theory—in solving a communications problem.

Chapter 1

Introduction

1.1 Growing proliferation of wireless networks

Recent years have witnessed a sharp increase in network deployments that rely on some form of wireless communications. The familiar and almost ubiquitous cellular mobile phone networks are an example. The sharp rise in the proliferation of mobile phones and mankind's increasing dependence on them is evident even to a casual observer. In his book "The Smart Mobs" [1], author Howard Rheingold talks about the social transformations being brought about by the increasing pervasiveness of mobile phones. The fact that Rheingold substantiates his case by citing examples of events that have *already* happened speaks for the increasing proliferation of mobile phones around the world.

There has also been an increase in the deployments of wireless data networks, for example those based on the IEEE 802.11b standard [2]. Such networks are appearing in offices and commercial establishments like airports and restaurants. At the same time, primarily voice oriented networks like those based on Global System for Mobile communications (GSM) are being beefed up to handle data traffic [3, page 23]. In fact, if the growth in the wireless device subscriber

base and the increasing popularity of the Internet are put together, it can be projected that wireless networks are likely to be an integral part of the Internet in the future [3, page 18].

1.1.1 Two types of wireless networks

Wireless networks can broadly be divided into two categories, namely, cellular networks and ad-hoc networks. Cellular networks are formed when a certain geographical area is divided into cells, whereby each cell is served by a central controller node [4, page 6]. The voice network such as GSM is an example of a cellular network. The master node in the GSM network is usually called the base station. Most data networks today (for example the wireless local area networks) also follow the cellular design principle, whereby the mobile nodes communicate with a central node that is termed as an Access Point [2].

Wireless ad-hoc networks differ markedly from their cellular counterparts in that there is no fixed infrastructure in the network [3, page 401]. That is, there are no nodes that serve as the base-stations. Hence, all the network tasks need to be completed by the nodes themselves in a distributed manner. Ad-hoc wireless networks have found military applications, where deployment of infrastructure is not possible.

1.2 What is unique about wireless networks?

The growth in the popularity and pervasiveness of wireless networks is making wireless networks the center stage for the research and development activities in the field of data networking. In this section, we look at some aspects of wireless links and wireless networks that set them apart from their wired counterparts. We also highlight how the fundamental nature of the wireless channel offers

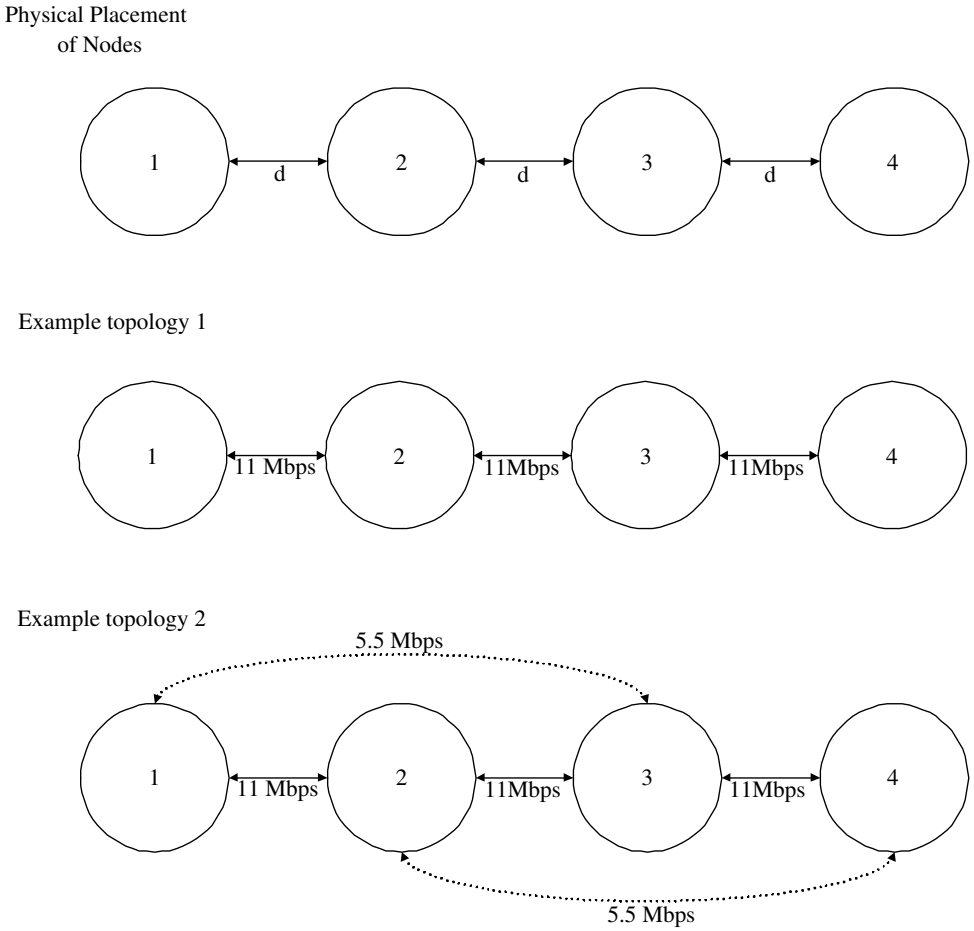

Figure 1.1: An example network configuration. Several network topologies are possible for the same physical placement of nodes.

some new modalities of communication that were not available in the wired networks.

1.2.1 The concept of a link

One of the most pertinent differences between wired communications and wireless communications concerns the concept of a communication link. A communication link between two nodes implies that the nodes in question can communicate directly with each other. In the wired world, there is a clear-cut concept of a communication link. There is a communication link between two nodes

if and only if there is a wire connecting the two nodes. On the other hand, there is no such notion of a communication link between two wireless nodes. Whether or not a link exists between two nodes depends on a host of factors, most notably, the signal-to-interference-and-noise ratio (SINR) at the receiver.

To see an implication, let us consider a hypothetical example network formed by communication nodes in figure 1.1. Without loss of generality, let us focus on node 1. The nodes with which node 1 can communicate directly (in a single hop) depends upon the transmission power and the physical layer signal processing. It is also possible that for a given transmit power, node 1 can communicate directly with all the other three nodes, albeit at different data rates. Hence, several network topologies are possible for the same physical placement of the nodes, depending upon the transmission power and the physical layer characteristics. In fact, the network topology depends not just on the transmission power of one node. It is the interplay between the transmission powers of all the nodes and the interference that they cause to the other nodes that determines the possible network topologies. Contrast this with a wired network, where the network topology is determined solely by the physical connection of the wires between the nodes.

1.2.2 The broadcast nature of the wireless channel

One of the peculiarities of wireless communications is that the communications are inherently broadcast in nature. Basically, a wireless transmitting node simply radiates power in form of electro-magnetic waves. With an omni-directional antenna, the radiated power would propagate in all directions, and can potentially be received by *all* nodes that are within a certain distance from the transmitting node.

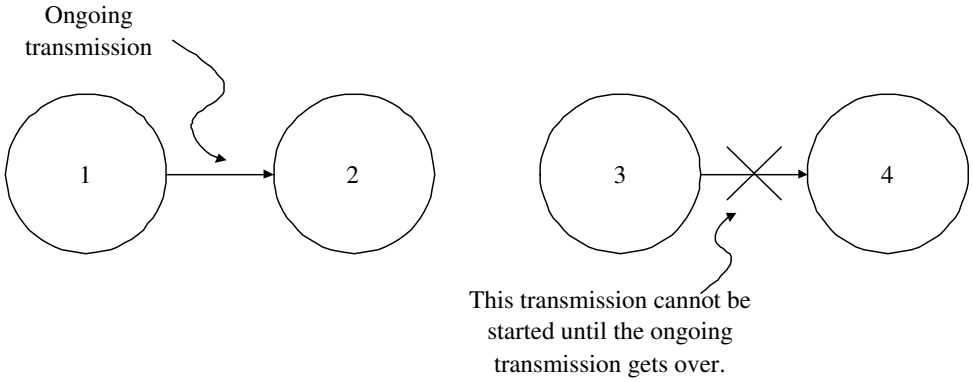

Figure 1.2: An implication of the broadcast nature of the medium. Transmission from node 3 to node 4 cannot take place if transmission from node 1 to node 2 is ongoing, assuming omni-directional antennas.

Let us now see an implication of the broadcast nature of the wireless channel. Consider the hypothetical network situation in figure 1.2. Let us say that at a given time instance, node 1 is sending data to node 2. This transmission, in effect, precludes the possibility of any communication from node 3 to node 4. This is because if node 3 were to start its transmission, there would be interference at node 2, resulting in the transmission from node 1 to node 2 to be lost. Contrast this with a scenario where the nodes are connected to each other with physical wires. In this case, simultaneous transmissions from node 1 to node 2 and node 3 to node 4 could continue.

The problem that we mentioned above is in fact the well known hidden-terminal problem [5]. A possible solution to the problem is the familiar Request-to-Send (RTS) and Clear-to-Send (CTS) handshake prior to the transmission of a packet [5]. In the RTS-CTS handshake, a node that has a data packet to send first sends out an RTS packet that is received by all the node that lie in the vicinity of the sending node. If the intended recipient receives the RTS packet and is ready to receive a data packet, it responds by sending out a CTS packet. It is only after this handshake between the transmitting node and the

intended receiver that the transmission of the data packets starts. Note that since both RTS and CTS packets are heard by all the nodes, packet collisions can be prevented. For instance, in the example network in figure 1.2, the CTS packet from node 2 (when node 2 responds to the RTS packet from node 1) will be heard by node 3. As a result, node 3 will not attempt any transmission to node 4 till the transmission from node 1 to node 2 is going on. It should be added that packet collisions are still possible, even with the RTS-CTS handshake. See [6] and the references therein for details.

An interesting observation about the RTS-CTS handshake above is that the RTS-CTS handshake itself relies on the broadcast nature of the wireless channel. Thus, in some sense, it makes use of the same capability that it tries to fight!

1.2.2.1 The problem of power control

The lack of a clear-cut definition of a communication link and the broadcast channel throw up the problem of power control in wireless ad-hoc networks. Basically, as discussed earlier, the transmission powers of the nodes determine the network topology. If the transmission powers of the nodes are large, all the nodes can possibly reach each other in a single hop. However, the broadcast nature of wireless communications means that a large transmission power also leads to higher interference on the other nodes. Hence, despite employing large powers, nodes may not be able to communicate with one another. On the other hand, if the transmission powers employed by the nodes are too small, the network might get fragmented. Hence, there is a need to perform power control, which means adjusting the powers transmitted by the different nodes in the network. This problem comes up as a result of the fundamental nature of the wireless medium and has no clear counterpart in wired networks.

1.2.2.2 Theoretical limit

We now discuss some recent theoretical results regarding the data transporting capacity of wireless ad-hoc networks. These results highlight both the unique nature of the wireless medium as well as the innovative communication schemes that can be employed on this medium.

Reference [7] considers the problem of computing the capacity of a wireless ad-hoc network with fixed (stationary) nodes. Communication nodes are assumed to be scattered randomly on a unit disk and sources and destinations are picked randomly. The nodes also have the capacity to act as relays. All the nodes transmit at a certain fixed power. The main result in [7] is that as the number of nodes is increased, the throughput per source-destination pair goes to zero. This is despite optimal scheduling of transmissions that is assumed in [7]. The primary factor resulting in a diminished throughput is the interference that the nodes cause to one another, thanks to the broadcast nature of the medium. An implication of the result in [7] is that very large scale wireless ad-hoc networks may be infeasible.

Reference [8] introduces mobility to the model considered in [7]. That is, it assumes that the nodes are capable of moving around in random motion. It is then shown that if the delay constraints are loose, mobility in the network can be *used* to obtain a constant throughput per source-destination pair, even as the number of nodes becomes large. The main idea is that as the nodes move around, the immediate neighborhood of the nodes changes with time. Thus, a node can split its packet into several parts and transmit the parts to different neighbors when they come close to itself. At a later time, the message can be delivered to the ultimate destination by the intermediate nodes, when they come close to the destination themselves. Thus, the actual transmission occurs

only between nodes that are close to each other.

The aforementioned results, in particular the latter result, underscore the fact that "out-of-the-box" solutions for communications might need to be devised to make a viable usage of the wireless medium.

1.2.2.3 Possibility of node cooperation

The broadcast nature of the wireless channel also gives rise to an intriguing possibility of nodes cooperating with each other. Basically, when a node transmits a packet, it can potentially be received by all its neighbors. These neighbors can then cooperate in delivering the packet to the final destination. As an example, consider [9] and the references therein, which deal with the possibility of user cooperation in cellular networks. Clearly, such possibilities are also available in ad-hoc wireless networks. In fact, even more interesting modes of node cooperation can be envisaged in wireless ad-hoc networks. For instance, a group of nodes can collectively cancel interference for some other node [10]. Reference [11] presents an information-theoretic analysis covering several such modalities allowed by the wireless medium.

Such possibilities can potentially be used for communication over wireless networks. We shall look at this more closely in Section 2.8. At this point, it suffices to note that cooperation between users, as described above, could not be realized easily in the wired networks. It is the broadcast nature of the wireless medium that makes such possibilities feasible.

1.2.3 Fluctuations in channel quality

One unique feature of wireless communication links comes from the fact that the quality of the channel can fluctuate with time. This happens when there

is relative motion between the source and the destination and/or the scatterers in the medium [12, page 801]. As a result, the received SNR varies with time. This is in sharp contrast to a wired link.

The fluctuation of received SNR can be seen to occur at two different time scales [4, page 21]. The long-term effects [4, page 26] result in significant fluctuations in the average received SNR and occur due to a significant change in the distance between the transmitter and the receiver during the course of the communication session. Basically, for a fixed radiated power, the average received power decays with distance. Thus, if the distance between the transmitter and the receiver changes appreciably, so does the average received power. Mobile nodes can also temporarily move into areas that are inaccessible for the radio waves, even though they lie within the coverage range of the network. An example is the familiar disruption of mobile phone conversation when entering elevators. This effect is usually referred to as shadowing.

On top of the long-term effects, there can also be fluctuation in the instantaneous received signal power. This effect is usually referred to as (short-term) fading [4, page 31]. As a result of fading, the average SNR is not changed. However, the instantaneous SNR undergoes rapid changes. The rapidity of the fluctuations in the instantaneous SNR depends upon the relative speed between the source and the destination—the higher the speed, the more rapid the fluctuations.

The effect of shadowing can be handled relatively easily by employing some form of an automatic gain control at the receiver or with power control at the transmitter, provided there is a feedback channel from the receiver back to the transmitter. Hence, short-term fading is more interesting and we discuss it below in more detail.

1.2.3.1 Should fading only be fought?

One approach of handling fading is to "fight" it. For instance, one could design signal-processing algorithms to mitigate the effects of fading. Generally, this involves making use of some form of diversity techniques (see for example [12, page 821]).

When the time-variation of the channel caused by fading is seen from a networking perspective however, the attitude towards fading tends to change somewhat. On one hand, fading makes the communication difficult by increasing the required SNR for a certain performance (for example E_b/N_0 of about 10 dB to achieve a bit-error-rate of 10^{-5} with BPSK modulation on an AWGN channel vs. 44 dB on a flat Rayleigh fading channel [12, page 816]). On the other hand, fading creates opportunities that can be exploited. Basically, as [13] puts it, fading allows the physical channel to be viewed as a "packet pipe" whose delay, rate and/or error probability characteristics vary with time. Contrast this with a wired communication channel whose characteristics remain largely time-invariant. Reference [13] considers a buffered single user point-to-point communication system and proposes a rate and power adaptation policy based on the fluctuations of the channel and the buffer occupancy. Reference [14] considers a similar situation and also comes up with an optimal adaptive policy that minimizes a linear combination of the transmission power and the buffer overflow probability. Such adaptations are not meaningful on wired links because the characteristics of the link do not vary with time.

Works such as [13] and [14] can be seen as solutions motivated from the information theoretic idea of water-filling [15]. The key idea in water-filling is to adapt the transmission power to the fading process, such that the signal is transmitted at a higher power when the channel is in a good state and at a low

power when the channel is in a bad state. A multi-user version of water-filling, resulting in the notion of multiuser diversity, can be found in [16].

In fact, the possibilities that fading creates in a multi-user setting are intriguing. Consider the problem of downlink scheduling on a cellular network [17]. The situation is as follows: There are a number of users whose data is arriving in a stochastic fashion at the base station. The base station needs to schedule the transmissions of the different users. Since the network traffic is bursty, a fixed assignment of slots/frequency bands to the users is not optimal. Instead, the allocation of the channel to the different users should be done dynamically. In doing the scheduling, the base station can take into account the state of the channel, and allow transmission to a particular user only if the channel between the base station and that particular user is in a "good" state. This makes intuitive sense, since, when the channel for a particular user is bad, there is no point in scheduling the transmission for that user. For theoretical bases of such channel dependent scheduling algorithms, we refer the readers to the results in [18], [19] and the references therein. Such ideas have already found practical application in the Qualcomm's High Data Rate (HDR) version of CDMA 2000 [17].

An interesting work in the area of channel dependent scheduling algorithms is [20]. In [20], antenna arrays are used in a "dumb" manner in order to artificially *create* fading in order to get the most out of a channel dependent scheduling algorithm. Hence, far from fighting fading, [20] views it as a welcome factor!

In short, fading, being unique to wireless links, creates new challenges as well as new opportunities that can be utilized for communication over wireless networks.

1.2.4 Other implications of mobility

Fading, as we saw above, is caused due to relative motion between the sender and the receiver and/or the surrounding environment. Mobility of users also has network level consequences. For example, in a wireless ad-hoc network, movement of nodes causes changes to the topology, requiring frequent enough routing updates. In cellular networks, users can move in and out of coverage ranges of the base-stations. Ideally, a user would like his communication session (e.g. a voice call) to continue regardless of such motion, and this throws up the problem of mobility management. A discussion of mobility in cellular networks can be found at [3, page 247].

In a heterogeneous environment with several networks, a mobile node might move from the coverage range of one kind of network into the coverage range of another kind of network [3, page 100]. For example, a person accessing the Internet on the road using the cellular mobile phone network might move into a building served by a wireless LAN. Ideally, from a user's perspective, such a change should be seamless, and ongoing connections and data transfers should not be interrupted.

It should be mentioned that the problem of guaranteeing seamlessness as described above is not unique to wireless networks. As [3, page 14] points out, a person who is accessing the Internet through a cable might also request for the same kind of seamlessness when changing the mode of access.

1.2.5 Device energy limitations

In most cases, mobile nodes are powered by batteries that usually have limited source of energy. Thus, communication methodologies need to be energy efficient besides being efficient from the data transfer perspective. This is much

in contrast with wired networks, where the nodes are usually connected to the mains supply. Hence, energy efficiency may not be a primary concern. On the other hand, energy management is a major issue in wireless communications.

1.3 Layered architectures

We have been discussing how wireless networks and wireless links present opportunities and create problems that are different from those in wired networks. Before moving further, we now look at what are called the "layered architectures". Layered architectures have thus far been the cornerstone of the design and development of communication networks like the Internet. We start the discussion of layered architectures by clarifying the term "architecture".

1.3.1 Defining architecture

The word "architecture" is widely used in diverse disciplines of engineering. Though it is tricky to define architecture precisely and unambiguously ([21], [22]), for our purpose, we can understand an architecture to be a high-level specification of the system specifying the breakdown of the overall task into smaller modules and the interfaces between the different modules.

Basically, any complicated engineering task admits itself to a decomposition of the problem at hand into smaller, more manageable tasks. In a system that implements the overall task, the smaller sub-tasks can be individually performed in separate modules. Such a high level functional decomposition leads to what can be called an architecture. An architecture specifies the tasks to be performed by the different modules, the interfaces to be provided by the modules, and the protocols that enable the different modules to work together toward the completion of the overall task. Practically speaking, an architecture translates

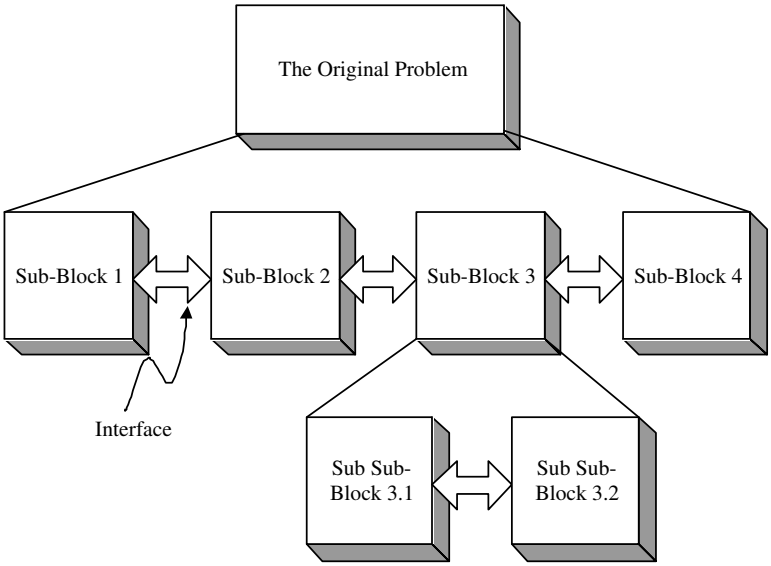

Figure 1.3: An illustration of a hierarchical architecture.

the system specification into a block-diagram representation with well-defined boundaries and information/data flow. Reference [21] takes a more abstract view of what "architecture" means and defines it as a set of rules and conventions over which a system is built.

A key component of a good architecture is the definition of the boundaries between the different blocks. It is important to define the boundaries between the blocks in such a way that the internal functioning of a block is of no concern to the other blocks. Such a definition allows the different modules to appear as "black-boxes" performing a certain function and providing a certain interface. In designing the individual black boxes, one may again need to identify sub-blocks within the particular block. We depict the situation schematically in figure 1.3.

It is important to distinguish between an architecture and an implementation conforming to that architecture. An architecture defines a logical decomposition of a problem which allows designers to focus on the individual sub-tasks. An implementation, on the other hand, deals with realizing a solution built under a

particular architecture using the available resources. To highlight this distinction, [21] compares, for example, the Victorian architecture and a building built to the Victorian architecture.

1.3.2 The layered communication architecture

As we discussed in the Section 1.3.1, a good architecture decomposes the overall problem into logically independent blocks and clearly defines the interfaces between these blocks. In the case of communication systems, the problem of data transfer is decomposed into what are commonly termed as layers. This leads to the familiar layered architectures, like the Open Systems Interconnection (OSI) seven-layer model [21], [23, page 20] and the four-layer Transmission Control Protocol/Internet Protocol (TCP/IP) model [24]. We will discuss these in more detail in Section 1.3.4.

The general approach followed in the layered architectures is that of divide-and-conquer, whereby the network tasks are divided into smaller sub-tasks and put in the form of a hierarchy of layers. Each layer (barring the highest layer) provides a certain service to the layer above itself. In doing so, it invokes a more basic service provided by the layer just below itself [21].

The service at every layer is realized by the implementation of certain logic and/or data processing, which are termed as the *protocol* at the layer. The division of the tasks between the layers is done carefully such that a layer does not need to know the details of *how* the layer below itself is providing the expected service [25, page 25]. In other words, a layer does not need to know what kind of protocol is being implemented at a lower layer, as long as the lower layer is providing the correct service.

In the context of communication systems, the layers actually refer to dis-

tributed systems [23, page 18] with at least two peer entities located at either side of the communication link. Thus, at every layer, the assigned service is realized by distributed processing involving the different entities from the layer. A common way to realize the protocols at the different layers is by adding headers to the data [25, page 28]. Thus, at the transmitting side, every layer receives a composite data packet that contains the original application data and the headers added by the layers above itself. On its part, the particular layer adds its own header and passes the packet on to the next lower layer. At the receiving side, each layer progressively strips off the header inserted at its peer entity at the transmitting end and passes on the remaining data packet to the layer above itself. Communication between the layers is limited to procedure calls invoking the services of a lower layer. This happens through well defined interfaces, referred to as the Service Access Points (SAP) [21]. Functionally, the layers are separated from each other, and can be designed in independence from one another.

1.3.3 Benefits of layering

There are several reasons behind the universal acceptance of layered architectures in the data networking community. Firstly, through layering, the overall problem of networking is broken into smaller and more manageable parts. Not only does this identify the individual problems to be solved, it also facilitates the development of standard protocols at the different layers. In fact, providing a framework for the development of standards was one of the goals behind the familiar OSI seven-layer model [21]. Secondly, layering of protocols has provided a structure through which the several innovations at the different parts of the networks can be handled. To clarify this point, recall that in a layered architec-

ture, the details of how a service is provided by a layer are not of importance. This means that innovations at every layer can continue, unhindered from the developments at other layers. Thirdly, layering has been a vital tool in enhancing the engineers' understanding of data networks—a study of the layering and layered communication architectures inevitably features in the introductory courses on computer communications and networking.

1.3.4 Important layered architectures

There have been several layered architectures for the different data networks in the past. The two most prominent ones are the OSI seven-layer model [21], [23, page 20] and the four-layer TCP/IP model [24]. The four-layer TCP/IP model is the model that has shaped the Internet. It came into being as part of an initiative undertaken by the United States (US) Department of Defense to connect different kinds of packet-switched data networks. The idea was to enable a communication device connected to a packet switched network to communicate with any other communication device connected to any other packet switched network. Thus, the TCP/IP model undertook the task of realizing a network of networks, where the individual networks were connected to each other by gateways for routing the packets [24]. A discussion of the key considerations behind the TCP/IP model can be found in [26].

The OSI seven layer model came into being as a result of an initiative taken up by the International Standards Organization (ISO) to come up with a set of standards that would allow disparate systems anywhere in the world to communicate with each other. The idea was to allow interoperability between communication equipments developed by the different vendors. The basic seven-layer reference model was published in early 1980's and it became an international

standard (ISO 7498) in 1983 [21]. The OSI model had split the lowest layer of the TCP/IP model into three separate layers. It had also done the same for the highest layer.

It is interesting to briefly look at the chronology of the events surrounding the development of the two models [27], [28, page 39]. In the case of the TCP/IP model, the protocols at the different layers were developed before formal layer definitions had been made. As mentioned above, the TCP/IP model has its origins in the efforts by the US Department of Defense to connect different packet switched networks. As highlighted in [26], the design goals in such an endeavor dictated a connection-less mode of communications with end-to-end flow control to be provided by the transport layer. These considerations led to the development of the TCP/IP protocol suite in the early seventies. With TCP/IP protocols getting bundled with the UNIX operating system, the popularity of these protocols in the research community shot up [27]. In due course of time, layer definitions were attached to the model [28, page 39] and thus came the four-layer TCP/IP model.

By contrast, in the case of the OSI seven-layer model, the layers were defined before the development of protocols [28]. As mentioned earlier, the idea behind the OSI model was to standardize the protocol development effort, and to do so, a flexible architecture was provided in the seven-layer model. Subsequent to the publication of the seven-layer model, protocols at the different layers were developed and published as separate international standards. As the protocols were being developed, the designers started pointing out some shortcomings and possible enhancements to the OSI basic reference model. A summary of the major changes to the original reference model and their motivation—with palpable political overtones—can be found in [29].

Which of the two models is better suited to the needs of the networking

community appears to be a hotly debated topic. Basically, the TCP/IP model was designed by the Department of Defense while the OSI model was conceived by the vendors of communications equipment. The differences in priorities led to understandable differences in the model (for example connection-oriented services in OSI vs. connection-less service in TCP/IP model) [27]. While comparing the two models in detail is beyond the scope of the present work, we refer the reader to [28, page 38] for an insightful comparison and critique of both the models. Reference [28, page 44] goes on to propose a five-layer "best of both worlds" model which is a hybrid of the OSI model and the TCP/IP model. To an extent, the five-layer model proposed in [28, page 44] reflects the complimentary position that the two models have come to attain. The OSI seven-layer model provides clarity to network organization and serves as a great tool for understanding data networks. In terms of protocols, TCP and IP remain the dominant protocols of the data networks, given the popularity of the Internet.

The discussion in this thesis does not explicitly relate to any particular layered architecture. However, for the sake of consistency, figure 1.4 shows the layered architecture that we use as the basis of discussion in this thesis. In table 1.1, we briefly describe the network functionality that we associate with each of the layers in figure 1.4. It should be added that the data-link control (DLC) and the medium-access control (MAC) layers are often seen as sub-layers of the link layer [23, page 24]. We shall do the same, though, as we mentioned above, this classification is not critical for the discussion in this thesis.

1.4 Layered architectures and wireless links

We saw in Section 1.1 that wireless networks are fast becoming the center stage for the communication networking research and in Section 1.2 that thanks to the

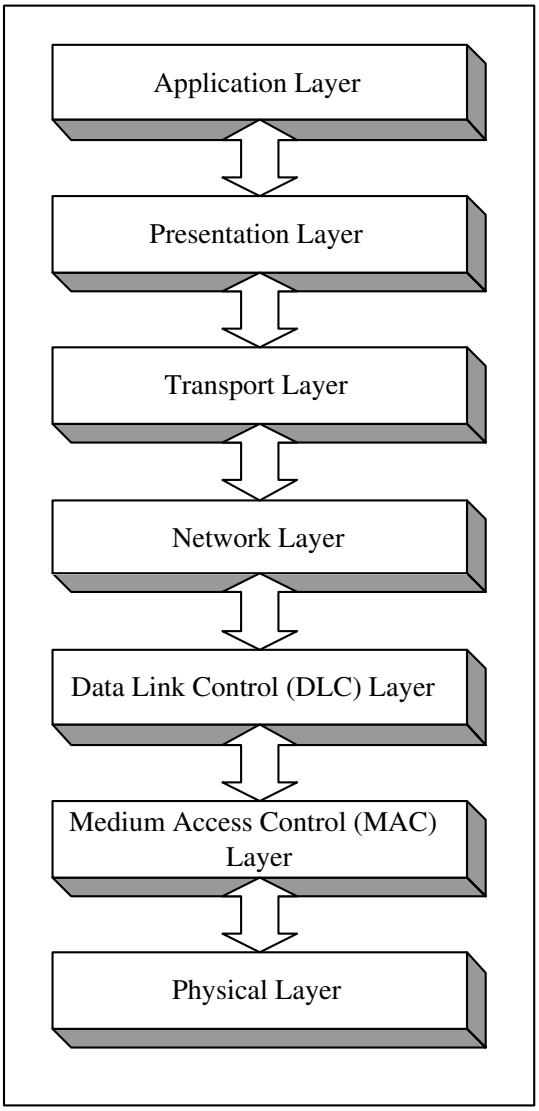

Figure 1.4: The reference model for the layered architecture.

Table 1.1: Network functionality performed by the different layers in figure 1.4

Layer	**Function**
Application	Generation of data (The source)
Presentation	Data encryption, Data compression
Transport	End-to-End flow control
Network	Routing
Data-Link Control (DLC)	Link-level error control
Medium Access Control (MAC)	Contention for medium access
Physical	Actual transmission/reception over the channel

unique nature of the wireless medium, several new possibilities and challenges that did not exist with wired networks open up with wireless networks. Then, in Section 1.3, we saw that protocols for the wired networks have been designed and handled through the layered architectures and that there are several benefits of layering. Given that wireless networks are gaining popularity, a question that naturally comes up is whether or not the layered architectures are suitable for wireless networks too. If so, how should the presence of wireless links in a network be dealt with in the framework of the layered architectures?

1.4.1 Wireless link as just another physical layer?

One obvious way of incorporating wireless links into the network design is to treat the wireless link as just a different kind of physical layer. With this viewpoint, the problem of handling the wireless link in a network decomposes into two separate problems. The first part of the problem is to actually design the wireless physical layer such that it can provide the desired quality (in terms of bit error rate etc.) to the higher layers. The second part of the problem is to boost the protocols at the higher layers such that they can work with the wireless physical layer which, as we discussed in Section 1.2, exhibits some peculiarities not found in the wired world.

This approach is appealing from an architectural viewpoint. Firstly, this approach is a natural extension of the design efforts conducted for earlier networks. Secondly, to a large extent, it separates the problem of design of the wireless link from the design of the application and the other network protocols—which is an architecturally desirable point.

By and large, the cellular networks of today deal with wireless links in the way described above [3, page 25]. The idea is to divide the network into two

portions, namely, the access network and the core network. The access network is where the wireless link resides. In the case of cellular networks, the core network is usually wired. Hence, the task is two-fold. On one hand, the access technologies need to be made reliable to provide better quality of service to the higher layers (the core network in this case). At the same time, the protocols in the core network need to be beefed up so that they can support several different wireless access technologies [3, page 27].

1.4.2 The idea of cross-layer design

The presence of wireless links in a network can be handled by treating the wireless link as just another physical layer, as we saw above. In that case, one maintains the general structure of the layered architectures, which implies separation of the layers at design-time and limited and controlled interactions between the layers at run-time.

One question that naturally comes up is whether this treatment of wireless links is in line with the unique characteristics of wireless links that we discussed earlier. For example, if one conforms to a strict separation of the wireless physical layer and the higher layers of the network, can the opportunities created by fading be adequately utilized at the higher layers? Then there are problems like power control and energy management which concern several layers of the stack at once. Power control, for instance, controls the network topology—a concern of the network layer. It also impacts how much spatial re-use can be achieved, that is, how far apart can two ongoing communication sessions be without interfering with each other—a concern of the MAC layer. Power control is also linked to the processing at the physical layer, because the signal processing at the physical layer determines how stringent the requirements on

the power control need to be. Finally, the transmitted power(s) determines the lifetime of the nodes (and the network) which one would want to maximize. Hence, the power of power control cannot possibly be handled at any one layer in isolation, as is done while designing protocols in the framework of the layered architectures.

Thus, on the face of it, the layered architecture seems very stiff and unable to adequately address the complicated modalities of communication possible in wireless networks. Ideally, one would like an architecture that, while possessing the desired features of the layered architectures, will also be more flexible and allow more synergy between the layers to exploit the inherent properties of wireless communications.

In line with this goal of a more flexible architecture, a new design paradigm is being discussed in the literature lately. It is generally given the name of cross-layer design. The key idea in cross-layer design is to allow enhanced information sharing and dependence between the different layers of the protocol stack [5], [17], [30]. It is argued that by doing so, performance gains can be obtained in wireless networks since the resulting protocols are more suited to be employed on wireless networks as compared to protocols designed in the strictly layered approach. Broad examples of cross-layer design include, say, design of two or more layers jointly, or passing of parameters between layers during run-time etc. We shall look at several specific examples and and a possible classification of the different cross-layer design proposals in Section 2.4.

1.5 Contributions of this thesis

This thesis deals with cross-layer design. There are two distinct pursuits that we take up: a thorough investigation of the protocol design methodology generally

termed as cross-layer design and a specific instance of cross-layer design involving the link layer and the physical layer. In the first pursuit—which is that of taking a close look at cross-layer design—we present a definition for cross-layer design, discuss the approaches for cross-layer design, and discuss some aspects of the impact of cross-layer design on architecture and performance. We also present a survey of the literature in this area and discuss the open issues and new opportunities for cross-layer design. In the second pursuit, we present an instance of cross-layer design whereby we consider the physical layer design in a point-to-point communication system with the link layer average service time as our metric of interest. We start with a primarily queueing-like analysis of the system and move on to apply some results from coding theory to gain further insights. All in all, this thesis makes the following contributions:

- We suggest a definition for cross-layer design (Section 2.2). Although arguably simple and obvious, the definition serves to unify the different interpretations that the term cross-layer design has assumed.

- We present a taxonomy of the existing cross-layer design proposals based on the kind of architecture violations the proposals represent (Section 2.4).

- We distill some insights from the current literature in the area of cross-layer design to come up with a unified platform, where we present a preliminary assessment of which layers need to be coupled and in what ways (Section 2.6).

- We spell out some open challenges and questions that designers proposing cross-layer design ideas need to start answering (Section 2.7).

- We discuss user-cooperation in wireless ad-hoc networks from the perspective of layering (Section 2.8). We find that incorporating user cooperation

in network protocol design would require significant violation of the layered architectures—cross-layer design, in other words—much like how the presence of wireless links in communication networks has.

- We look at a specific instance of cross-layer design, whereby we study the design of physical layer conditioned on a Stop-and-Wait Automatic Repeat Request (ARQ) system at the link layer. Our metric of interest is the link layer average service time. This cross-layer design represents design-coupling between two layers without the creation of new interfaces, which is one of the categories in the taxonomy of the cross-layer design proposals presented in Section 2.4.

- The cross-layer design that we consider leads us to necessary and sufficient conditions on parameters of physical layer processing like forward error correction and constellation size such that the link layer average service time is favorably affected (Section 3.5.1). We define Service-Time Improving (STI) codes as those forward error correcting codes that improve the link layer average service time while keeping the symbol rate and the constellation-size fixed.

- Using the Pollazcek-Khinchin formula for the queueing delay in M/G/1 queues, we study the effect of physical layer processing on the average link layer delay for the ARQ system under consideration (Section 3.6).

- We merge the necessary and sufficient condition on code parameters for a code to be STI code with the Varshamov-Gilbert (VG) bound and the sphere-packing bound (Section 4.4). This allows us to study the existence of binary STI codes. To the best of our knowledge, merging results from coding theory with conditions developed from a queueing viewpoint is

unique. We also consider the existence of STI codes in the asymptotic regime of large packet lengths using the asymptotic form of the VG bound and the channel capacity theorem (Section 4.5).

1.6 Organization of this thesis

Besides this introductory chapter, this thesis is organized into four other chapters. In Chapter 2, we take a close look at cross-layer design and the issues surrounding this protocol design methodology. Hence, the material in Chapter 1 is a preamble to the discussion in Chapter 2. Next, in Chapter 3, we look at the design of physical layer with the link layer average service time as the metric of interest. This chapter serves as an example of a cross-layer design involving physical layer and link layer. The additional background and concepts that are needed for an appreciation of the results discussed in Chapter 3 are covered in Section 3.2. Next, in Chapter 4, we expand on the results related to forward error correction from Chapter 3 by merging them with the coding theoretic results related to linear block codes. The coding theoretic results that are used in Chapter 4 are covered in Section 4.2. Finally, our main conclusions and some recommendations for further work are presented in Chapter 5.

Chapter 2

Cross-Layer Design: A survey and the road ahead

2.1 Introduction

As we discussed in Chapter 1, cross-layer design is an upcoming design paradigm for protocol design in wireless networks. There has in fact been much talk about cross-layer design for wireless communication networks lately. It has been argued repeatedly that layer boundaries, as specified in the layered architectures, are not suitable for wireless communications and performance gains can be made by giving up strict layering to do cross-layer design [5], [17]. Interestingly, to the best of our knowledge, there is no clear-cut definition for cross-layer design. Hence, not surprisingly perhaps, the term cross-layer design has assumed several different interpretations, with the general idea being that of a protocol design methodology that exploits the dependence between the different layers of the protocol stack to obtain performance gains. In this chapter, we take a closer look at cross-layer design in an attempt to identify the road ahead, and to address an important question, namely, "Where do we go from here?"

Looking at the state of the literature in the area of cross-layer design, we make three observations. Firstly, there is no clear consensus on what cross-layer design implies. As a result, isolated individual efforts, often representing varied interpretations of what cross-layer design means, are being made to address the shortcomings of protocol layering. Secondly, the synergy between the performance viewpoint from a communication and/or networking standpoint and the implementation concerns is weak—most proposals focus and elaborate on the performance gains. Thirdly, the fact that exploration of new paradigms of communications made possible by the wireless medium—for example user-cooperation—also requires cross-layer design does not seem to be getting sufficient attention. It is our aim in this chapter to discuss these three observations in more detail and take preliminary steps to address them.

We start the discussion by suggesting a definition for cross-layer design. Given its self-explanatory name, cross-layer design arguably needs no definition. However, not having *any* definition can lead to the term being abused and create confusion. Our definition, though arguably obvious, serves to unify the several interpretation of cross-layer design. We then take a snapshot of the current activity in the area of cross-layer design by arranging several cross-layer design proposals in a taxonomy and similarly categorizing the initial proposals for implementing cross-layer interactions. We then distill some key insights from the existing cross-layer design proposals to come up with a preliminary assessment of which layers need to be coupled and how. Next, we spell out some open challenges in this area and raise some questions that researchers making cross-layer design proposals can start answering. Finally, we discuss a communication methodology involving node cooperation which, while demonstrating a new opportunity created by wireless networks, significantly challenges the layered architecture.

Cross-layer design touches not just communications and networking, but is also intimately connected to concepts related to communications architecture. Layered architectures have served to make the protocol design activity systematic and modular, as we discussed in Chapter 1. Potential performance gains can always motivate a designer to not follow the layered architectures and do cross-layer design. But cross-layer design cannot be seen as an end in itself. It is absolutely essential to not overlook the architectural costs that accompany the performance gains that cross-layer design ideas might bring. This has recently been pointed out in [10], which highlights the importance of architecture and also warns designers of the possibility of inadvertent performance losses in presence of interaction between conflicting cross-layer design proposals. Our work in this chapter complements [10]—the definition of cross-layer design and the taxonomy of cross-layer design proposals provide a clearer appreciation of the "cautionary" perspective promoted in [10]. In addition, we also highlight some other open challenges and a new opportunity for cross-layer design, as mentioned above. All in all, we present both a survey of the ongoing work in the area of cross-layer design, and a platform over which new research can be built. We do not propose a new cross-layer design or discuss specific cross-layer design proposals in detail. Rather, we encourage a more holistic treatment of cross-layer design itself.

In the rest of this chapter, we present our definition for cross-layer design in Section 2.2. Next, in Section 2.3, we look at the general motivation behind cross-layer design in wireless networks. Then, in Section 2.4, a taxonomy of the existing cross-layer design proposals is presented and a similar categorization of the proposals for the implementation of cross-layer interactions is presented in Section 2.5. In Section 2.6, the key insights regarding which layers need to be coupled and how are presented. Next, in Section 2.7, we list the open

challenges in this area and in Section 2.8, we present a new opportunity for cross-layer design. Finally, in Section 2.9, we present the conclusions from this chapter.

2.2 Understanding Cross-Layer Design

2.2.1 A definition for Cross-Layer Design

A layered architecture, like the seven layer Open Systems Interconnect (OSI) model [23, page 20], defines a hierarchy of services to be provided by the individual layers, as we discussed in Section 1.3. As we saw, the services at the layers are realized by designing protocols for the individual layers, which can be implemented on the target platform to obtain a complete system.

At the protocol design phase, the designer has two choices. Protocols can be designed by respecting the rules of the original architecture. In the case of the layered OSI reference model, this would mean designing protocols such that they only make use of the services at the lower layers and not be concerned about the details of how the service is being provided. It also implies that the protocols would not need any interfaces that are not present in the reference architecture. Alternatively, protocols can be designed by violating the reference architecture. Since the reference architectures in communication and networking has traditionally been layered, its violation is generally termed as cross-layer design.

Definition 2.1. *Protocol design by the violation of a layered communication architecture is cross-layer design with respect to the original architecture.*

Comment 2.1.1. *Violation of a layered architecture involves giving up the luxury of designing protocols at the different layers independently. Protocols so*

designed impose some conditions on the processing at the other layer(s).

Comment 2.1.2. *Cross-layer design is defined as a protocol design methodology. However, a protocol designed with this methodology is also termed as cross-layer design.*

For exposition, consider a hypothetical three-layer model with the layers denoted by L_1, L_2 and L_3—L_1 being the lowest layer and L_3 the highest. One could design an L_3 protocol that needs L_1 to pass a parameter to L_3 at run-time. This calls for a new interface, not available in the architecture. Alternatively, one could view L_2 and L_1 as a single layer, and design a joint protocol for this "super-layer". Or, one could design the protocol at L_3, keeping in mind the processing being done at L_1—again giving up the luxury of designing the protocols at the different layers independently. All these are examples of cross-layer design with respect to the original architecture.

Architecture violations, like those introduced by cross-layer design, clearly undermine the significance of the architecture, since the architecture no longer represents the actual system. If many architecture violations accumulate over time, the original architecture can completely lose its meaning, leading to a situation termed as architecture-erosion in the software engineering community [22]. Architecture violations can have detrimental impact on the system longevity, as has recently been argued for the case of cross-layer design in [10].

2.2.2 Cross-Layer Design: A historical context

It is interesting to note that violations of the layered architectures have been proposed much before the recent interest in cross-layer design too. For instance, [31] has advocated the concept of "soft-layers" that allow more flexible information sharing between adjacent layers. Similarly, [32] talks of integrated layer

processing. What differentiates the current interest in the cross-layer design is the motivation for the architecture violations. In the past (e.g. in [31] and [32]), the architecture violations were proposed mainly to ensure *efficient* implementation of protocols—something the layered architectures by themselves do not guarantee [33]. On the other hand, the current surge in cross-layer design activity is motivated by the inability of the architecture to accommodate wireless links satisfactorily. It is argued that layered architectures either do not address the problems or do not sufficiently exploit the new opportunities created by the wireless links [5], [17] and hence cross-layer design is needed. We will limit our attention to such cross-layer designs that are motivated by the presence of wireless links in a communication network.

2.3 General motivation for Cross-layer design

Before moving further to look at some specific examples, let us see some general motivation for cross-layer design in wireless networks. On the pessimistic side, wireless links create several new problems for protocol design. The classic case of a TCP sender erroneously mistaking errors on wireless links to be an indicator of network congestion[1] is an example. On the optimistic side, wireless networks offer several avenues for opportunistic communication. For instance, the variation of the channel quality with time allows opportunistic usage of the channel [36], as we discussed in Section 1.2.3.1 while discussing fading from a networking perspective. Addressing the unique problems created by the wireless medium, as well as developing protocols to opportunistically use the wireless medium, both generally require cross-layer design.

[1]TCP is the famous transport layer congestion control protocol which infers packet loss to be an indication of congestion. See [34] for detailed introduction to TCP and [35] for some results regarding TCP over wireless links.

Additionally, the wireless medium offers some new modalities of communication. For instance, the physical layer can be made capable multi-packet reception (MPR) [37]. MPR refers to the capability of a receiver to be able to decode more than one simultaneously received packets. Contrast MPR with the so-called "collision model" [23, page 276], whereby, if more than one packet arrive together at the receiver, none can be successfully decoded. It should be clear that the medium-access problem, which traditionally assumed a collision model (for example in [23, page 276]), changes significantly in presence of MPR. The nodes can also make use of the broadcast nature of the channel and cooperate with one another in involved ways, one example of which we will discuss in Section 2.8. Making use of such "novel" modes of communication in protocol design also involves violating the architecture as we will discuss in Section 2.8.

2.4 Cross-Layer Design: a taxonomy

In recent times, a large number of cross-layer designs have been proposed. A classification based on the layers that are coupled by the different proposals can be found in [30]. In this section, we classify the existing cross-layer design proposals according to the kind of architectural violations they represent. Two points should be mentioned here. Firstly, our coverage of the cross-layer design proposals is meant to be representative and not exhaustive. Secondly, the reference architecture we assume is the one we presented in figure 1.4 , with the functionalities of the layers as specified in table 1.1.

The following architectural violations can be identified:

1. Creation of new interfaces (figure 2.1 A, B, C).

2. Merging of adjacent layers (figure 2.1 D).

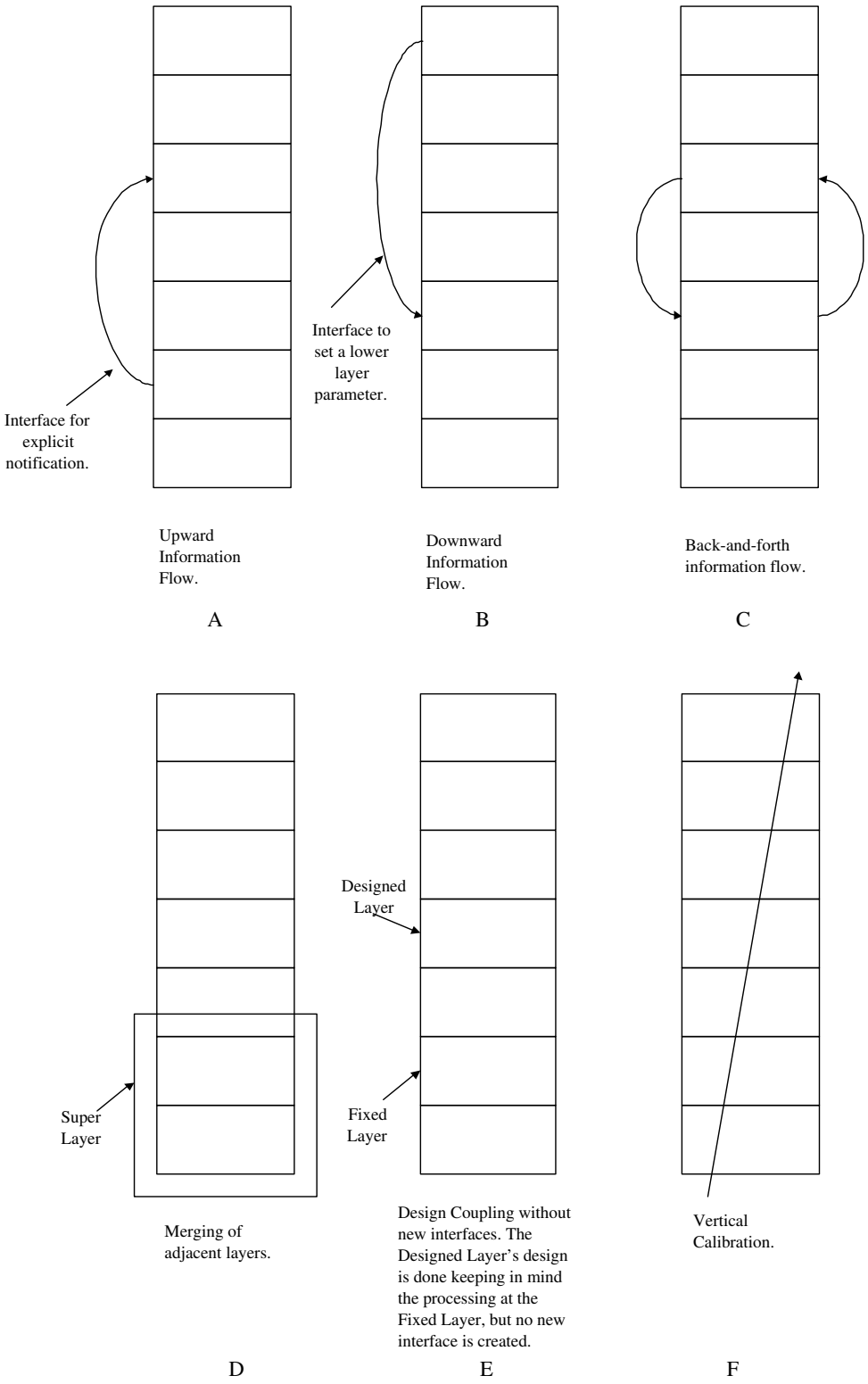

Figure 2.1: Illustrating the different kinds of cross-layer design proposals. The rectangular boxes represent the protocol layers.

3. Design coupling without new interfaces (figure 2.1 E).

4. Vertical calibration across layers (figure 2.1 F).

We now discuss these categories in more detail and point out the relevant examples.

2.4.1 Creation of new interfaces

Several cross-layer designs require creation of new interfaces between the layers. These can further be divided into three categories according to the direction of information flow along the new interfaces:

1. Upwards: From lower-layer(s) to a higher-layer (figure 2.1 A).

2. Downwards: From higher-layer(s) to a lower-layer (figure 2.1 B).

3. Back and forth: Iterative-flow between the higher and lower layer (figure 2.1 C).

2.4.1.1 Upward information flow

A higher layer protocol that requires some information from the lower layer(s) at run-time results in the creation of a new interface from the lower layer(s) to the higher layer, as shown in figure 2.1 A. For instance, if the end-to-end Transmission Control Protocol (TCP) path contains a wireless link, errors on the wireless link can trick the TCP sender into making erroneous inferences about the congestion in the network and, as a result, the performance deteriorates. Creating interfaces from the lower layers to the transport layer to enable explicit notifications alleviates such situations. For example, the explicit congestion notification (ECN) from the router to the transport layer at the Transmission

Control Protocol (TCP) sender can explicitly tell the TCP sender if there is congestion in the network to enable it to differentiate between errors on the wireless link and network congestion [17]. Similarly, [38] discusses the Explicit Link Failure Notification (ELFN) to address the problems created for TCP by the disruption of links in the end-to-end route due to mobility in wireless ad-hoc networks and proposes extensions to ELFN by adding more such explicit notifications.

Examples of upward information flow are also seen in the literature at the Medium-Access Control (MAC) layer (link layer in general) in form of channel-adaptive modulation or link-adaptation schemes ([36], [39], [40], [41]). The idea is to adapt the parameters of the transmission (e.g. power, modulation, code-rate etc.) in response to the channel condition, which is made known to the MAC layer (link layer) by an interface from the physical layer. Reference [14] extends such a cross-layer adaptation loop by deciding the link layer parameters by considering both the channel condition as well as the instantaneous buffer occupancy at the transmitter.

It is interesting to compare and contrast cross-layer design proposals that rely on upward flow of information to what can be called as self-adaptation loops at a layer. By a self-adaptation loop, we mean an adaptive higher layer protocol that respond to events that, within the constraints of layering, are directly observable at the layer itself. Hence, self-adaptation loops do not require new interfaces to be created from the lower layer(s) to the higher layer and cannot be classified as cross-layer designs. For example, consider the auto-rate fall-back mechanism for rate-selection [42] in wireless devices with multi-rate physical layers. The idea is that if some number of packets sent at a particular rate are successfully delivered, the data rate is increased, whereas, if a packet failure is experienced, data-rate is dropped. In this case, the rate selection mechanism

responds to the acknowledgments, which are directly observable at the MAC layer. Similar examples of self-adaptations are provided in [43] and [44]. In fact, TCP provides another example of a self-adaptation loop since TCP changes its window size in response to the acknowledgements observed at the transport layer itself.

2.4.1.2 Downward information flow

Some cross-layer design proposals rely on setting parameters on the lower layer of the stack at run-time using a direct interface from some higher layer, as illustrated in figure 2.1 B. Such downward flow of information is termed as Hints in [45]. As an example, the applications can inform the link layer about their delay requirement, and the link layer can then treat packets from the delay sensitive applications with priority [46].

2.4.1.3 Back and forth information flow

Two layers, performing different tasks, can collaborate with each other at run-time. Often, this manifests in an iterative loop between the two layers, with information flowing back-and-forth between the layers, as highlighted in figure 2.1 C.

Consider the collaboration between the MAC and the physical (PHY) layers in the uplink of a wireless LAN system, as in the Network-assisted diversity multiple access (NDMA) proposal ([47] and the references therein). Traditionally, collision resolution has been done exclusively by the MAC layer. With more sophisticated signal processing, the PHY layer becomes capable of recovering packets from collisions, and hence can collaborate with the MAC layer. This is the idea in the NDMA proposal [47]. Basically, upon detecting a collision, the base-station first estimates the number of users that have collided, and then

requests a suitable number of retransmissions from the set of colliding users. Thereupon, signal processing lets the base station separate the signals from all the colliding users.

As another example, consider the problem of joint scheduling and power control in wireless ad-hoc networks. Examples include [48], [49], [50] and [51]. Basically, power control determines the effective topology of the network by determining which nodes can communicate with one another in a single hop. If the transmitted power is too large, then many nodes may be connected by a single hop, but the interference also would be large. On the other hand, keeping the power too small can make the network fragmented or create too many hops and hence added MAC contention. Protocols resulting from considering the joint problem of power control and scheduling often result in an iterative solution: Trying to keep the power level at an optimal level by responding to the changes in averaged throughput (e.g. see [51]). Reference [48] considers the joint scheduling and power control problem in time-division multiple access (TDMA) based wireless ad-hoc networks. A scheduling algorithm chooses the users that will transmit, and then a power control algorithm determines if the transmissions of all the chosen users can simultaneously go on. If the answer is a no, the scheduling algorithm is repeated. This iteration between scheduling and power control is repeated till a valid transmission schedule has been found.

While there are differing views on which layer should power control belong to[2], the collaborative nature of the cross-layer design mentioned above and the back-and-forth information flow that they require should be clear.

[2]Power control is mentioned as a physical layer task in [49]. A different view is taken in [52] and [48] where power control is placed at the MAC layer and in [53] where power control is placed at the network layer.

2.4.2 Merging of adjacent layers

Two or more adjacent layers can be designed together such that the service provided by the new "super-layer" is the union of the services provided by the constituent layers. While possibly increasing the design complexity substantially, the super-layer can be interfaced with the rest of the stack using the interfaces that already exist in the original architecture.

Although we have not come across any cross-layer design proposal that explicitly creates a super-layer, it is interesting to note that the collaborative design between the PHY and the MAC layers that we discussed in Section 2.4.1.3 while discussing the NDMA idea tends to blur the boundary between these two adjacent layers.

2.4.3 Design coupling without new interfaces

Another category of cross-layer design involves coupling between two or more layers at design time without creating any extra interfaces for information-sharing at run-time. We illustrate this in figure 2.1 E. For instance, consider the design of MAC layer in ad-hoc networks with smart antennas at the physical layer in [54] and the references therein. Similarly, [55] considers the design of MAC layer for the uplink of a wireless LAN when the PHY layer is capable of providing multi-packet reception capability [37]. Likewise, [56] and [57] consider the design of the physical layer conditioned on a Stop-and-Wait Automatic Repeat Request (ARQ) protocol at the link layer[3].

Usually, the motivation for conditioning the design of a layer on another layer is a change in technology at one layer, which needs to be complemented by corresponding changes to the other layer(s). For instance, multi-packet re-

[3]We shall discuss the cross-layer design considered in [56] and [57] in more detail in Chapters 3 and 4 in this thesis.

ception at the physical layer changes the role of the MAC layer, and hence the MAC layer needs to be redesigned, as in [55].

2.4.4 Vertical calibration across layers

The final category of cross-layer design proposals, as the name suggests, refers to adjusting parameters that span across the layers, as illustrated in figure 2.1 F. The motivation behind such cross-layer design is easy to understand. The performance seen at the level of the application is a function of the parameters at all the layers below it. Hence, it is conceivable that a joint tuning can help to achieve better performance than what individual settings of parameters—as would happen had the protocols been designed independently—can achieve.

As an example, [58] looks at optimizing the throughput performance of the Transmission Control Protocol (TCP) by jointly tuning power management, Forward Error Correction (FEC) and Automatic Repeat Request (ARQ)[4] settings. Similarly, [59] presents an example of vertical calibration where the delay requirement dictates the persistence[5] of the link-layer ARQ, which in turn becomes an input for the deciding the rate-selection through a channel-adaptive modulation scheme.

Vertical calibration can be done in a static manner, which means setting parameters across the layers at design time with the optimization of some metric in mind. It can also be done dynamically at run-time, which emulates a flexible protocol stack that responds to the variations in the channel, traffic and overall network conditions.

Static vertical calibration does not create significant consideration for implementations since the parameters can be adjusted once at design-time and

[4]FEC and ARQ are discussed in more detail in Section 3.2.
[5]explained in Section 3.2.2.

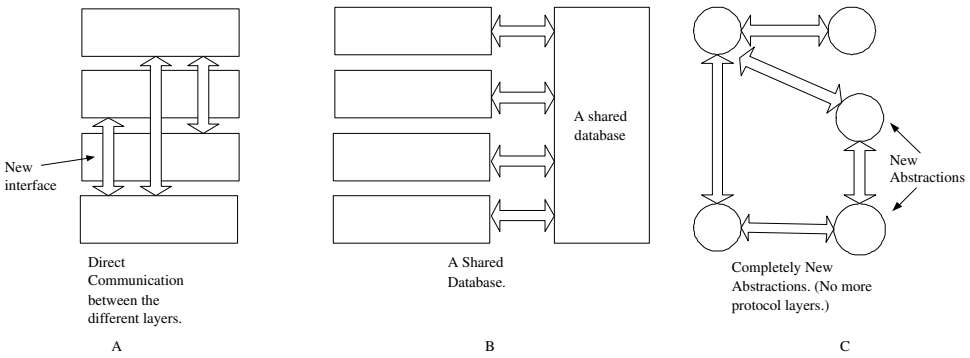

Figure 2.2: Proposals for architectural blueprints for wireless communications.

left untouched thereafter. Dynamic vertical calibration, on the other hand, requires mechanisms to retrieve and update the values of the parameters being optimized from the different layers. This may incur significant cost in terms of overheads and also impose strict requirements on the parameter retrieval and update process to make sure that the knowledge of state of the stack is current and accurate.

2.5 Proposals for new architectures

Cross-layer design proposals that we looked at in Section 2.4 demonstrate the violation of layered architectures at the protocol design phase itself. Hence, a question that naturally comes up is, "Can there be architectures that are general enough such that protocols for wireless networks can be designed without violating them?". In fact, this is an involved question. Determining what the new architectures should look like requires the study of not only the performance issues from a communication or networking viewpoint, but also an understanding of the implementation related issues. Nevertheless, some preliminary proposals have been made in the literature. These can be put into three categories:

1. Allowing the layers to communicate with each other (figure 2.2 A).

2. A shared database across the layers (figure 2.2 B).

3. Completely new abstractions (figure 2.2 C).

2.5.1 Allowing the layers to communicate

A straightforward way to allow information sharing between the layers is to allow them to communicate each other, as depicted schematically in figure 2.2 A. Practically speaking, this means making the variables at one layer visible to the other layers at run-time. Notice that under strictly layered architectures, every layer manages its own variables and its variables are of no concern to other layers.

There are many ways in which the layers can communicate with one another. For instance, protocol headers may be used to allow flow of information between the layers. Alternatively, the extra "inter-layer" information could be treated as internal packets. Reference [60] presents a comparative study of several such proposals and goes on to present another such proposal, namely, the Cross-layer signaling shortcuts (CLASS). CLASS allows any two layers to communicate directly with one another. Similarly, the Hints and Notifications proposal discussed in [45] makes network layer the hub of inter-layer communication.

These proposals are appealing in the case where just a few cross-layer information exchanges are to be implemented in systems that were originally designed in conformance with the layered architectures. In that case, one can conceivably "punch" a few holes in the stack while still keeping it tractable. However, in general, when variables and internal states from the different layers are to be shared between the different layers as prescribed by such proposals, a number of implementation issues relating to managing shared memory spaces

between the layers may need to be resolved.

2.5.2 A shared database across the layers

The other class of proposals propose a common database that can be accessed by all the layers, as illustrated in figure 2.2 B. See for instance [61], [30]. In one sense, the common database is like a new layer, providing the service of storage/retrieval of information to all the layers.

The shared database approach is particularly well suited to vertical optimizations. An optimization program can interface with the different layers at once through the shared database. The main issue here is the design of the interactions between the different layers and the shared database.

2.5.3 Completely new abstractions

The third set of proposals present completely new abstractions, which we depict schematically in figure 2.2 C. Consider, for example, the proposal in [62] which presents a new way to organize the protocols—in heaps—and not in stacks as done by layering. A similar concept is discussed in [63] that breaks the protocols into smaller units called building blocks. The overall behavior of the network is then described by the interactions between the different building blocks.

Such novel organizations of protocols are appealing as they allow rich interactions between the building blocks of the protocols. Hence, potentially they offer great flexibility in designing and optimizing protocols. However, they also create new considerations as discussed in [62].

2.6 A unified platform

We took stock of the ongoing activity in the area of cross-layer design in sections 2.4 and 2.5. We found that in the literature today, there are several different cross-layer design proposals as well as some initial proposals on how the new architectures should look like. The following key insights can be extracted from the ongoing work regarding the cross-layer design for wireless ad-hoc networks:

- Cross-layer design is required between network and MAC layers—the functionalities of routing and medium-access control cannot be designed in complete isolation.
- An interface from the physical layer to the MAC layer can allow channel adaptive schemes to be employed, resulting in performance improvements.
- Notifying the transport layer protocol explicitly about the underlying network status helps to improve the performance of the end-to-end flow control protocol. Hence, interfaces from the network and/or link layer to the transport layer are needed.

We now discuss the basis for these cross-layer couplings and identify the relevant research papers that have been making inroads into these areas.

2.6.1 Coupling between Network and MAC layers

Cross-layer design is required between the network and the MAC layer because routing and MAC interact, implying that the behavior and performance of one depends upon the other [64], [65]. Hence, these two functionalities cannot be designed in complete isolation from one another.

In cross-layer design between these two layers, two important considerations need to be taken into account. Firstly, the design of the routing and MAC

functionalities has to be done keeping in mind the impact of the choices on the end-to-end performance (see for instance [66], [67], [68]). Secondly, the role of power control has to be balanced between power control as a means to avoid MAC layer contention on a per-transmission basis (the MAC layer power control approach [52]) and power control with the view of topology control (the network layer power-control approach [53]).

2.6.2 Channel knowledge at the MAC layer

An interface from the physical layer to make the channel knowledge available at the MAC layer potentially allows the nodes to make opportunistic usage of the channel. This is an attractive option with multiple-rate physical layers common in the current wireless devices. In recent times, a number of papers have been discussing such channel-adaptive MAC protocols. In fact, one sees a clear evolution of the channel-adaptive MAC layer idea from receiver based auto rate (RBAR) in [36] where the receiver picks the best possible rate and informs the transmitter during the RTS-CTS handshake; to opportunistic auto rate (OAR) in [39] where the transmitter, if sensing the channel to be in a good state, sends multiple packets back-to-back; to medium access diversity (MAD) in [40], where the channel aware MAC layer scheme also incorporates spatial diversity. As before, the interaction of such channel-adaptive MAC layer schemes with the other higher layer metrics needs to be carefully studied for systematic protocol development.

2.6.3 Explicit notifications to the Transport layer

The role of the transport layer is to provide an end-to-end connection. Given that in a wireless ad-hoc network, packet loss occurs for reasons other than

congestion (e.g. errors on the channel, node mobility etc.), there is a need for explicit notifications to be provided to the transport layer from the lower layers of the stack. Examples and further justifications of such notifications can be found in [17] and [38] and the references therein.

2.6.4 Other couplings

Apart from the three couplings listed above, there is always an avenue for balancing processing across different layers. For instance, error-correction at the the link and the transport layers needs to be balanced. Likewise, static/dynamic vertical calibration may be needed for optimizing the energy and delay performance of the overall stack. Also, security issues need to be handled across the layers in a holistic manner (see for example [5] and [69]).

2.7 Open Challenges

In Section 2.6, we extracted the key insights about which layers need to be coupled and how, based on the current literature. The discussion in Section 2.6 is really only a starting point toward identifying how the architectures for wireless networks should look like. There are additionally several open questions, some of which cannot be addressed from a performance viewpoint alone and require a consideration of architectural concerns too. For example:

- What should be the role of the physical layer in wireless networks?
- Is the conventional view of the network—that of a collection of point-to-point links—appropriate for wireless networks?
- How do the different cross-layer design proposals co-exist with one another?

- Will a given cross-layer design idea possibly stifle innovation in the future?
- What are the cross-layer designs that will have the most significant impact on network performance, and hence should be most closely focused on?
- Has a given design proposal been made with a thorough knowledge of the effect of the interactions between the parameters at different layers on network performance?
- Under what network and environmental condition be a particular cross-layer design proposal be invoked?
- Can the mechanisms/interfaces used to share information between the layers be standardized?
- How do we make sure that the new architectures allow innovative usage of the wireless medium that we are likely to see in the future?

We now look at some of these issues in greater detail.

2.7.1 The role of the physical layer

In wired networks, the role of the physical layer has been rather small—that of sending and receiving packets when required to do so from the higher layers. As we have seen, advances in the signal processing at the physical layer can allow it to play a bigger role in wireless networks. Consider, for instance, multi-packet reception capability at the physical layer. It clearly changes the medium-access problem at the MAC layer significantly. Since the MAC layer's functionality is intimately connected to the network layer (and hence to the rest of the stack), we see that signal processing advances at the physical layer promise to have a significant impact on all aspects networking protocol design. Cross-layer designs

relying on advanced signal processing at the physical layer are an interesting research ground for the future.

2.7.2 The right communication model

Wired networks, by their very nature, are essentially a collection of well-defined point-to-point communication links. The same cannot be said about wireless networks because the wireless medium is inherently broadcast and there is no clear-cut concept of a communication link in wireless networks. This gives rise to several new possibilities for communication, one of which we will look at closely in Section 2.8. The more fundamental question that also comes up is whether it still makes sense to "create" links in a wireless network. Some recent work has made use of the inherent broadcast nature of the wireless medium to come up innovative communication schemes for wireless networks. For example, the idea of "hitch-hiking" in wireless networks proposes to use the broadcast medium, along with packet combining at the physical layer, to achieve energy efficiency in multi-casting over wireless networks [70]. Initial works notwithstanding, it remains largely an open challenge to figure out the appropriate communication model for wireless networks.

2.7.3 Co-existence of cross-layer design proposals

An important question to be answered is how different cross-layer design proposals can co-exist with one another. To clarify by example, say the MAC layer in a stack responds to the variation in the channel by adjusting the data rate. The question is, will additionally adjusting the frame length at the link layer, as in [71], help further? How will an over-riding control from, say, the transport layer, trying to control the link layer parameters, interact with these adaptation

loops?

The question of co-existence of cross-layer design ideas is pertinent when it comes to determining whether some cross-layer design proposals can stifle further innovation. Let us say the physical layer and the link layer are optimized for a certain performance metric in a cross-layer design scheme. If this scheme is deployed first, can other schemes that also rely on some (other) cross-layer couplings, or those that assume no-coupling between the link layer and the physical layer be deployed too at a later time?

Designers need to start establishing which cross-layer design proposals may or may not be employed in conjunction with a given cross-layer design scheme, as also stressed in [10].

2.7.4 When to invoke a particular cross-layer design?

One of the stated motivations behind cross-layer design is to achieve the network equivalent of "impedance matching" [17]. The idea is to make the protocol stack responsive to the variations in the underlying network conditions so that an optimal operating point is always maintained.

To achieve such optimal operation, designers need to establish the network conditions under which the proposed cross-layer designs would result in performance improvements. Reference [10] presents an example to illustrate how a cross-layer design involving an iterative optimization of throughput and power leads to a loss in performance. The example underscores the need for designers to establish the network conditions under which their design proposals should and should not be used. Indeed, we also need to come up with efficient mechanisms to make assessments about the relevant state of the network accurately.

2.7.5 Standardization of interfaces

The holy grail of the current interest in cross-layer design arguably is to figure out where the layered architectures of today lack and how the new architectures in the future should look like. A key component of making inroads towards the new architectures is to identify the interfaces between the protocol abstractions. This pursuit is non-trivial, but is intimately related to the cross-layer design activity that we are witnessing today. We captured some initial thoughts in the unified platform but a number of open questions remain. The first question is the boundaries between the between modules of the system. Should we stick to the traditional layer boundaries as in Figs. 2.2 A and B and determine the new interfaces from there, or should we look at completely new boundaries, as in Fig. 2.2C? Or a combination? Then, the interfaces between the modules (layers or otherwise) determine how efficiently can information be shared between them—at what kinds of overheads and delays. This, in turn, determines how effective cross-layer design proposals can be. Hence, a better synergy is needed between the protocol design effort, the implementation concerns and the architecture related efforts. For instance, proposers of cross-layer design relying on back-and-forth information flow between layers or dynamic vertical optimizations need to start considering the impact of delays in the retrieval/updating of information on the protocol performance, and also the overheads associated with a cross-layer design.

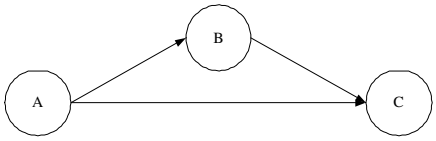

Figure 2.3: A relay channel. The source's transmission is heard by both the relay and the destination. The relay can then transmit some additional data that can help the destination decode the source's message.

2.8 New opportunities for Cross-Layer Design

2.8.1 The broadcast nature of the wireless medium

We have been looking at the ongoing work in the area of cross-layer design. The key driver for the cross-layer design activity that we have thus far focused on is the inadequacy of the reference layered architectures in addressing the peculiarities of the wireless links. In this section, we look at a fundamentally different modality of communication made possible by the wireless medium from the perspective of the layered architectures. The purpose is to highlight an avenue for protocol design that will, much like the wireless links, challenge the layered architectures and require cross-layer design.

The wireless medium is inherently a broadcast medium, as we discussed in Section 1.2.2. Whenever a node transmits any data, the transmission can be heard by all nodes in the vicinity simultaneously. This means that a protocol designer is no longer constrained to view the network as a collection of point-to-point links as is the case in wired networks—the designer can instead design protocols that rely on the inherent broadcast nature of the channel. One such scheme is to design protocols that get the nodes to co-operate with each other in accomplishing data transfer. Consider, for example, the relay network shown in figure 2.3. It consists of a data source, a destination and a relay node. In the traditional networking view, there is either a direct one-hop path from the

source to the destination or a two-hop path comprising the source-relay and the relay-destination links. In the case of wireless medium, a third possibility, namely that of the relay collaborating with the source exists. Basically, when the source transmits, the transmission can be heard by both the relay and the destination. The relay can then send some data that is a function of the data that it received from the source. For instance, the relay can just repeat the data from the source. This has the effect of creating a rate-1/2 repetition code [72, page 74] at the destination. More sophisticated coding schemes can also be collaboratively created. As an example, [73] discusses the distributed turbo-code in a relay network, whereby the source and the relay nodes collaborate to create a turbo-code at the destination. The idea can similarly be extended to the case of multiple relays, resulting in an even more diversity, and hence performance improvement over the traditional point-to-point multi-hop case.

As mentioned above, in this section, we consider the problem of data-transfer between two nodes of an ad-hoc network making use of a cooperation scheme. The question we are addressing is as follows: Do the layered architectures allow protocols that rely on node co-operation as discussed above, or is it inevitable to violate the layered architectures for developing such schemes? The answer, as we will see next, is that violation of the layered architectures in inevitable for incorporating node co-operation into protocol design.

2.8.2 Types of co-operation schemes

For the sake of exposition, we identify two forms of co-operation schemes. We call these two as planned co-operation and unplanned co-operation. In planned co-operation, the co-operation scheme, participating nodes and the roles to be performed by the participating nodes are decided a-priori, *before* the transmis-

sion starts. For instance, in the example of the distributed turbo code on the relay network above, the relay node and the role to be played by it (in this case assisting the source to form a turbo code [73]) have to be decided before the transmission can begin. On the other hand, in the case of *unplanned* co-operation, any node that is in a position to co-operate can do so. Thus, the roles played by the different nodes are not decided a-priori. A simple example of an unplanned co-operation scenario can be constructed as follows: Consider again the relay network in figure 2.3. Instead of always employing the node B as a relay collaborating with node A (the source), one can develop a scheme such that node B only sends information if node C is unable to correctly receive the packets sent out by node A. For instance, one can develop a distributed Automatic Repeat Request (ARQ) protocol such that, if needed, node B can send incremental redundancy to node C. Distributed ARQ results in spatial diversity which can contribute positively to the performance. Such a scheme is considered in [74]. Notice that unlike the case of planned co-operation, the role for the relay node B was not a-priori assigned. Node B co-operated by sending the incremental redundancy solely because it found itself in a position to do so.

We now look at both planned and unplanned co-operation from the layering perspective. We find that incorporating planned cooperation involves significantly more violation of the layered architectures than unplanned cooperation.

2.8.3 Planned Co-operation

Consider the wireless ad-hoc network shown in figure 2.4. Let us say that the node A has to send data to a far-away node G. Like in conventional (point-to-point links) case, the first problem is to establish data paths from the source node to the destination node. This, in fact, is the classic routing problem.

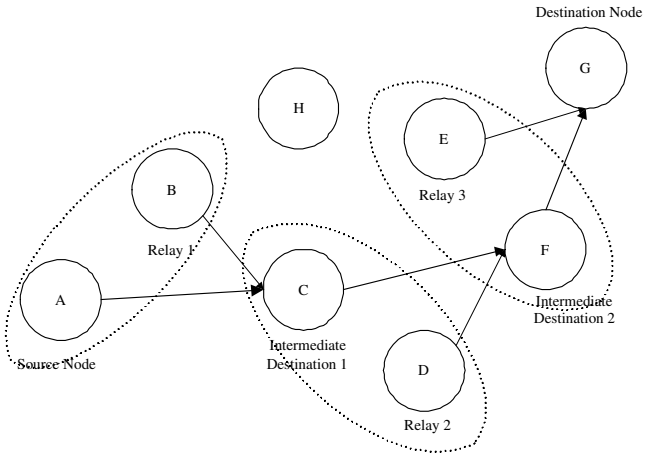

Figure 2.4: Data transfer with node co-operation in an ad-hoc network. Node A is the source and Node G is the destination. Nodes B, D and E act as relays co-operating with nodes A, C and F respectively.

In the case of co-operative communication, the additional task that needs to be performed is to assign which nodes will co-operate and how. We contend that this task is best performed at the network layer since the network layer has the best view of the current network topology. Hence, the new network layer problem comprises firstly, the selection of the data path; and secondly, the selection of the co-operating nodes. Thus, for example, the new network layer protocol may assign node B to be a relay (a co-operating node) and node C to be an intermediate destination, as indicated in figure 2.4. Now, the latter part of the new network layer problem, namely, that of assigning the cooperating nodes, is inevitably coupled to the physical layer problem of designing the error-correcting code for the co-operation scheme. For instance, in the case of multiple turbo codes in [73], the number of relays to be selected in determined by the rate of the intended error-correcting code. Hence, there is an inevitable design-time coupling (a violation of the layered architecture) between the physical layer and the network layer. The new network layer cannot be designed in complete isolation from the physical layer.

Consider next the link layer. The link layer, in the traditional point-to-point networking scenario, provides the service of getting packet across a single hop to the network layer. In the case of co-operation, there is no real link. However, one can still view the packet as moving across the network through a set of intermediate nodes which form the "links". For instance, we can view the transmission between nodes A and C in figure 2.4 as traversing a link. Since the task of establishing the roles of the different nodes (e.g. which node is an intermediate destination, which node is/are the relay(s)) is done by the network layer, we need this information to be transferred from the network layer to the new "link-layer". In fact, the MAC layer also needs this information, since the order in which the nodes access the channel is a function of the roles that they have been assigned, which in turn is coupled with the physical layer problem of code design. Hence, we need an interface for run-time information sharing between the network-layer and the link-layer and there is design-time coupling between the physical layer and the link layer. Also, there needs to be run-time interaction between the link layer and the physical layer to implement the combining of the signals that are received from different sources (e.g. from the source and the relay) in non-overlapping time slots. A schematic representation of the architecture violations needed is shown in figure 2.5.

2.8.4 Unplanned Co-operation

Next, consider unplanned co-operation. In this case, the co-operating nodes are not decided prior to the transmission. Thus, the routing problem is the same as that of the traditional routing problem and does not need to be coupled with the physical layer. On the other hand, as highlighted in [74], the realization of the unplanned co-operation translates to a complicated link/MAC layer design

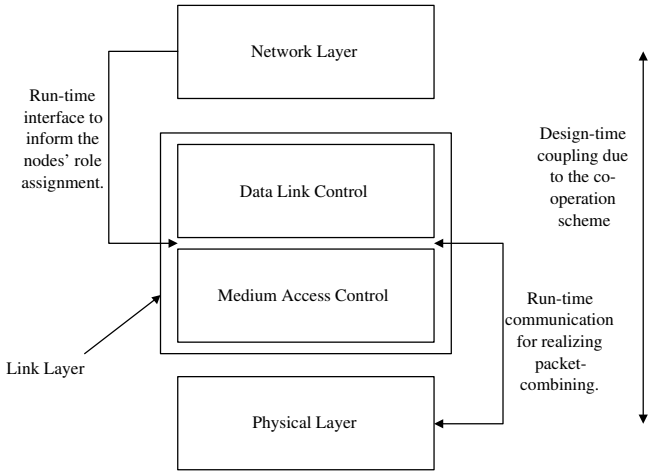

Figure 2.5: An assessment of the architecture violations needed to allow protocols that rely on planned co-operation in the network.

problem. Furthermore, there still needs to be run-time communication between the link layer and the physical layer so that packet-combining etc. can be realized. Hence, unplanned cooperation also requires architecture violations, albeit to a lesser extent as compared with planned cooperation.

2.8.5 Summing up

Node cooperation, as we have been discussing, is well established in the information theory community (see for instance, [75, page 428]). However, incorporating it into network protocol design is a relatively new area. A number of issues are yet to be resolved—a brief discussion of some of these can be found in [74]. It is clear that node cooperation strains the layered architectures, as we have seen above. Cross-layer design problems arising from the protocol design with node cooperation, when combined with "traditional" cross-layer design motivated by the peculiarities of the wireless links promises to be a fertile research ground for cross-layer design activity in the future.

2.9 Conclusions

We set out to identify the road ahead for cross-layer design. A natural starting point was to assess where we are today. To this end, we took stock of the current activity in the area of cross-layer design. The different interpretations of cross-layer design were brought together by creating a taxonomy. Looking forward, we distilled some existing results into a unified platform and made a preliminary assessment of which layers need to be coupled and in what ways. We moved on to highlight some open challenges in this area by taking into account both performance and architecture viewpoints. Finally, we presented a new possibility for communication in wireless networks that would inevitably require cross-layer design.

Our key message to the community is to move forward with a more holistic view of the situation. Cross-layer design is intimately linked to several issues that span communications, networking, architecture, implementation etc. and hence should be seen in the right context. Also, while proposing cross-layer design proposals to address the issues raised by wireless links, the designers should also keep an eye for the new opportunities being created by the wireless communication networks. A systematic study of cross-layer design problems, combined with the fundamentally different communication paradigms presented by wireless networks, will go a long way in defining the communication architectures of the future.

Chapter 3

Physical Layer Design with a Higher Layer Metric in Mind

3.1 Introduction

In the previous chapter, we looked at the various facets of cross-layer design. We created a taxonomy of the ongoing work in the area of cross-layer design. We identified design-time coupling between layers, without the creation of additional interfaces, as a category of cross-layer design. In this chapter, we present a specific instance of such a cross-layer design involving the physical layer and the link layer.

In particular, we are concerned with the physical layer (PHY) design for a point-to-point communication system. We assume a Stop-and-Wait Automatic Repeat Request (ARQ) [76, page 459] at the link layer and assume a stationary channel, characterized by a fixed probability of error. Examples include an Additive White Gaussian Noise (AWGN) channel. We come up with quantitative guidelines on the PHY processing such that metrics like the link layer average service time and the link layer average delay are favorably affected.

Our approach hinges on comparing an original system with a modified system, where the modified system is different from the original system in terms of its PHY processing. We abstract out from the details of the PHY processing by specifying the packet duration and the probability of error. As such, we capture the change of PHY processing in the modified system by making its packet duration and packet error probability different from the original system. Through the comparison between the two systems, we come up with parametric quantitative conditions on the ratio of the packet duration in the modified system to the original system and the ratio of the packet error probability in the two systems such that the performance in the modified system is better (implying a lower average service time in the modified system). Using queueing theoretic arguments, we also explore the connection between average service time and average delay with Poisson arrivals.

We consider two special cases of our parametric conditions. First, we consider the case when the baseline system represents an uncoded system while the modified system represents a system with Forward Error Correction. This gives us a necessary and sufficient condition on code rate and coded packet error probability for the coded system to exhibit superior performance as compared to the uncoded system. Next, we consider the case where the difference between the baseline system and the modified system is that of a constellation size. Again, this lets us come up with a necessary and sufficient condition that must be satisfied for a change in the constellation size to improve performance.

This work is similar to the several recent works that have dealt with the problem of choosing appropriate physical layer processing on the basis of a quantitative study of the impact of PHY processing on the higher layer metrics. In [77], the effect of erasure-correcting codes on TCP throughput is studied. An optimal trade-off between the transmission power and redundancy of Reed-

Solomon codes with the aim of maximizing the end-to-end TCP throughput is presented in [78]. A similar problem is considered in [79] where the physical layer parameters are chosen from a set of values with the aim of maximizing TCP throughput. Link-layer metrics are considered in [56] and [80]. In [56], the impact of physical layer processing on average delay in Stop-and-Wait ARQ is considered [1]. In [80], a fluid analysis is performed to analyze the effect of forward error correction on the delay statistics in hybrid-ARQ. Simulation based studies of wireless TCP performance with forward error correction at the physical layer are presented in [81], where two BCH codes (approximately rate-1/2 and rate-1/3) are considered. More recently, [82] and [83] deal with studying the effect of multiple antennas at the physical layer on TCP throughput.

In the rest of this chapter, we first provide a brief review of the background needed for the results presented in this chapter in Section 3.2. We emphasize that Section 3.2 has been added for the sake of completeness and the treatment is not meant to be exhaustive. Rather, we only include and explain those terms and concepts that we use later in the chapter. More details may be obtained by following the references. Next, we describe the system model in Section 3.3. Then, in Section 3.4, we describe the modified system and come up with the parametric condition under which the modified system exhibits a lower average service time as compared to the original system. We relate the parametric condition to PHY processing in Section 3.5. In Section 3.6, we look at the average delay, whereby we study the delay behavior of the modified system with Poisson arrivals. Next, in Section 3.7, we relate the delay results to physical layer processing. We look at illustrative numerical examples in sections 3.7.1.1 and 3.7.2.1, before presenting the conclusions in Section 3.8.

[1]Part of the material presented in this chapter has been reported in [56].

3.2 The background

Before moving on with the analysis, we first briefly present the relevant background and concepts that are used later in the chapter. In the rest of this section, we first look at physical layer processing in 3.2.1, where we look at the basic concepts related to digital modulation in Section 3.2.1.1, the effect of a band-limited communication channel on digital transmission in Section 3.2.1.2, forward error correction in Section 3.2.1.3, and a brief discussion of error-probability performance of Binary phase-shift-keying in Section 3.2.1.4. Next, in Section 3.2.2, we look at link layer processing and discuss the basic ARQ mechanisms, with a particular focus on Stop-and-Wait ARQ. Finally, in Section 3.2.3, we discuss the Pollazcek-Khinchin formula for the average queueing delay in an M/G/1 queue.

A reader familiar with the aforementioned concepts may want to skip the rest of this section and move straight to Section 3.3, taking note of the following observations:

Observation 3.1. *We use the term "bandwidth" to mean the amount of spectrum available for data transmission. It is thus measured in Hertz(Hz). This is unlike the usage in the networking community, where "bandwidth" often implies data rate in, say, bits/sec.*

Observation 3.2. *Whenever we use a term like "symbol-duration" or "symbol-rate", what we mean by the word symbol is a modulated data symbol.*

3.2.1 Physical Layer Processing

The term physical layer processing, as used in this thesis, refers only to the baseband processing [12, page 151]. This includes operations like forward-error

correction and digital modulation at the transmitting end, and demodulation and decoding at the receiver end. We shall now briefly look at these operations and introduce some other related concepts and terminology that will be used in our later discussion.

3.2.1.1 Digital Modulation

Wireless communication devices work by radiating information-bearing electromagnetic waves. Information is "inscribed" onto a carrier-wave by the process of modulation, which means changing some characteristic—like the amplitude, the phase or the frequency—of the carrier according to the data to be transmitted [84, page 260]. We will be concerned only with digital modulation.

Definition 3.1. Digital modulation *refers to inscribing digital information onto a carrier wave by changing the characteristics of the carrier wave in a discrete manner.*

Thus, there is a notion of a symbol in a digital modulation and the characteristic being changed stays constant for the symbol duration, before undergoing a discrete change in the next symbol. For example, in binary phase-shift keying (BPSK) modulation [85, page 174], binary information is inscribed onto the carrier wave by sending the carrier wave at one of the following two phase-shifts: 0 or π. Each of the phase shifts lasts for the symbol duration, after which a new phase shift is taken on. The same is true for other digital modulation methods too. A pictorial representation of some digital modulation methods—vividly clarifying the notions of symbol and symbol duration—is available at [85, page 174].

At the baseband level, digital modulation basically maps a bit sequence to a complex-valued output. In particular, a so-called M-ary digital modulation,

M being a positive integer that is typically a power of 2, takes a sequence of $\log_2 M$ bits as its input and produces one of the M possible complex-valued outputs. The output produced for a given input bit sequence is unique and is determined by the mapping between the bit sequences and the complex values (see for example [12, page 171]).

Definition 3.2. *A 2-dimensional diagram obtained by plotting the complex output values of a modulator on the complex plane is called the* signal space diagram *or a* constellation diagram *(see [12, page 173– 175] for several examples).*

Definition 3.3. *The total number of outputs from a digital modulator is called the* constellation size.

Thus, the constellation diagram for the BPSK modulation consists of just two points: one at +1 and another one at -1; and the constellation size of BPSK modulation is 2, as there are only two possible outputs that a BPSK modulator produces. The significance of the constellation size is that, for a given number of bits to be transmitted, the larger the constellation size, the smaller the number of modulated symbols needed. Of course, the size of the constellation has implications on the performance. Generally speaking, as the constellation size in increased without changing the transmitted power, the constellation points get closer to each other in the constellation diagram. Compare, for example, the placement of constellation points in the Binary PSK and 8-PSK [12, page 173], a modulation that sets the phase of the carrier-wave to one out of 8 different values. As the points get closer to one another on the constellation diagram, noise on the channel causes more errors. Hence, for a given signal-to-noise ratio, one can expect a higher probability of symbol error for a modulation with a larger constellation size.

We primarily deal with BPSK modulation which was described above. In

Section 3.7.2.1, we also look at the Quadrature phase shift keying (QPSK) modulation. Conceptually, QPSK and BPSK are similar—both work by modulating the phase of the carrier wave. The difference between the two, as the name suggests, is that in QPSK, there are four possible phase values unlike BPSK, which has only two. For example, the four phase values may be as follows: 0, $\pi/2$, π and $3\pi/4$ [12, page 173]. Hence, the constellation diagram of QPSK has 4 points. In terms of performance analysis, QPSK can be viewed as two orthogonal BPSK modulated carriers [12, page 268].

3.2.1.2 The significance of finite bandwidth

Communication channels have a finite bandwidth[2]. This influences how a digitally modulated signal is received at the baseband level. As we have seen, digitally modulated signals contain discrete changes in the characteristics of the carrier-wave happening every T seconds, T being the symbol duration. At the baseband level, this is equivalent to a sequences of pulses which exhibit an abrupt and sharp change in their value every T seconds (see for example the pictorial representation of the baseband equivalent of pulse-amplitude modulation at [12, page 172]). When such a signal passes through a channel that has a finite bandwidth, the tails of the pulses get smeared. The result is what is termed as inter-symbol interference (ISI), implying that a received symbol interferes with the detection of several other symbols [85, page 136], [12, page 555]. The consequence can be a significant loss in performance, unless ISI is mitigated through signal processing or pulse design.

Whether or not a channel causes ISI depends upon the relation between the equivalent baseband channel bandwidth W[3] and the symbol duration T [12,

[2]Recall that by bandwidth, we mean the amount of spectrum.

[3]Using complex baseband equivalent representation, a bandpass channel of bandwidth $2W$ can be shown to be equivalent to a baseband channel of bandwidth W [84, page 109].

page 558]. The following result holds:

Result 3.1. *If $T < \frac{1}{2W}$, ISI is inevitable. If $T \geq \frac{1}{2W}$, it is possible to transmit data in a way such that the receiver experiences no ISI. Achieving ISI-free reception involves designing the signaling pulse such that the so-called Nyquist's criterion for zero ISI [12, page 557] is satisfied.*

One popular method to achieve ISI free transmission is to employ what is called a raised-cosine pulse [12, page 560], [85, page 139]. Mathematical expressions for the raised-cosine pulse and its frequency spectrum can be found at the references (see for example, [12, page 560], [85, page 139]). What we are interested in is a parameter called as the roll-off factor of the raised cosine pulse.

Definition 3.4. *The* roll-off factor *r of a raised-cosine pulse, satisfying $0 \leq r \leq 1$, specifies how much excess bandwidth, as compared to $\frac{1}{2T}$, is required, if the raised-cosine pulse with a roll-off factor r is employed [12, page 560].*

Result 3.2. *Employing a raised-cosine pulse with roll-off r requires bandwidth equal to $(1 + r)\frac{1}{2T}$. Equivalently, a channel of bandwidth W can support a minimum symbol duration given by $(1 + r)\frac{1}{2W}$ [12, page 560].*

As an example, let us say that the bandwidth W is 4 kHz and $r = 1$. In that case, the smallest symbol duration for which ISI-free reception is possible is 0.25 ms, resulting in a symbol rate of 4 k symbols/s.

3.2.1.3 Forward Error Correction

Forward error correction, or coding in short, refers to the addition of redundant bits to the data stream. The motivation behind adding the redundant bits is to provide resilience against the errors that happen as the data is transmitted over

a noisy channel. Excellent introductions to the various aspects of forward error correction, including several examples of error-correcting codes and decoding algorithms, can be found at [85, page 304] and [12, page 416]. In fact, coding theory is a well-developed field and there are several books—for example [72], [76], [86]—that treat coding theory in good detail. We shall take a snapshot of some results for a class of codes termed as linear block codes [72, page 68] in Section 4.2. As far as the material in this chapter is concerned, we just need a fair understanding of the term code rate and a familiarity with what block codes are. We discuss these now.

Definition 3.5. *A* binary block code *maps a k-bit binary string (referred to as a k-tuple) to an n-bit binary string (referred to as an n-tuple), where $n \geq k$. The code is referred to as an (n, k) code.*

Definition 3.6. *The* rate *of a code is a measure of the amount of redundancy that is added. The rate of an (n, k) code is given by the ratio k/n.*

If the symbol duration and the constellation size are kept unchanged, coding results in an increase in the duration of the packet. This simply follows from the fact that coding increases the total number of bits to be transmitted, thanks to the added redundant bits. It should be clear that the increase in packet duration is related to the code rate—the lower the code rate, the more the increase in the packet duration.

Definition 3.7. *An (n, k) block code is called a* linear *(n, k) code if and only if its 2^k code words form a k-dimensional subspace of the vector space of all the n-tuples over the (Galois) field GF(2)[76, page 52].*

There are several known linear block codes which are discussed thoroughly in the references listed above. We will be considering examples of two: the Hamming codes and the Bose-Chaudhuri-Hocquenghem (BCH) codes.

Definition 3.8. *Hamming Codes are single-error correcting linear block codes [12, page 423], [72, page 90]. For every positive integer $m \geq 2$, there exists a $(2^m - 1, 2^m - m - 1)$ Hamming code. As mentioned above, such a code is capable of correcting all single-bit error patterns in a codeword with length equal to $2^m - 1$.*

Definition 3.9. *BCH codes [12, page 437], named after their inventors, are a class of linear block codes that are capable of correcting multiple errors in a block of data. Specifically, for positive integers m $(m \geq 3)$ and t, there exists a $(2^m - 1, k)$ binary BCH code capable of correcting all error patterns of weight t or less such that $k \geq 2^m - 1 - mt$.*

Hamming codes are a special class of BCH codes, obtained for the case of $t = 1$.

3.2.1.4 Error probability performance

The performance of digital communication systems is usually measured in terms of a normalized version of the signal-to-noise ratio, termed as E_b/N_0 (Pronounced as "E B on N naught" or "EbNo")[85, page 117]. E_b represents the energy per bit and N_0 is the noise power spectral density [85, page 117]. E_b/N_0 is an appealing metric for benchmarking the performance of digital communication systems because it is able to indicate how well a system performs at the bit level—it gives an indication of the energy that needs to be spent in transferring *one* bit of data at a certain probability of error [85, page 118].

Result 3.3. *The expression for the probability of bit error p_b for BPSK modulation over an AWGN channel without inter-symbol interference is is given by*

$$p_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \tag{3.1}$$

where $Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^\infty e^{-t^2/2} dt$ *is the well-known Q-function [12, page 40]. A derivation of this result can be found at [12, page 255].*

It is often of interest to compare the probability of error that two different systems exhibit. Consider, for example, a comparison between an uncoded system and a system which employs a rate R_c code.

Observation 3.3. *We use* E_b/N_0 *to refer to the energy per* information *bit, that is, uncoded bits.*

As we saw earlier, if the constellation size and the symbol duration are kept unchanged, it takes more time to transmit a packet in the system with coding. Hence, if while making the comparison, the transmit power is kept same in both the systems, the energy spent per packet is higher in the system with coding and, as a consequence, so is the E_b/N_0 since with coding, more energy is being spent per bit of information transmitted. On the other hand, if the comparison has to be in terms of the E_b/N_0, the transmit power in the coded system has to be lowered. This will make the channel more noisy, as far as the transmission of the coded bits through the channel is concerned. We shall compare systems by keeping the E_b/N_0 constant, unless stated otherwise.

3.2.2 Link Layer Processing

Link layer provides service to the network layer—that of getting a packet across a single hop in a network. The link layer is generally divided into two sub-layers: the data link control (DLC) sub-layer and the medium access control (MAC) sub-layer[23, page 24]. The MAC sub-layer is needed to negotiate the

access to the medium between different users when the communication medium is shared, for example, in ad-hoc wireless networks. The DLC sub-layer provides the service of link-level error control to the higher layers. Since we deal with a point-to-point communication system, we will primarily be dealing with the data-link control portion of the link layer.

The two primary mechanisms at the DLC layer are error-detection [23, page 57] and re-transmission [23, page 64].

Definition 3.10. Error detection*, as the name suggests, refers to the process of detecting if a received packet contains errors.*

Much like error correction (discussed in Section 3.2.1.3), error-detection is also accomplished by adding extra bits to the data packet. The difference is that, unlike error correction where the goal is to be able to both detect *and* correct the errors, the goal in error detection is only to detect the errors. Codes that are used for error correction can also be used for error detection (see for instance [72, pages 9, 240, 248]). Alternatively, codes can be specifically devised for error detection only, for example the Cyclic-Redundancy-Check (CRC) codes [23, page 61]. A good discussion and several examples of error detecting codes are available at [23, pages 57–64].

Error detection allows the DLC entity at the receiver to determine whether or not a packet has been received in error. If a feedback channel is available from the receiver to the transmitter, the DLC entity at the receiver may request its counterpart at the transmitter to re-send the packet. This is typically done by sending a positive acknowledgement—called an ACK—confirming the error-free receipt of a packet; or a negative acknowledgement—called a NAK—informing the transmitting DLC entity about the erroneous receipt of the packet.

Definition 3.11. Automatic Repeat Request (ARQ) *is a mechanism whereby*

the receiver data-link control entity requests the transmitter data-link control entity to re-transmit a packet in which an error has been detected at the receiver; and the transmitter abides by the request by sending the packet again.

There are three basic types of ARQ: Stop-and-Wait ARQ, Selective Repeat ARQ and Go-Back-N ARQ. The schemes are conceptually similar to one another—the difference is in the details of how the acknowledgements and the re-transmissions are actually carried out. Thorough coverage of all the three types of ARQ schemes are available at [72, page 393], which takes a more coding-theoretic approach and [23, page 64], which takes a more networking type approach in covering the material. The basic ARQ mechanisms can also be combined with forward error correction to yield what are called Hybrid ARQ schemes. We refer the reader to [72, page 409] for further details and the different types of hybrid ARQ schemes.

We will only be dealing with Stop-and-Wait ARQ. In Stop-and-Wait ARQ, the transmitter releases a packet and *stops* to *wait* for the response from the receiver before releasing the next packet—hence the name stop-and-wait. Depending on the outcome of the previous transmission attempt, the transmitter either re-transmits the old packet or transmits a new packet. Stop-and-Wait ARQ is simple to implement and analyze.

Definition 3.12. *In a stop-and-wait ARQ system, we define* round-trip time *as the time taken from the start of the transmission of a packet at the transmitter to the receipt of the ACK or NAK about the packet outcome.*

The round-trip time comprises the propagation delay on both the forward and the reverse channels, the processing delay at the receiver, as well as the transmission delay on the forward channel, that is, the packet duration. It can be shown that if the round-trip time is much larger than the packet duration—

for example in satellite networks—stop-and-wait ARQ makes inefficient usage of the communication channel [72, page 397]. Other types of ARQ schemes are more attractive in such situations (see [72, page 400] for an illustrative example).

The final concept about Stop-and-Wait ARQ[4] is what is called the persistence of the ARQ scheme [59]. An ARQ scheme is said to be highly persistent if the transmitter continues the retransmission of a packet until the receiver has positively acknowledged its receipt. Contrast this with what is called a truncated ARQ, whereby the transmitter only retransmits a packet for a fixed number of times. If the packet is still not positively acknowledged, the transmitter just drops it and proceeds to the next packet.

Observation 3.4. *In a highly persistent ARQ scheme, the packet passed on to the higher layers from the receiver DLC entity is always error-free.*

3.2.3 Delay results on M/G/1 queues

An M/G/1 queue is a single server queue with memoryless (Poisson) arrivals and a general service time distribution [23, page 186]. An M/G/1 queue specializes to an M/M/1 queue when the service time distribution is exponential [23, page 162].

Definition 3.13. *The* average queueing delay *is the average time spent by a packet while it is waiting to be served.*

Result 3.4. ***Pollazcek-Khinchin Formula*** *The average queueing delay in an M/G/1 queue is given by the Pollazcek-Khinchin (P-K) formula . The P-K formula states that*

$$T_Q = \frac{\lambda \bar{X^2}}{2(1 - \lambda \bar{X})} \tag{3.2}$$

[4]the idea can be applied to other types of ARQ too.

where T_Q represents the average queueing delay, λ is the average arrival rate, $\bar{X}$ is the average service time and $\bar{X^2}$ is the second moment of the service time. For the system to be stable (implying finite mean queueing delay), we must have $\lambda < 1/\bar{X}$ and $\bar{X^2} < \infty$.

Proof. Available at [23, page 186]. □

Observation 3.5. *The dependence of average queueing delay on the second moment of the service time means that, in general, a lower average service time does not imply a lower average queueing delay [23, page 190].*

3.3 System Model

Having gone through the relevant background in Section 3.2, we are now in a position to describe our system model and proceed with the analysis.

3.3.1 Description of the model

We consider a Stop-and-Wait ARQ system [76, page 459]. We assume that data packets arrive from the higher layers in a stochastic manner. At the transmitter, the packet contents are first encoded with an error detecting code (for example a cyclic redundancy check code [72, page 121]) and then with a forward error correction code. Subsequently, after modulation, the encoded data packet is transmitted over the channel. At the receiving end, demodulation, error correction and error detection are performed in the respective order. If an error is detected, the receiver requests a re-transmission of the packet via a feedback channel, as shown schematically in figure 3.1.

We assume that the error detecting code works without any logical errors. This means that a retransmission is requested if and only if the decoder is

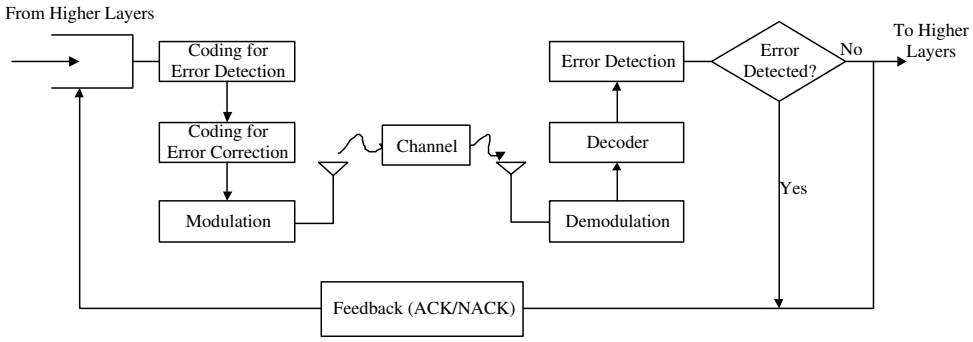

Figure 3.1: The system model under consideration.

unable to correct all the errors introduced on the channel. We further assume a noiseless feedback and a highly persistent ARQ [87], which, as we saw in Section 3.2.2, means that a packet is re-transmitted until its receipt has been positively acknowledged. We assume an additive white Gaussian noise channel. Hence, the channel gain remains constant from symbol to symbol. The additive noise process is independent from symbol to symbol and follows a Gaussian distribution with zero mean.

3.3.2 A discussion of the assumptions

The assumption about perfect error detection is reasonable, given the very reliable performance achieved by the commonly used error detecting codes like the Cyclic Redundancy (CRC) codes [72, page 121].

If the round-trip time is much larger than the packet duration (for example in satellite links), using stop-and-wait ARQ results in severe throughput penalty. Hence, by considering Stop-and-Wait ARQ, we are making a tacit assumption that the round-trip time is almost the same as the packet duration. This implies that the feedback is almost instantly available. In a local area network where nodes are located close to the receivers (e.g. a group of people in a meeting room), this is reasonable. Indeed, in the acknowledgment mode of the IEEE

802.11 standard, Stop-and-Wait ARQ is used and the receiving station needs to compulsorily acknowledge the receipt of a packet within a short time of the packet reception. This practical consideration, coupled with the relatively easier analysis that Stop-and-Wait ARQ admits itself to, motivates us to consider Stop-and-Wait ARQ. The approach that we take can be used for other ARQ systems too.

Considering a highly persistent ARQ mechanism eases the analysis without changing the essence of the problem at hand. It is noteworthy that high persistence at the link layer has been proposed for shielding the end-to-end connections from the link-level errors [87]. Assuming that the forward and the reverse channels are independent is reasonable as the forward and the reverse channels are usually realized by using separate time slots or separate frequency bands. Furthermore, since the reverse channel in our case only carries a single bit acknowledgement which can be protected with error-correcting codes, it is reasonable to assume that the reverse channel is error-free.

We assume that the froward channel is time-invariant. Considering the case of time-varying channels thus remains a natural avenue of extension of the present work.

3.3.3 Metric of interest

From a queueing theoretic viewpoint, one can view the combination of the link layer and the physical layer as "serving" the packets from the higher layers.

Definition 3.14. *We define the* service time *as the time taken between the removal of a packet from the link layer buffer at the transmitting end and the delivery of the packet to the higher layer at the receiving end.*

A timing diagram for the Stop-and-Wait ARQ process is shown in figure 3.2.

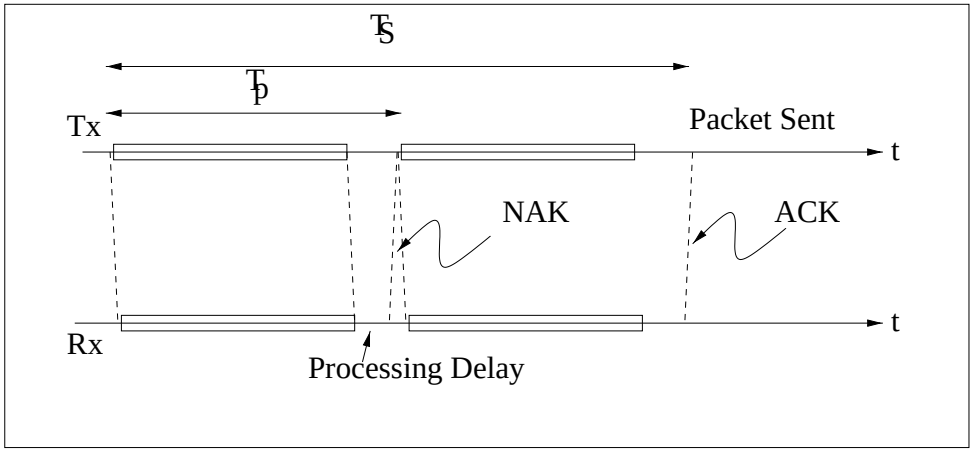

Figure 3.2: Timing diagram of the ARQ System. The packet in the illustration requires one retransmission.

In light of the assumptions discussed in Sections 3.3.1 and 3.3.2, the service time T_s has a discrete geometric probability density function (pdf) [88, page 131] given by

$$P\left(T_s = kT_p\right) = p^{k-1}\left(1-p\right) \qquad k = 1, 2, \ldots \tag{3.3}$$

where k represents positive integers, T_p is the round-trip time, and p ($p \neq 0$) is the probability of packet error. Notice that since we assume an immediate feedback, the round-trip time is exactly equal to the packet duration. Hence, from here on, we will use the notation T_p to mean the packet duration.

The average service time $\bar{T}_s$ is

$$\bar{T}_s = \frac{T_p}{1-p} \tag{3.4}$$

where T_p is the packet duration, and p ($p \neq 0$) is the probability of packet error. We are interested in evaluating the impact of PHY processing on the average service time $\bar{T}_s$ given in (3.4).

3.4 A Modified ARQ System

Next, with reference to the ARQ system described above, we consider a system with modified parameters. Suppose the round-trip time is given by $T_p^{'}$, and the probability of a packet needing re-transmission is given by $p^{'}$, where $T_p^{'}$ and $p^{'}$ are related to T_p and p as

$$\frac{T_p^{'}}{T_p} = \alpha \qquad \text{and} \qquad \frac{p^{'}}{p} = \beta \tag{3.5}$$

where α and β are positive real numbers satisfying $\alpha > 0$ and $0 \leq \beta \leq \frac{1}{p}$ respectively. The bounds on α and β follow because $\alpha T_p > 0$ and $0 \leq \beta p \leq 1$.

The parameters α and β capture the changes caused to the ARQ system due to changes in the PHY processing, for example, due to error-correction coding and/or change of constellation size. This is because as we saw in Section 3.2, changes to PHY layer, for example by coding and/or changes in the constellation size, change the duration of the packets as well as the error-probability. We are interested in evaluating the impact of α and β on $\bar{T}_s$ in (3.4).

3.4.1 Impact of α and β on Average Service Time

The average service time in the modified system is

$$\bar{T_s}^{'} = \frac{T_p^{'}}{1 - p^{'}} = \frac{\alpha T_p}{1 - \beta p}. \tag{3.6}$$

Definition 3.15. *Define $\nu(\alpha, \beta, p)$ as the ratio of the average service time in the original and the modified systems. That is,*

$$\nu(\alpha, \beta, p) = \frac{\bar{T}_s}{\bar{T_s}^{'}} = \frac{1 - \beta p}{\alpha(1 - p)}. \tag{3.7}$$

Observation 3.6. *$\nu(\alpha, \beta, p) > 1$ indicates a lower average service time in the modified system characterized by α and β, when the probability of packet error in the original system is p.*

Lemma 3.1. *$\nu(\alpha, \beta, p) > 1$ if and only if $\beta < \frac{1-\alpha(1-p)}{p}$.*

Proof. Since $\alpha > 0$, $\beta < \frac{1-\alpha(1-p)}{p} \Rightarrow \frac{1-\beta p}{\alpha(1-p)} = \nu(\alpha, \beta, p) > 1$.

Conversely, $\beta \geq \frac{1-\alpha(1-p)}{p} \Rightarrow \frac{1-\beta p}{\alpha(1-p)} = \nu(\alpha, \beta, p) \leq 1$. □

3.5 Relating to Physical Layer Processing

3.5.1 Forward Error Correction

As discussed in Section 3.2.1.3, employing a rate R_c error-correcting code, without changing the constellation size and the symbol duration, increases the packet duration by a factor of approximately $1/R_c$, neglecting any extra symbols (for example for synchronization etc.) and the processing delay.

Definition 3.16. *With respect to an uncoded system, we define* Service Time Improving (STI) *codes as forward error correcting codes that, when employed, reduce the average service time in the Stop-and-Wait ARQ system with the constellation size and the symbol duration kept unchanged.*

Theorem 3.1. *A rate R_c code is an STI code if and only if*

$$p_c < \frac{R_c - 1 + p_u}{R_c} \tag{3.8}$$

where p_u and p_c are the probabilities of packet error for uncoded and coded transmissions respectively.

Proof. This result follows from Lemma 3.1 by replacing α by $1/R_c$, p by p_u and β by p_c/p_u. □

Corollary 3.1.1. *All STI codes satisfy*

$$R_c > 1 - p_u. \tag{3.9}$$

Proof. $R_c \leq 1 - p_u \Rightarrow \frac{R_c - 1 + p_u}{R_c} \leq 0 \Rightarrow p_c \geq \frac{R_c - 1 + p_u}{R_c}$. Hence, from Theorem 3.1, if $R_c \leq 1 - p_u$, the code is not an STI code. Thus, (3.9) is a necessary condition for a code to be an STI code. □

We now briefly look at the intuition behind the results in Theorem 3.1 and Corollary 3.1.1. Basically, every code rate is associated with a certain increase in packet duration. Hence, a coded system exhibits a lower average service time if and only if it exhibits a *small enough* probability of error with respect to the uncoded system. How small an error probability is small enough is captured in Theorem 3.1. Also, as a consequence, if the rate of the code is too low, the increase in the packet duration is so much that no matter how much the probability of error is reduced (by coding), the average service time only increases. How low a rate is too low for a given uncoded packet error probability is captured in the Corollary 3.1.1. Such dual effect of coding has also been mentioned elsewhere in the literature—see for example, [77], [80].

3.5.1.1 Numerical example

Consider a scenario where packets are of length $L = 120$ bits, and the Signal-to-Noise ratio per information bit, E_b/N_0, is 5 dB. Considering a channel of bandpass bandwidth 4 kHz, and raised-cosine filter with roll-off of 1 [12, page 560], we obtain $T_p = 0.06$ s. This is because, a bandpass bandwidth of 4 kHz is equivalent to a baseband bandwidth of 2 kHz, and with a raised-cosine filter of roll-off 1, the symbol rate is 2 k symbols/s. Thus, to send a packet of length $L = 120$ bits—which requires 120 BPSK modulated symbols—the time needed

is 0.06 s. Notice that with this choice of bandwidth and symbol rate, we can assume an ISI-free reception, based on the discussion in Section 3.2.1.2.

Next, to compute the probability of packet error in the uncoded case (p_u), we first note that using (3.1), the bit-error probability p_b at E_b/N_0 of 5 dB is 5.95×10^{-3} . Since the noise process is independent from symbol-to-symbol, the probability of packet error p_u is simply given by $p_u = 1 - (1 - p_b)^L = 0.512$. Substituting $T_p = 0.06$ s and $p_u = 0.512$ into (3.4), we get the average service time in the uncoded system to be 0.123 s. Next, consider the following three coded systems:

System A A hypothetical rate-1/3 code with probability of packet error, $p_c = 0$.

System B A (7, 4) Hamming Code.

System C A (63, 45) Bose-Chaudhuri-Hocquenghem (BCH) Code. This code can correct all error patterns of weight 3 or less[5].

We plot the three systems in figure 3.3 on an $\langle R_c, p_c \rangle$ plane. Note that the rate for **System C** is marked as $120/189 = 0.635$ since the packet is padded to 135 bits prior to encoding. This is because we are employing a block code—BCH code in this case— and we need the number of bits in the message to be such that an integer number of code blocks are present in the transmitted packet. The packet error probabilities for **System B** and **System C** are obtained by Monte-Carlo simulation.

In figure 3.3, the codes which fall in the shaded region are STI codes with respect to the uncoded system. The boundary of the shaded region is obtained from (3.8). Thus, from Theorem 3.1, a code is an STI code if and only if it falls

[5]A brief description of the Hamming codes and BCH codes is presented in Section 3.2.1.3.

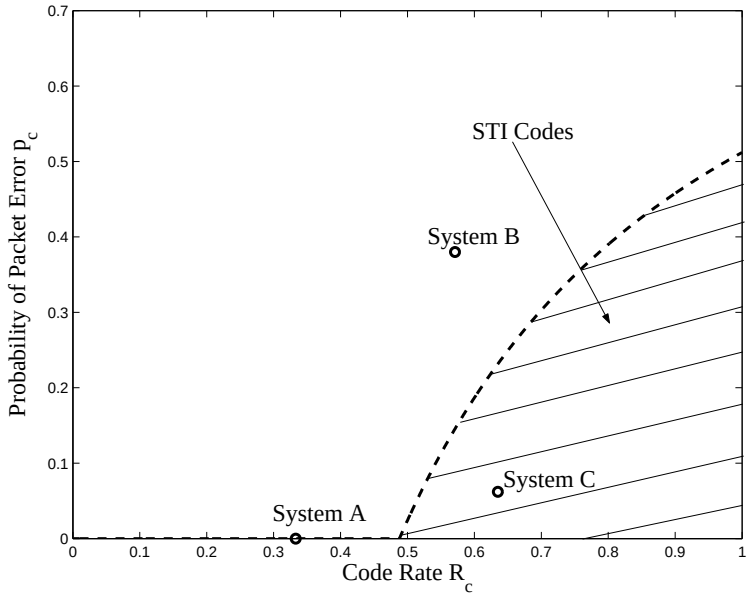

Figure 3.3: Pictorial Representation of Theorem 3.1. A code is an STI code if and only if it falls in the shaded region.

Table 3.1: $\bar{T}_s'$ values for the three coded systems. The average service time for the uncoded system is $\bar{T}_s = 0.123s$.

System	Coding	Measured T_s'	$\nu(0, \alpha, \beta, 0.512)$
A	Hypothetical rate-1/3	0.1800s	0.683
B	(7,4) Hamming	0.1769s	0.695
C	(63,45) BCH	0.1007s	1.221

in the shaded region. Hence, we expect only the code in **System C** to be an STI code.

We verify this by performing discrete event simulations for the ARQ system. The $\nu(\alpha, \beta, 0.512)$ values recorded for the three coded systems are tabulated in table 3.1. Note that each of the three coded systems is characterized by a unique α and β. As expected, $\nu(\alpha, \beta, 0.512) > 1$ only in **System C**. Equivalently, the average service time is lower compared to the uncoded system only in **system C**, as seen in table 3.1.

3.5.2 Digital Modulation

Next, consider comparing a digital modulation that sends ml bits per modulated symbol, where $m > 1$, without any error-correction coding and without changing the symbol duration, to a modulation which sends l bits per modulated symbol (for example $l = 1$ for BPSK). In this case, the packet duration becomes approximately $1/m$ times the original packet duration.

Theorem 3.2. *The average service time is lower in a system with ml bits per symbol modulation if and only if*

$$p_{ml} < \frac{p_l + m - 1}{m} \tag{3.10}$$

where p_l and p_{ml} represent the packet error probabilities with the l bits per symbol modulation and the ml bits per symbol modulation respectively.

Proof. This result follows from Lemma 3.1 by replacing α by $1/m$, p by p_l and β by p_{ml}/p_l. □

A numerical illustration of this result will be shown in Section 3.7.2.1, where we shall also illustrate the result relating the constellation size to the average delay.

3.6 Average Delay

So far we have only looked at the average service time. In this section, we will look at how the average service time relates to the average delay.

Definition 3.17. *We define* Average Delay *as the sum of the average service time and the average queueing delay.*

We assume that the arrival of packets from the higher layer is a Poisson process [23, page 164] with an average rate of λ packets/sec. Recall that with Poisson arrivals, the buffered ARQ system under consideration is an M/G/1 queue (see Section 3.2.3). The average queueing delay, which was defined in Section 3.2.3, can be obtained using the P-K formula (see (3.2)).

As we highlighted in Observation 3.5, in M/G/1 queues, a lower average service time is not sufficient to also ensure a lower average delay. In light of Observation 3.5, the following lemma is useful:

Lemma 3.2. *Let Q_1 and Q_2 be two M/G/1 queues with average service times $\bar{X}$ and $\bar{X}'$ and the second moments of service times Y and Y' respectively. Let W and W' denote the average delay (sum of service time and the queueing delay) for Q_1 and Q_2 respectively. Then, if $\bar{X} > \bar{X}'$ and $Y > Y'$, then $W > W' \forall \lambda$ such that $0 \leq \lambda < 1/\bar{X}$.*

Proof. The proof of this Lemma follows from the P-K formula. We have,

$$W = \bar{X} + \frac{\lambda Y}{2(1-\lambda\bar{X})} \qquad \text{and} \qquad W' = \bar{X}' + \frac{\lambda Y'}{2(1-\lambda\bar{X}')}. \tag{3.11}$$

If $\bar{X} > \bar{X}'$ and $Y > Y'$, then from (3.11), $W > W'$ for all λ, such that $0 < \lambda < 1/\bar{X}$. □

3.6.1 Average Delay in the Modified System

The ARQ systems under consideration have geometric service distribution, described in (3.3). Let S and S' denote the second moments of the service time in the original (with packet duration T_p and probability of error p) and the modified (with packet duration T_p' and probability of error p') ARQ systems

respectively. That is, $S = E(T_s^2)$ and $S' = E(T_s'^2)$. Since the distribution of the service time is geometric in the case of Stop-and-Wait ARQ, we have [88, page 133]

$$S = (1+p)(\bar{T}_s)^2 \qquad \text{and} \qquad S' = (1+p')(\bar{T}_s')^2 \tag{3.12}$$

where $\bar{T}_s$ and $\bar{T}_s'$ are given in (3.4) and (3.6) respectively.

Let $\bar{T}$ and $\bar{T}'$ represent the average delay in the original and the modified systems respectively. Using (3.11), $\bar{T}$ works out to

$$\bar{T} = \frac{T_p}{1-p} + \frac{\lambda(1+p)T_p^2}{2(1-p)(1-p-\lambda T_p)}. \tag{3.13}$$

The expressions for $\bar{T}'$ can be worked out by replacing p by βp and T_p by αT_p in (3.13).

Definition 3.18. *Define $\eta(\alpha, \beta, p)$ as the ratio of the average packet delay in the original and the modified system for a given arrival rate λ:*

$$\eta(\alpha, \beta, p) = \frac{\bar{T}}{\bar{T}'} \qquad \textit{for a given } \lambda. \tag{3.14}$$

Observation 3.7. *For a given λ, $\eta(\alpha, \beta, p) > 1$ indicates a lower average delay in the modified system characterized by α and β, when the probability of error in the uncoded system is p.*

In Lemmas 3.4 and 3.5, we shall study the behavior of $\eta(\alpha, \beta, p)$ as a function of α, β and $\nu(\alpha, \beta, p)$. In Lemma 3.4, we study the case of $\alpha > 1$, that is, when the modified system has a higher packet duration than the original system (for example with FEC). Next, in Lemma 3.5, we look at the case of $\alpha < 1$, that is, when the packet duration in the modified system is smaller than the original

system (for example due to an increase in the constellation size). We first present a precursor result in the following Lemma.

Lemma 3.3. *Assuming both the original and the modified systems are stable, the following statements are true:*

1. *If* $\nu(\alpha, \beta, p) > 1$ *and* $S > S'$, $\eta(\alpha, \beta, p) > 1$.

2. *If* $\nu(\alpha, \beta, p) < 1$ *and* $S < S'$, $\eta(\alpha, \beta, p) < 1$.

Proof. Apply Lemma 3.2. □

Lemma 3.4. *The following statements are true for all values of* λ *for which both the original and the modified systems are stable:*

1. *If* $\alpha > 1$ *and* $\nu(\alpha, \beta, p) > 1$, *then* $\eta(\alpha, \beta, p) > 1$.

2. *If* $\alpha > 1$ *and* $\beta > 1$, *then* $\eta(\alpha, \beta, p) < 1$.

3. *If* $\alpha > \frac{\sqrt{1+p}}{1-p}$, *then* $\eta(\alpha, \beta, p) < 1$.

Proof. Consider the statements one by one:

1. From Lemma 3.1, $\nu(\alpha, \beta, p) > 1 \Rightarrow \beta < \frac{1-\alpha(1-p)}{p}$. Furthermore, if $\alpha > 1$, then $\frac{1-\alpha(1-p)}{p} < 1$, which means that $\beta < 1$. Hence,

$$\frac{S}{S'} = \frac{1+p}{1+\beta p}\nu(\alpha, \beta, p)^2 > 1. \tag{3.15}$$

   Since $\frac{S}{S'} > 1$ and $\nu(\alpha, \beta, p) > 1$, from Lemma 3.3, we conclude that $\eta(\alpha, \beta, p) > 1$.

2. $\alpha > 1 \Rightarrow \frac{1-\alpha(1-p)}{p} < 1$. Hence, $\beta > 1 \Rightarrow \beta > \frac{1-\alpha(1-p)}{p} \Rightarrow \nu(\alpha, \beta, p) < 1$, where the last deduction comes from Lemma 3.1. Since $\beta > 1$ and $\nu(\alpha, \beta, p) < 1$, we have

$$\frac{S}{S'} = \frac{1+p}{1+\beta p}\nu(\alpha,\beta,p)^2 < 1. \tag{3.16}$$

Since $\nu < 1$ and $\frac{S}{S'} < 1$, from Lemma 3.3, we conclude that $\eta(\alpha,\beta,p) < 1$.

3. $\alpha > \frac{\sqrt{1+p}}{1-p} \Rightarrow \alpha > \frac{1}{1-p} \Rightarrow \beta > \frac{1-\alpha(1-p)}{p} \Rightarrow \nu(\alpha,\beta,p) < 1$ from Lemma 3.1.

   Next consider, if $\alpha > \frac{\sqrt{1+p}}{1-p}$,

$$\begin{aligned} \frac{S}{S'} &= \frac{1+p}{1+\beta p}\nu(\alpha,\beta,p)^2 = \frac{1+p}{1+\beta p}\frac{(1-\beta p)^2}{\alpha^2(1-p)^2} \\ &< \frac{(1-\beta p)^2}{1+\beta p} < 1 \end{aligned} \tag{3.17}$$

   because $\beta \geq 0$.

   Since $\nu(\alpha,\beta,p) < 1$ and $\frac{S}{S'} < 1$, from Lemma 3.3, we conclude $\eta(\alpha,\beta,p) < 1$.

□

Lemma 3.5. *The following statements are true for all values of λ for which both the original and the modified systems are stable.*

1. *If $\alpha < 1$ and $\nu(\alpha,\beta,p) < 1$, then $\eta(\alpha,\beta,p) < 1$.*

2. *If $\alpha < 1$ and $\beta < 1$, then $\eta(\alpha,\beta,p) > 1$.*

Proof. Consider the statements one by one:

1. $\nu(\alpha,\beta,p) < 1 \Rightarrow \beta > \frac{1-\alpha(1-p)}{p}$. $\alpha < 1 \Rightarrow \frac{1-\alpha(1-p)}{p} > 1$. Hence $\beta > 1$. Hence,

$$\frac{S}{S'} = \frac{1+p}{1+\beta p}\nu(\alpha,\beta,p)^2 < 1. \tag{3.18}$$

Hence, from Lemma 3.3, $\eta(\alpha, \beta, p) < 1$.

2. $\alpha < 1 \Rightarrow \frac{1-\alpha(1-p)}{p} > 1$. Hence $\beta < 1 \Rightarrow \beta < \frac{1-\alpha(1-p)}{p} \Rightarrow \nu(\alpha, \beta, p) > 1$. Hence,

$$\frac{S}{S'} = \frac{1+p}{1+\beta p}\nu(\alpha, \beta, p)^2 > 1. \tag{3.19}$$

Hence, from Lemma 3.3, $\eta(\alpha, \beta, p) > 1$.

□

3.7 Relating to Physical Layer Processing

3.7.1 Forward Error Correction

Theorem 3.3. *The following statements are true for Poisson arrivals and for all arrival rates under which both the uncoded and the coded system are stable.*

1. *STI codes reduce the average delay as compared to uncoded transmission.*
2. *Non-STI codes of rate $R_c < \frac{1-p_u}{\sqrt{1+p_u}}$ increase the average delay with respect to uncoded transmissions, regardless of p_c, the probability of error with coding. p_u, as before, is the probability of packet error in the uncoded system.*

Proof. Note that with FEC, $\alpha > 1$ if the constellation size and the symbol rate in symbols/sec are kept unchanged. Now consider the statements one by one:

1. Apply Statement 1 from Lemma 3.4.
2. First of all, notice that codes with rate $R_c < \frac{1-p_u}{\sqrt{1+p_u}}$ are not STI codes as they do not satisfy Corollary 3.1.1. Next, apply Statement 3 from Lemma 3.4 and use $R_c = \frac{1}{\alpha}$ and replace p by p_u.

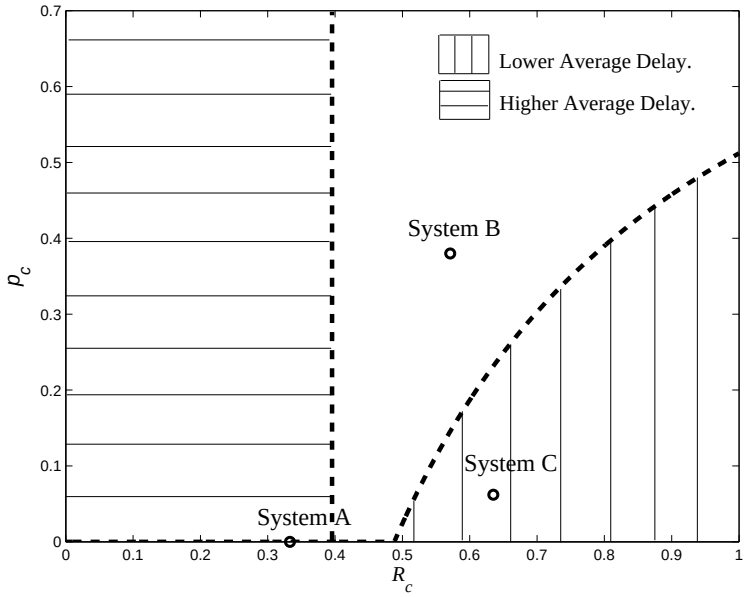

Figure 3.4: R_c, p_c regions where the coded systems exhibit higher and lower average delay compared to uncoded systems. No immediate conclusion can be made about the codes falling in the unshaded region.

□

3.7.1.1 Numerical example

Theorem 3.3 presents the impact of coding on the average delay. In this section, we illustrate the result in Theorem 3.3 by presenting a discrete-event simulation example. Consider once again, the situation considered in the numerical example in Section 3.5.1.1. Recall that in Section 3.5.1.1, we compared the average service time in three coded systems, named as **System A**, **System B** and **System C**, to the average service time in the uncoded system. Out of the three systems, only **System C** employed an STI code.

In this section, we shall study the average delay performances of the three coded systems with respect to the uncoded system. Before doing so, we first plot in figure 3.4, the regions in the $\langle R_c, p_c \rangle$ plane where codes that reduce or increase the average delay with respect to the uncoded system fall. The

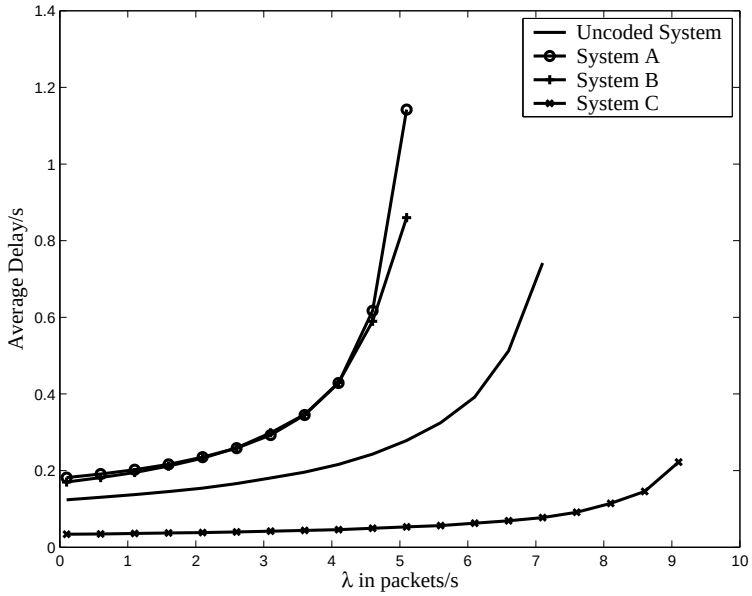

Figure 3.5: The average delay performance of the different systems. As expected, **System A** exhibits a higher average delay as compared to the uncoded system and **System C** exhibits a lower average delay. **System B** also exhibits a higher delay compared to the uncoded system, though this could not be predicted from the analysis.

boundaries of the shaded regions in figure 3.4 have been obtained using firstly, the condition for a code to be an STI code in (3.8) and secondly, statement 2 in Theorem 3.3. Based on figure 3.4, we expect **System C** to exhibit a lower average delay whereas **System A** to exhibit a higher average delay compared to the uncoded system for all arrival rates where the systems are stable. As for **System B**, no immediate deduction can be made from Theorem 3.3.

The results of a discrete-event simulation on the different systems are presented in figure 3.5. As expected, **System A** exhibits a higher average delay for all arrival rates and **System C** exhibits a lower average delay for all arrival rates as compared to the uncoded system. We also find that **system B** also exhibits a higher average delay as compared to the uncoded systems, though this could not be predicted from the analysis alone.

3.7.2 Digital Modulation

Next, let us compare the average delay performance in a modulation that sends ml bits per symbol, where $m > 1$, to a modulation which sends l bits per symbol (for example $l = 1$ for BPSK), with the symbol rate kept unchanged and without any error-correction coding. As before, let p_l and p_{ml} be the packet error probabilities with the l bits per symbol modulation and the ml bits per symbol modulation respectively.

Theorem 3.4. *The following statements are true for all arrival rates under which both the systems are stable:*

1. *If*

$$p_{ml} > \frac{p_l + m - 1}{m} \tag{3.20}$$

   then the system with ml bits per symbol modulation exhibits higher average delay.

2. *If $p_{ml} < p_l$, then the average delay in the system with ml bits per symbol modulation is smaller.*

Proof. Since $m > 1$, the packet duration in the modified system is smaller than the original system ($\alpha < 1$). Now consider the statements one by one.

1. If (3.20) holds, then from Theorem 3.2, $\nu(\alpha, \beta, p) < 1$. Next, apply Statement 1 of Lemma 3.5.

2. Apply Statement 2 of Lemma 3.5 and substitute $\beta = \frac{p_{ml}}{p_l}$.

□

3.7.2.1 Numerical example

Consider a communication link with bandwidth of 15 kHz. Let us say that the data packets have a length $L = 100$ bits and BPSK modulation with a raised-cosine filter of roll-off zero is employed. ISI-free transmission is desired and accordingly, the symbol rate is set at 15 k symbols/sec. Since BPSK modulation sends one modulated symbol for every data bit, assuming that no error-correcting code is used, the packet duration T_p is $T_p = 100 \times (1/15000)$ sec $= 0.0067$ sec. Let us say that the E_b/N_0 at the receiver is 6 dB. Using (3.1), we find the bit error probability to be $p_b = 2.39 \times 10^{-3}$, and accordingly, the packet error probability p_l to be $p_l = 1 - (1 - p_b)^L = 0.2127$.

Let us now see the impact of changing the modulation to Quadrature Phase Shift Keying (QPSK) without introducing any error-correcting code and without changing the symbol duration and the transmitted power. Firstly, note that unlike BPSK, QPSK sends two bits per symbol. Hence, if the symbol rate is unchanged, it takes half as much time to send the packet, leading to a packet duration of 0.0033 sec. Next, note that the transmitted power is kept unchanged and symbol duration are unchanged. This means that the energy *per bit* is halved as compared to the BPSK case. Since QPSK is essentially two orthogonal BPSK constellations in tandem [12, page 268], the bit error probability in the QPSK case is simply the bit error probability in BPSK system at E_b/N_0 of 3 dB. The three dB reduction in the E_b/N_0 is because of the constant power and symbol duration, which, as discussed above, halves the energy per bit. Making these calculations, we find p_b in the QPSK system as 2.29×10^{-2} from (3.1) and hence the probability of packet error $p_{2l} = 1 - (1 - p_b)^L = 0.9011$. This results in $\beta = 0.9011/0.2127 = 4.236$. Note that $\alpha = 0.5$. Substituting these values in Theorem 3.2, we notice that the average service time in the QPSK system is

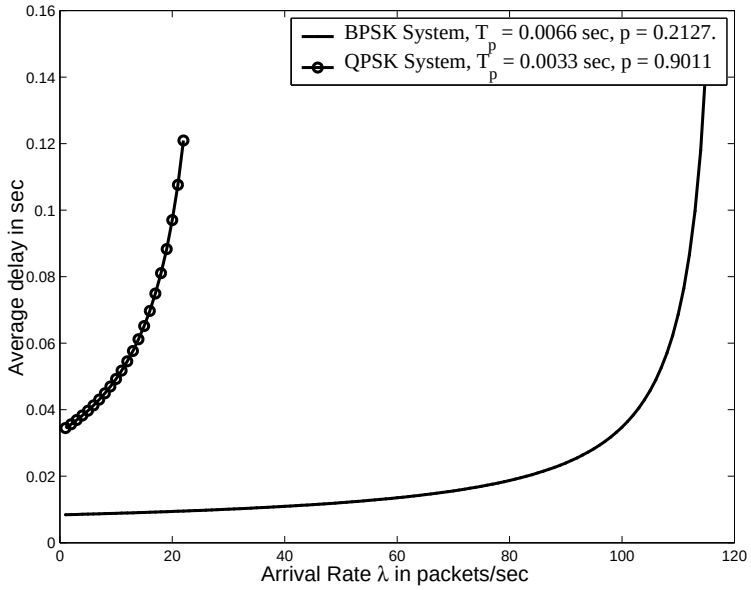

Figure 3.6: The simulated delay performance against λ. As expected, the QPSK system exhibits a higher average service time as well as a higher average delay compared to the BPSK system.

expected to be larger than in the BPSK system. Next, from Theorem 3.4, we further expect $\eta(0.5, 4.236, 0.2127) < 1$ for all λ under which both systems are stable. That is, for all arrival rates, we expect the average delay to be larger in the QPSK system.

We verify these by discrete event simulation, whose results are shown in figure 3.6. The higher average service time in the QPSK system is seen at low arrival rates. The higher average delay in the QPSK system is also clear from figure 3.6.

3.8 Conclusions

We developed quantitative conditions on the physical layer processing with the link-layer average service time as our metric of interest. We saw two specific instances of these guidelines, namely, for the case of forward error correction

and for the constellation size. We moved on to look at the impact of physical layer processing on the average delay. All the design effort presented in this chapter can be seen as an example of a form of cross-layer design, whereby the PHY layer is designed conditioned on Stop-and-Wait ARQ at the link layer. The benefit in this approach is that it provides a mechanism to ensure that the functionalities at two different layers do not act at cross-functions to one another. The cost is the close coupling between the layers. For example, if the ARQ method in the system model were to be changed, the guidelines that we have developed for physical layer processing may no longer hold.

Chapter 4

Queueing Meets Coding

4.1 Introduction

We developed guidelines for physical layer processing in a Stop-and-Wait ARQ system in Chapter 3 and obtained a necessary and sufficient condition for a rate R_c code to be an STI code. In this chapter, we interpret the necessary and sufficient condition for being an STI code in Theorem 3.1 in light of the Varshamov-Gilbert bound and the sphere-packing bound, which are two well known results in coding theory. By doing so, we study the existence of binary STI linear block codes. We also consider the case of large packet lengths and determine sufficient conditions for the existence of codes that reduce the average service time with respect to uncoded transmission using the asymptotic form of the VG bound and the channel capacity theorem.

Notice that the analysis in Chapter 3 is based primarily on a queueing theoretic view whereby the packets from the higher layers are seen as receiving service from the physical layer and the link layer. This chapter extends the queueing theoretic view by making use of ideas from coding theory. Combining coding and queueing enables us to study the existence of linear block codes

that are STI with respect to a given uncoded system. The queueing analysis in Chapter 3 alone does not provide this insight. At a more theoretical level, we bring results from a queueing analysis and coding theory together. Such an approach leads to a better appreciation of both the disciplines. The methodology can be used to relate FEC to other higher layer metrics too.

Several recent works (for example [77], [78], [80]) have dealt with the impact of FEC on higher layer metrics. However, to the best of our knowledge, the use of results from coding theory to augment the queueing theoretic analysis—as done here—is unique.

In the rest of this chapter, in Section 4.2, we review some known coding theoretic results that we will be using in the later analysis. Next, we do a quick recap of the necessary and sufficient condition for a rate R_c code to be an STI code in Section 4.3. Next, we take up the issue of existence of STI codes in Section 4.4. It is in Section 4.4 that we merge the queueing inspired necessary and sufficient condition with the VG bound and the sphere-packing bound. We move on to look at the asymptotic case of large packet length in Section 4.5 and finally draw our conclusions in Section 4.6.

4.2 Known results for linear block codes

All texts on Coding theory (see for example [76], [72]) cover the topic of binary linear block codes in good detail. Here, we just look at some specific results that are relevant to the current discussion. For the results and terminologies presented in this section, we do not provide a detailed and formal proof, since they can be readily obtained by following the references. Rather, we just provide some intuitive arguments that motivate the formal proof and, where applicable, some additional insight into the results. We only deal with binary codes.

In the rest of this section, we first introduce the fundamental concepts in Section 4.2.1. Next, in Section 4.2.2, we look at the results relating to the error correction capabilities of linear block codes. We move on to look at the known bounds on code size with given parameters in Section 4.2.3 and illustrate these bounds with a numerical example in Section 4.2.3.1. We then look at the asymptotic forms of these bounds in Section 4.2.4. Next, we look at a result known as the sphere-packing bound in Section 4.2.5. The sphere-packing bound makes use of the Hamming bound and provides a lower bound on the probability of codeword error for an (n, k) linear block code. Readers familiar with the aforementioned concepts may want to skip the rest of this section and move straight to Section 4.3.

4.2.1 Fundamental Concepts

4.2.1.1 Terminology

A block code consists of a set of fixed length code words. Every message input, which is also of a fixed length, is mapped to a code word [72, page 69].

Definition 4.1. *A block code of length n and 2^k codewords is called a* linear (n, k) *code if and only if its 2^k code words form a k-dimensional subspace of the vector space of all the n-tuples over the (Galois) field GF(2)[76, page 52].*

Definition 4.2. *For an (n, k) code, n is called the* block length *and k is called the* message length.

Definition 4.3. *The* rate *of an (n, k) code is given by the number k/n.*

Definition 4.4. *The* Hamming Weight *of a code word is the number of non-zero components in a code word. We denote the Hamming weight of a codeword $\mathbf{v}$ by $w(\mathbf{v})$.*

Definition 4.5. *The* Hamming Distance *between two codewords* $\mathbf{v}$ *and* $\mathbf{w}$ *is defined as the number of places where they differ. The Hamming Distance is denoted as* $d(\mathbf{v}, \mathbf{w})$.

Definition 4.6. *The* minimum distance *of a block code* C *is defined as*

$$d_{min} = \min\{d(\mathbf{v}, \mathbf{w}) : \mathbf{v}, \mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\} \tag{4.1}$$

4.2.1.2 The Generator and the Parity-Check matrices

Every linear block code is characterized by two matrices: the generator matrix $\mathbf{G}$ and the parity check matrix $\mathbf{H}$. For an (n, k) linear block code, $\mathbf{G}$ is a $k \times n$ matrix. The $1 \times n$ codeword $\mathbf{v}$ corresponding to the $1 \times k$ message $\mathbf{u}$ is given by $\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$ [76, page 53]. The k rows of the $\mathbf{G}$ matrix are linearly independent.

The parity check matrix $\mathbf{H}$ is an $(n - k) \times n$ matrix. A vector in the row space of the $\mathbf{H}$ matrix is orthogonal to any vector that is in the row space of the $\mathbf{G}$ matrix. Conversely, a vector that is orthogonal to the vectors in the row space of the $\mathbf{G}$ matrix is in the row space of the $\mathbf{H}$ matrix. Hence, an n-tuple $\mathbf{v}$ is a codeword generated by a $\mathbf{G}$ if and only if $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$, where $\mathbf{0}$ is the zero vector [76, page 55].

Result 4.1. *The minimum distance of a linear block code is equal to the minimum weight of its nonzero code words.*

Proof. The Hamming distance between two code words is the weight of their sum over GF(2) [76, page 63]. Also, the addition of two codewords results in another codeword. Thus, the smallest Hamming distance between any two codewords is the weight of that code word which has the minimum weight amongst the non-zero code words in the code—in other words, the minimum weight of its non-zero codewords. See [76, page 63] for a formal proof. □

Next, we look at an important property of the parity check matrix. This property will be used in the derivation of the Varshamov-Gilbert bound in Section 4.2.3.

Result 4.2. *Let C be an (n, k) linear block code with parity check matrix $\mathbf{H}$. For each code vector of Hamming weight w, there exist w columns of $\mathbf{H}$ such that the vector sum of these w columns is equal to the zero vector. Conversely, if there exist w columns of $\mathbf{H}$ whose sum is the zero vector, there exists a code vector of Hamming weight w in C.*

Proof. The proof follows from the property that an n-tuple $\mathbf{v}$ is a code word if and only if $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$. Details can be found at [76, page 64]. □

The following two results follow directly from Result 4.2:

Result 4.3. *If no d - 1 or fewer columns of $\mathbf{H}$ add to $\mathbf{0}$, the code has minimum weight at least d.*

Result 4.4. *The minimum distance of a code is equal to the smallest number of columns of $\mathbf{H}$ that sum to $\mathbf{0}$.*

4.2.1.3 Hamming sphere

We now introduce the notion of a Hamming Sphere as that will motivate a geometric interpretation of error correction.

With block-length of n, there are 2^n possible n-tuples. Out of these, 2^k are chosen as codewords. From a geometric viewpoint, the 2^n n-tuples are points in an n-dimensional vector space. The points that are selected as codewords form a k-dimensional subspace. The vector space formed by the binary n-tuples is similar to the familiar Euclidean Space, where the notion of Euclidean distance is replaced by the notion of Hamming Distance.

Similar to the concept of a sphere in a Euclidean space, we can define a *Hamming sphere*. A Hamming Sphere of radius t is a collection of points that lie within a Hamming distance of t from the n-tuple at the center.

Definition 4.7. *The volume of a Hamming Sphere of radius t, $V(n,t)$ is*

$$V(n,t) = \sum_{j=0}^{t} \binom{n}{t}. \tag{4.2}$$

Observation 4.1. *The volume of a Hamming sphere is simply the number of points in the sphere.*

4.2.2 Error Correction Capability

Result 4.5. *A linear block code C whose minimum distance d_{min} satisfies*

$$d_{min} \geq 2t + 1 \tag{4.3}$$

can correct all error patterns of t or fewer errors.

Proof. The intuition behind this result is that if d_{min} satisfies $d_{min} \geq 2t+1$ and an error pattern of weight t or less occurs, the perturbed codeword will still be closer to the sent codeword as compared to all the other codewords. Hence, the perturbed codeword can be correctly decoded by choosing the codeword that is nearest—in terms of the Hamming distance—to the received codeword. A formal proof of this result, also highlighting why it makes sense to decode the perturbed codeword to the nearest codeword, is available at [76, page 66]. □

Result 4.6. *A block code C whose minimum distance satisfies*

$$d_{min} \leq 2t + 2 \tag{4.4}$$

can not correct all error patterns of weight l, where $l > t$.

Proof. Again, the idea here is that if $l > t$, then there is at least one error pattern of weight l that will perturb the transmitted codeword so much that the Hamming distance between the received codeword and the transmitted codeword will be larger than the Hamming distance between the received codeword and some other code word. Hence, choosing the nearest codeword from the perturbed codeword will result in a decoding error. A formal proof is available at [76, page 67]. □

Thus, a code that has $2t + 1 \leq d_{min} \leq 2t + 2$ can guarantee correcting all error patterns of weight t or less. It may also be able to correct some error patterns of weight $t + 1$. However, the code cannot correct *all* error patterns of weight $t + 1$.

4.2.3 Bounds on Code Size

There are three fundamental parameters in a linear block code. They are, the block length n, the message length k, and the error-correcting capability t. From these three parameters, we can derive the code rate R_c simply as $R_c = k/n$ and the minimum distance d_{min}.

Bounds on code size are usually expressed as inequalities involving n, k and t. The coding bounds can be used to answer one of the following three related questions:

1. **Bound on t:** Given the block length n and message length k, what are the lower and upper bounds on the error-correcting capability t?

2. **Bound on k:** Given the block length n and an error-correcting capability t, what are the lower and upper bounds on the message length k?

3. **Bound on n:** Given the message length k and an error-correcting capability t, what are the lower and upper bounds on the block length n?

There are several known bounds on the three parameters mentioned here (See for reference [86, page 76]). In our discussion, we will make use of what is known as the Varshamov-Gilbert (VG) bound [86, page 86]. The VG bound provides a sufficient condition for the existence of codes with certain parameters. We start though with the Hamming bound ([76, page 83]), as that serves to put the VG bound in perspective. The Hamming bound will also be used in the development of the sphere-packing bound in Section 4.2.5.

Result 4.7. ***Hamming Bound*** *All binary (n, k) linear block codes with t-error correcting capability satisfy the following inequality:*

$$n - k \geq log_2 \left[1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t} \right] \tag{4.5}$$

This bound is known as the Hamming Bound. *Hamming Codes [76, page 79] achieve the Hamming Bound with equality.*

Proof. A proof of Hamming bound using the concept of coset leaders is available at [76, page 83]. Here we look at another proof that provides insight into the geometric significance of this result.

Let C be an (n, k) linear code with t-error correcting capability. Around each code point, a Hamming sphere of radius t contains all the received code words that are obtained as a result of perturbation of the transmitted code word by an error pattern of weight t or less. Since C is a t-error correcting code, the Hamming spheres centered at the code points do not overlap. From (4.2), the volume of each of the Hamming spheres is $1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}$. Thus we have

$$2^k \leq \frac{2^n}{\left[1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}\right]} \tag{4.6}$$

simply because the number of spheres is bounded by the total volume divided by the volume of an individual sphere. The result follows. □

The Hamming bound provides an upper bound on t, given n and k. Alternatively, it also provides an upper bound on k, given an n and t. It must be added that tighter upper bounds on t, given n and k are also known [86, page 76].

The Hamming bound is a necessary condition that all linear block codes satisfy. Next, we look at two results that provide sufficient conditions for the existence of (n, k) block codes with specified t.

Result 4.8. ***Gilbert Bound*** *There exists an (n, k) block code with t-error correcting capability if*

$$n - k \geq log_2 \left[1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{2t}\right] \tag{4.7}$$

This bound is known as the Gilbert Bound *([72, page 76], [89], [90]).*

Proof. If we have

$$2^k \leq \frac{2^n}{\left[1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{2t}\right]} \tag{4.8}$$

then we can choose 2^k points out of the 2^n to be the codewords such that $1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{2t}$ points surrounding each chosen point is left out. The 2^k chosen points will be the codewords of an (n, k) block code with t-error correcting capability, as the codewords will have a minimum distance of at least $2t + 1$. The result follows by simple algebraic manipulation on (4.8). □

The Varshamov-Gilbert bound [90] improves on the Gilbert Bound in the finite n case. The asymptotic version of both Gilbert bound and the VG bound are the same.

Result 4.9. ***Varshamov-Gilbert Bound*** *It is possible to construct an (n, k) code with minimum distance at least d for which the following inequality holds:*

$$\binom{n-1}{1} + \binom{n-1}{2} + \ldots + \binom{n-1}{d-2} < 2^{n-k} - 1. \tag{4.9}$$

Proof. If the inequality in (4.9) is satisfied, then it can be shown ([90], [86, page 86]) that an $(n-k) \times n$ matrix with no d - 1 or fewer columns adding to $\mathbf{0}$ can be formed. The (n, k) code with minimum distance at least d is then simply the null space of the $(n-k) \times n$ matrix.

□

The mathematical forms of the expressions in (4.5), (4.7) and (4.9) are similar. However, there is a crucial difference between the interpretation of the Hamming bound and the interpretation of the Gilbert and the VG bounds. For given n and k, let t_{HB} and t_{GB} be the largest values of t that satisfy (4.5) and (4.7) respectively. We expect $t_{GB} \leq t_{HB}$. From the Hamming bound, no (n, k) code can correct more than t_{HB} errors in a block. From the Gilbert bound, there exists at least one (n, k) code that can correct t_{GB} errors per block. The Gilbert bound does not rule out the possibility that (n, k) codes with t-error correcting capability exist, where $t_{GB} \leq t \leq t_{HB}$. All it says is that error-correcting capability of at least t_{GB} is achievable. A similar interpretation holds for the VG bound.

The Gilbert bound and the Varshamov-Gilbert bound are often referred to as "lower-bounds" on the minimum distance (or the error-correcting capability) of an (n, k) code (See for example [76, page 84]). This is somewhat misleading. The

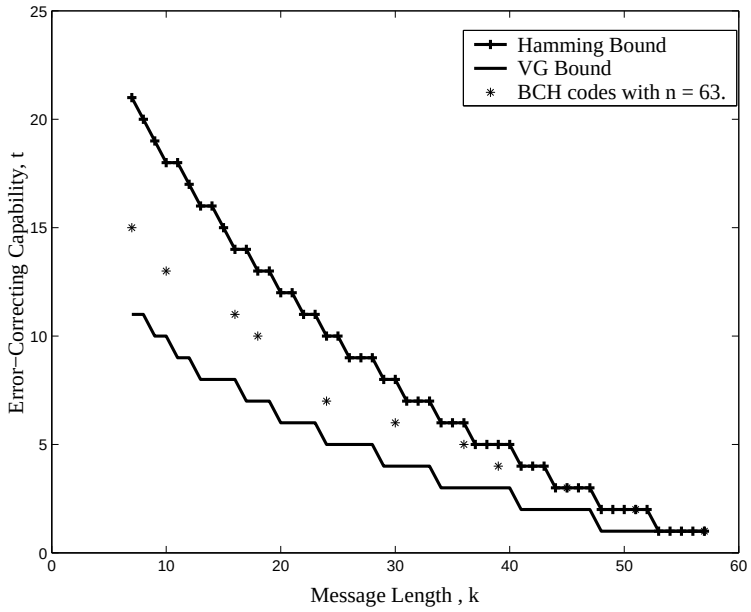

Figure 4.1: Given $n = 63$, the various bounds on t as k takes on different values.

Gilbert bound and the Varshamov-Gilbert bound only tell what is achievable in terms of error-correcting capability of an (n, k) code, and not what all (n, k) codes actually achieve, which would be the true lower bound. Interestingly, in his original paper, Varshamov referred to what is now known as the VG bound as a sufficient condition for the existence of a code with specified parameters, and not as a lower bound on minimum distance of an (n, k) code [90].

4.2.3.1 Numerical Illustration

Let's say the block length $n = 63$. We let message length k range from 7 to 57 and investigate what the Hamming bound and the VG bound predict for the error-correcting capability t. The result is shown in figure 4.1. For comparison, we also show the error-correction capability of Bose-Choudhury-Hocquenghem (BCH) codes [12, page 438].

We make two observations from Figure 4.1. Firstly, as expected, all BCH code points fall below the Hamming Bound. Secondly BCH codes exhibit much

better error correcting capabilities than what is predicted by the Varshamov-Gilbert (VG) bound, particularly for small k. This highlights the looseness of the VG bound for small block lengths.

A similar numerical example comparing the Hamming bound and the Gilbert bound is presented at [72, page 77]. In that example, the redundancy (defined as $r = n - k$) needed to achieve single-error correcting capability ($t = 1$) is plotted as a function of the block length n. As expected, the Hamming bound in (4.5) provides the minimum redundancy that *must* be added. The Gilbert bound in (4.7), on the other hand, provides the minimum redundancy, which if added, guarantees the existence of a code with $t = 1$.

4.2.4 Asymptotic forms of bounds on Code Size

We now discuss the asymptotic form of the coding bounds discussed earlier. The asymptotic forms of the bounds apply as the block length $n \to \infty$. We first make the following definition.

Definition 4.8. *Define the* Relative Minimum Distance *of an* (n, k) *code as the ratio of the minimum distance* d_{min} *and the block length* n *[72, page 79]. The relative minimum distance is denoted as* δ*. Hence,*

$$\delta = \frac{d_{min}}{n}. \tag{4.10}$$

Result 4.10. *When* $0 \leq \delta \leq 1/2$*, we have*

$$\lim_{n\to\infty} \left[\frac{1}{n} \log \sum_{j=0}^{m} \binom{n}{j} \right] = H(\delta) \tag{4.11}$$

where m *is a positive integer defined as* $m = \lfloor \delta n \rfloor$*,* $\lfloor \bullet \rfloor$ *being the largest integer smaller than the argument and* $H(\delta)$ *is the entropy function defined as*

$$H(\delta) = -\delta \log \delta - (1-\delta)\log(1-\delta). \tag{4.12}$$

All the logarithms in (4.11) and (4.12) are to the base-2.

Proof. A proof based on Sterling's formula [86, page 466] is presented in [72, page 80]. A more intuitive proof is presented in [75, page 284]. □

Result 4.11. ***Hamming Bound (Asymptotic)*** *For large block lengths, all linear block codes with rate R and relative minimum distance δ satisfy*

$$R \leq 1 - H(\delta/2) \tag{4.13}$$

where $H(\bullet)$ is the entropy function defined in (4.12).

Proof. The result can be proven by considering (4.5) as $n \to \infty$ and then by applying (4.11). □

Result 4.12. ***Varshamov-Gilbert Bound (Asymptotic)*** *For large block lengths, there exist rate R codes that satisfy*

$$R \leq 1 - H(\delta) \tag{4.14}$$

where $H(\bullet)$ is the entropy function defined in (4.12).

Proof. The result can be proven by considering (4.9) as $n \to \infty$ and then by applying (4.11). □

The interpretation of these bounds in the asymptotic case is similar to the case of finite n. The Hamming bound puts an upper bound on the rate of a code with a specified relative minimum distance. The Varshamov-Gilbert bound, on the other hand tells us a rate at which a code with a specified relative minimum

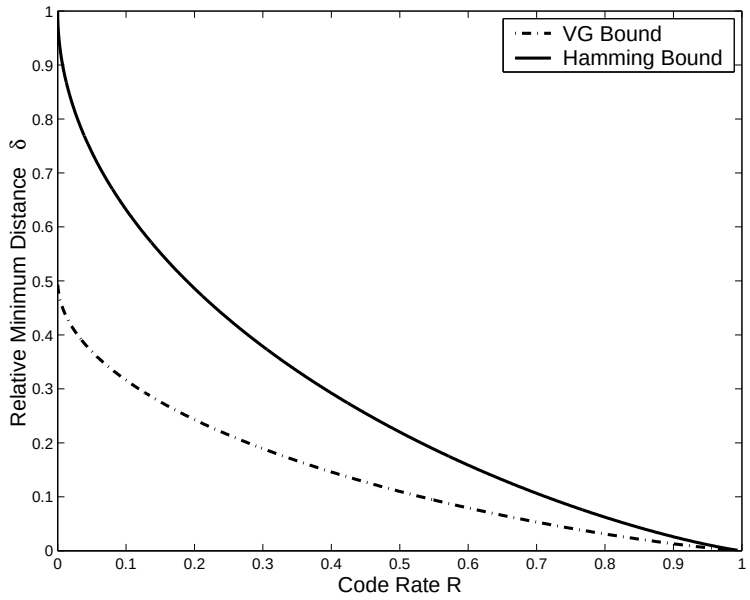

Figure 4.2: Asymptotic Hamming and Varshamov-Gilbert bounds

distance exists. We plot the two bounds in figure 4.2. In figure 4.2, the relative minimum distance as predicted by the two bounds is plotted as a function of code rate R. For a given rate R, the relative minimum distance δ for any code is smaller than the upper limit posed by the Hamming bound. Likewise, for a given rate R, codes that can achieve the δ value predicted by the VG bound exist.

No known binary codes actually meet the VG bound in the asymptotic sense. As an example, the relative minimum distance of Bose-Choudhary-Hocquenghem (BCH) codes goes to zero as n is increased [12, page 463]. For higher alphabets, Algebraic-Geometry (AG) codes are known to exceed the VG bound [72, page 81], [91].

4.2.5 The sphere-packing bound

We have been looking at results relating the code parameters. In this section, we look at the so-called sphere-packing bound, which is a result that deals with

the error probability performance of linear block codes over a binary-symmetric channel. A binary-symmetric channel is a channel that takes two inputs: 1 or 0; and returns two outputs: 1 or 0. With the crossover probability given by ϵ, the input is the same as the output with a probability $1-\epsilon$ [75, page 186]. The sphere-packing bound gives a lower bound on the performance of an (n,k) code that operates on a binary-symmetric channel with crossover probability given by ϵ.

Definition 4.9. Correct decoding *is said to have happened when the decoded codeword is exactly equal to the transmitted codeword.*

Result 4.13. *Let the probability of correct decoding in an (n,k) code be P_c. Then, if $\epsilon < 1/2$, the following inequality holds:*

$$\begin{aligned} P_c &\leq \frac{1}{M}\sum_{m=0}^{M-1}\sum_{w=0}^{\kappa}\binom{n}{w}\epsilon^w(1-\epsilon)^{n-w} + \frac{1}{M}\sum_{m=0}^{M-1}A_{\kappa+1,m}\epsilon^{\kappa+1}(1-\epsilon)^{n-\kappa-1} \\ &= \sum_{w=0}^{\kappa}\binom{n}{w}\epsilon^w(1-\epsilon)^{n-w} + \frac{1}{M}\epsilon^{\kappa+1}(1-\epsilon)^{n-\kappa-1}\sum_{m=0}^{M-1}A_{\kappa+1,m}, \end{aligned} \tag{4.15}$$

where κ is obtained from the Hamming bound as the largest possible error-correcting capability of an (n,k) linear block code (the maximum t that satisfies (4.5) for given n and k), $M = 2^k$, and $\sum_{m=0}^{M-1} A_{\kappa+1,m}$ is given by

$$\sum_{m=0}^{M-1}A_{\kappa+1,m} = 2^n - M\sum_{w=0}^{\kappa}\binom{n}{w}. \tag{4.16}$$

Proof. A detailed proof of this result can be obtained from [92, pages 161-164]. Let us outline the intuition behind the result.

Basically, given an n and a k, the Hamming bound limits the maximum number of errors that can be corrected per block. In the notation above, this

value is κ. Hence, a Hamming sphere of radius κ can be created around each codeword, with the codewords as the centers of the spheres. Note that, for general values of n and k, such Hamming spheres will *not* cover the entire space[1]. In the notation above, $\sum_{m=0}^{M-1} A_{\kappa+1,m}$ is the total number of n-tuples that do not fall within radius κ from any of the codewords.

It turns out that the optimal decoding strategy—the one that maximizes the probability of correct decoding averaged over all the codewords—is as follows: if the received code vector falls within a Hamming sphere centered around a codeword, decode the received vector to the codeword at the center; if the received code vector does not fall within radius κ to any codeword, decode it to a codeword that is at a Hamming distance $\kappa + 1$ to the received vector. The reason this strategy leads to optimal performance is because, for $\epsilon < 1/2$, $(1-\epsilon)^n > \epsilon(1-\epsilon)^{n-1} > \epsilon^2(1-\epsilon)^{n-2} > \ldots > \epsilon^n$. This implies that it is more beneficial to correct error-patterns of smaller weights. We refer the reader to [92, page 163] and [86, page 89] for more details. It can be shown that if the optimal decoding strategy is followed, the probability of correct decoding of an (n, k) code is exactly equal to the right hand side of (4.15). Hence, for a general (n, k) code, and a general decoding strategy, the probability of correct decoding is upper-bounded as in (4.15). □

Note that having an upper-bound on the probability of correct decoding is equivalent to having a lower-bound on the probability of codeword error. Specifically, for a given n and k, let P_c^{max} be the numerical value of the right hand side of (4.15). Basically, P_c^{max} denotes the highest possible probability of correct decoding with an (n, k) code. Hence, the lowest possible probability of error is simply $1 - P_c^{max}$.

[1]If the Hamming spheres so created cover the entire space, the code is called a perfect code [72, page 76]. A Hamming code is an example.

4.3 STI Codes: A brief recap

With respect to an uncoded system, STI codes were defined in Section 3.5.1 in Chapter 3 as those forward error correcting codes that reduce the average service time in the Stop-and-Wait ARQ system as compared to the uncoded transmission with the constellation size and symbol duration kept unchanged. It was then shown in Theorem 3.1 that a rate R_c code is an STI code if and only if $p_c < \frac{R_c - 1 + p_u}{R_c}$ (see (3.8)), where p_u and p_c are the probabilities of packet error for uncoded and coded transmissions respectively. Corollary 3.1.1 of Theorem 3.1 showed that all STI codes satisfy $R_c > 1 - p_u$ (see (3.9)).

In this chapter, we will merge the necessary and sufficient condition in (3.8) with the VG bound and the sphere-packing bound to study the existence of binary STI codes.

4.4 Existence of full-length STI codes

4.4.1 The preliminaries

Suppose an (n, k) block code is employed. In a packet of length L bits, there are $n_B = \lceil \frac{L}{k} \rceil$ code words in every transmitted data packet, where $\lceil \bullet \rceil$ represents the smallest integer greater than the argument. This is because if L is not a multiple of k, the data packet will be padded prior to the encoding—as we did in the numerical example that we considered in Section 3.5.1.1. The probability of packet error is given by $p_c = 1 - (1 - p_W)^{n_B}$, where p_W is the probability of codeword error.

Definition 4.10. *We define* full length *codes as those* (n, k) *block codes that have* $k = L$.

Observation 4.2. *For all full length codes,* $n_B = 1$ *and* $p_c = p_W$.

That is, the probability of packet error is the same as the probability of a codeword error. This serves to simplify the analysis and presentation; the results can easily be generalized to the case where a packet is composed of several messages, that is, when $n_B > 1$.

Since we consider $(L + P, L)$ codes, where $P = 1, 2, \ldots$, the code rates in question are

$$R = \frac{L}{L+P} \qquad P = 1, 2, \ldots \tag{4.17}$$

The question we are interested in is as follows: For what values of P does an $(L + P, L)$ STI linear block code exist?

The following two lemmas follow directly from the results developed in Section 3.5.1 in Chapter 3 , i.e., from the necessary and sufficient condition for being an STI code.

Lemma 4.1. *An* $(L+P, L)$ *code is an STI code if and only if the probability of packet error* $p_c < f(P)$*, where, for a non-negative integer* x*, the function* $f(x)$ *is defined as*

$$f(x) = \frac{(L+x)\,p_u - x}{L}. \tag{4.18}$$

Proof. The rate of an $(L + P, L)$ code is $R = \frac{L}{L+P}$. Substituting this into (3.8) yields the result.

□

Lemma 4.2. *For all* $(L+P, L)$ *STI codes,* $P \leq P_{max}$*, where* P_{max} *is defined as*

$$P_{max} = \left\lfloor \frac{Lp_u}{1-p_u} \right\rfloor, \tag{4.19}$$

$\lfloor \bullet \rfloor$ *being the largest integer smaller than the argument.*

Proof. Substitute $R = \frac{L}{L+P}$ into (3.9).

□

Lemma 4.2 tells us of the non-existence of STI codes for $P > P_{max}$. Next, we will look at two arguments—one based on the the Varshamov-Gilbert bound and the other based on sphere-packing bound—in conjunction with Lemma 4.1 to gain further insight into the existence of STI codes when $P \leq P_{max}$. The argument based on the VG bound establishes the existence of $(L + P, L)$ STI codes for a given P; the argument based on the sphere-packing bound establishes the non-existence of $(L + P, L)$ STI codes for a given P. We first present an outline of both the arguments qualitatively.

From Lemma 4.1, we know that an $(L + P, L)$ code is an STI code if and only if $p_c < f(P)$. The VG bound argument establishes the existence of an $(L + P, L)$ STI code by assessing whether an $(L + P, L)$ code with minimum distance large enough to guarantee $p_c < f(P)$ exists. On the other hand, the sphere-packing bound argument makes use of the sphere-packing lower bound on the probability of codeword error for an $(L + P, L)$ linear block code. Let us say that for a given P, we find that the lower-bound on the error-probability of an $(L + P, L)$ code is given by $\pi_{SP}(P)$. If $\pi_{SP}(P) > f(P)$, then no $(L + P, L)$ linear block code can satisfy $p_c < f(P)$. Hence, we can conclude that for the given P, no $(L + P, L)$ STI linear block code exists.

Throughout this section, we assume that the modulation employed is Binary Phase-Shift-Keying (BPSK) and that a hard-decision decoding is performed. This implies that a sharp decision on whether the received symbol represents a 0 or a 1 is made prior to passing the bit to the decoder of the error-correcting code. See [12, page 447] for further details and [12, page 440] for an alternative

kind of processing termed as soft-decision decoding. An implication of assuming hard-decision decoding is that from the point of view of the encoder/decoder, the channel is a binary symmetric channel (BSC)[2].

Throughout this discussion, we will assume that the E_b/N_0 for all the coded systems (different values of P lead to different coded systems) is kept the same as it is in the uncoded system. This implies reducing the transmitted power[3], and hence a higher probability that a coded bit will be received erroneously as P is increased. Specifically, the channel bit-error probability, when an $(L+P, L)$ code is employed will be given by (see [12, page 255])

$$q(L,P) = Q\left(\sqrt{\frac{2E_b}{N_0}\frac{L}{L+P}}\right), \tag{4.20}$$

where $Q(y) = \frac{1}{\sqrt{2\pi}}\int_y^\infty e^{-t^2/2}dt$ is the well-known Q-function [12, page 40]. Notice that for a given P, $q(L,P)$ is the cross-over probability in the BSC between the output of the encoder and the input to the decoder.

Generally speaking, as P increases, the requirement $p_c < f(P)$ in Lemma 4.1 becomes increasingly stringent since $f(P)$ is a decreasing function of P. On the other hand, the bit-error probability on the channel $q(L,P)$ increases. Hence as P increases, an $(L+P, L)$ code needs to be able to correct more errors on average to satisfy the increasingly stringent $p_c < f(P)$ requirement for being an STI code.

4.4.2 The VG bound argument

Definition 4.11. *For a non-negative integer x, let $t(x)$ be the smallest t such that*

[2]see Section 4.2.5 for the description of a BSC.
[3]See Section 3.2.1.4.

$$\sum_{i=0}^{t} \binom{L+x}{i} q(L,x)^i (1-q(L,x))^{L+x-i} > 1 - f(x) \tag{4.21}$$

where t takes on positive integer values, $q(L,x)$ is obtained from 4.20 and $f(x)$ is calculated from (4.18).

Definition 4.12. *For a non-negative integer x, define $d(x) = 2t(x) + 1$, where $t(x)$ is obtained from Definition 4.11.*

Lemma 4.3. *An $(L+x, L)$ linear block code with minimum distance at least $d(x)$ satisfies $p_c < f(x)$.*

Proof. An $(L+x, L)$ block code with minimum distance at least $d(x) = 2t(x)+1$ can correct all error patterns of weight $t(x)$ (see Result 4.5 in Section 4.2.2). By definition, $t(x)$ is the smallest value of t that satisfies (4.21) for a given x. Thus, the probability of error p_c of an $(L+x, L)$ code with minimum distance at least $d(x)$ satisfies

$$\begin{aligned} p_c &\leq \sum_{i=t(x)+1}^{L+x} \binom{L+x}{i} q(L,x)^i (1-q(L,x))^{L+x-i} \\ &= 1 - \sum_{i=0}^{t(x)} \binom{L+x}{i} q(L,x)^i (1-q(L,x))^{L+x-i} < f(x). \end{aligned} \tag{4.22}$$

□

Definition 4.13. *For a non-negative integer x, define the function $g(x)$ as*

$$g(x) = log_2 \left\{ \sum_{i=0}^{d(x)-2} \binom{L+x-1}{i} \right\} \tag{4.23}$$

where $d(x)$ is defined in Definition 4.12.

Lemma 4.4. *If x satisfies $x > g(x)$, then there exists an $(L + x, L)$ STI code.*

Proof. From the VG bound (see Section 4.2.3) and the definition of $g(x)$, if $x > g(x)$, then there exists an $(L + x, L)$ code with minimum distance at least $d(x)$. Now, applying Lemma 4.3, we note that such a code can achieve $p_c < f(x)$. Hence, by Lemma 4.1, the code is an STI code. □

4.4.3 The sphere-packing bound argument

Let $\pi_{SP}(x)$ denote the value of the lower bound on the probability of error on an $(L + x, L)$ linear block code. $\pi_{SP}(x)$ can be obtained using the sphere-packing bound as we discussed in Section 4.2.5. Note that in obtaining the $\pi_{SP}(x)$ value for an $(L + x, L)$ code, we use $q(L, x)$ as the cross-over probability of the BSC between the encoder and the decoder.

Lemma 4.5. *If, for an x value, $\pi_{SP}(x) > f(x)$, then no $(L + x, L)$ STI code exists.*

Proof. From the Sphere-packing bound, for all $(L+x, L)$ codes, the probability of codeword error $p_c(L, x) \geq \pi_{SP}(x)$. Thus, if $\pi_{SP}(x) > f(x)$, from Lemma 4.1, no $(L + x, L)$ code is an STI code. □

4.4.4 Numerical example

We consider $L = 21$ bits on an AWGN channel with E_b/N_0 fixed at 2 dB. The uncoded error probability p_u is $p_u = 1 - (1 - q(L, 0))^L = 0.5519$. Next, from (4.19), we find that $P_{max} = 25$. Hence, from Lemma 4.2, we know that all $(21 + P, 21)$ STI codes satisfy $P \leq 25$.

Let us first look at the VG bound argument. In figure 4.3, we plot the function $g(P)$. For all values of $1 \leq P \leq 25$, $g(P)$ is obtained by first computing

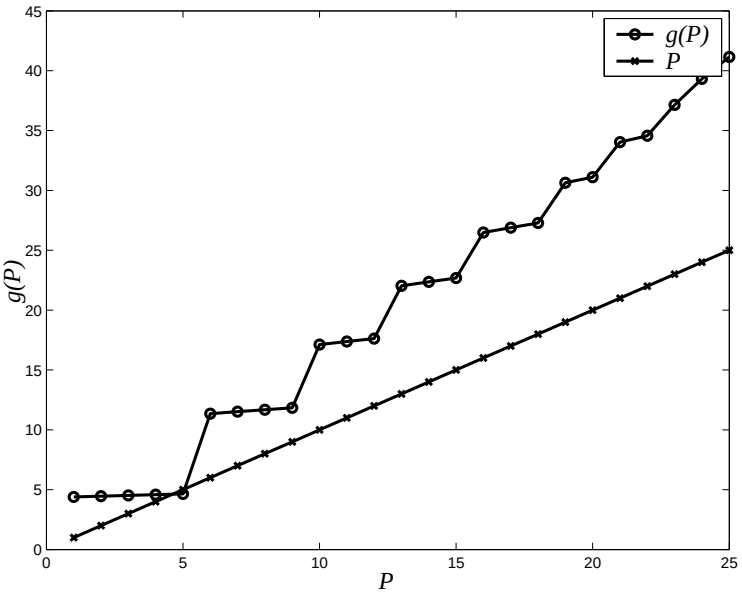

Figure 4.3: The only value where $P > g(P)$ is $P = 5$.

$t(P)$ (Definition 4.11), then $d(P)$ (Definition 4.12) and finally using $d(P)$ in (4.23). We notice from figure 4.3 that only at $P = 5$ do we have $P > g(P)$. Hence, we conclude that there exists $(26, 21)$ STI code.

Next, consider the sphere-packing bound argument. In figure 4.4, we plot $\pi_{SP}(P)$ and $f(P)$ for $1 \leq P \leq 25$. Notice that for all values of P except $P = 5$ and $P = 9$, $\pi_{SP}(P) > f(P)$, implying that except $P = 5$ and $P = 9$, there does not exist a $(21 + P, 21)$ STI linear block code. Consider, for example, the (31, 21) BCH code. Since there exists no STI code for $P = 10$, we do not expect the (31, 21) BCH code to be an STI code. Discrete event simulation verifies this. The average service time with the (31, 21) BCH code is found to be more than the average service time in the uncoded system ($\nu(31/21, \beta_{(31,21)}, 0.5519) = 0.936 < 1$, where $\beta_{(31,21)}$ represents the β value obtained with the (31, 21) BCH code).

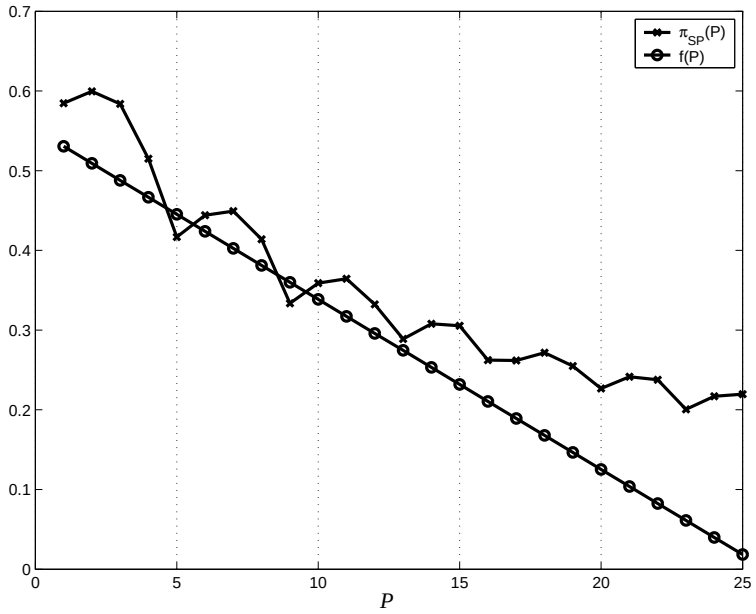

Figure 4.4: $\pi_{SP}(P) > f(P)$ for all P except $P = 5$ and $P = 9$. Hence, if $(P \neq 5)$ and $(P \neq 9)$, then no $(21 + P, 21)$ STI code exists.

4.5 Large packet length

We now consider the case when packet length $L \to \infty$. In this regime, the probability of uncoded packet error, $p_u \to 1$. Let's say $p_u = 1 - \epsilon$, where ϵ is a positive real number with $\epsilon \to 0$.

Lemma 4.6. *A rate R_c code is an STI code if and only if the probability of packet error for coded system satisfies $p_c < 1 - \epsilon/R_c$.*

Proof. Apply Theorem 3.1. □

We assume BPSK modulation over an AWGN channel and, as before, assume that the receiver performs hard-decision decoding. Let the bit error probability on the channel be p_b. For logical consistency in Lemma 4.7, we assume $p_b < 0.25$.

In Lemmas 4.7 and 4.8, we obtain sufficient conditions for the existence of STI codes using the asymptotic form of the VG bound and the channel capacity

theorem [75, page 191] respectively. We then proceed to explain the difference between the two results.

Lemma 4.7. *STI codes exist for all code rates that satisfy* $\epsilon < R_c < 1 - H(2p_b)$*, where* $H(\bullet)$ *stands for the binary entropy function [72, page 79].*

Proof. The lower bound, namely $R_c > \epsilon$, follows from Corollary 3.1.1 of Theorem 3.1. As for the upper bound, note that in the asymptotic regime, the weak law of large numbers [75, page 50] comes into play, and with high probability, $e_\infty = Np_b$ errors occur in a data block of length N. By the asymptotic form of the VG bound [72, page 81], for rates $R_c < 1 - H(2p_b)$, codes with minimum distance large enough to correct all the errors exist [93]. Thus, STI codes exist for all rates $\epsilon < R_c < 1 - H(2p_b)$. □

Lemma 4.8. *STI codes exist for all code rates that satisfy* $\epsilon < R_c < 1 - H(p_b)$*.*

Proof. The lower bound on R_c, as before, follows from Corollary 3.1.1 of Theorem 3.1. Regarding the upper bound, note that $1 - H(p_b)$ is the capacity of a binary symmetric channel with crossover probability p_b [75, page 187]. By the channel capacity theorem [75, page 191], for all rates less than capacity, there exist codes that achieve arbitrarily small probability of error. In other words, there exist codes that can satisfy the error probability requirement for being an STI code. Thus, STI codes exist in the range $\epsilon < R_c < 1 - H(p_b)$. □

When $p_b < 0.25$, $1 - H(2p_b) < 1 - H(p_b)$. Hence, the VG bound argument in Lemma 4.7 underestimates the maximum rate for which STI codes exist. This is because the VG bound argument is essentially a sufficient condition for the existence of codes with large enough minimum distance. The channel capacity result is obtained from an information-theoretic viewpoint, without any explicit consideration for the minimum distance of the codes employed on the channel. Hence, the difference between the two results.

4.6 Conclusions

We started with a necessary and sufficient condition for a rate R_c code to be an STI code, i.e., a code that improves average service time with respect to uncoded transmission, while keeping the bandwidth, constellation size and the symbol rate unchanged. We then studied the existence of STI linear block codes by merging the queueing-inspired necessary and sufficient condition with the VG bound and the sphere-packing bound. Invoking coding theoretic results revealed significant richness in the existence behavior of STI codes—such richness could not be apparent from the queueing analysis alone. We also considered the asymptotic case of large packet length, and provided sufficient conditions for the existence of STI codes in the asymptotic regime. We found that in this regime, arguments based on the channel capacity theorem and the asymptotic VG bound lead to slightly different results.

Thus, we have presented and illustrated a new approach for evaluating the impact of FEC on higher layer metrics. In doing so, we have also applied results from the disciplines of coding theory and queueing theory together in addressing a communication problem.

Chapter 5

Conclusions and Further Work

This thesis looked at cross-layer design which is a protocol design methodology that is finding increasing mention in the literature lately. The name cross-layer design seems self explanatory and makes one believe that no further explanation is needed. In reality, however, cross-layer design is a slippery concept with no clear-cut definition. In Chapter 2, we provided a definition for cross-layer design. We then presented the different interpretations that the term cross-layer design has assumed by creating a taxonomy of the existing cross-layer design proposals based on the kind or architecture violations they represent. We then distilled some existing results and presented them in a unified platform through which we made a preliminary assessment on which layers need to be coupled and how. We spelled out some open challenges in this area and looked at protocol design making use of the cooperation between nodes. While co-operation between the nodes is naturally allowed by the broadcast nature of the wireless medium, we found that incorporating node co-operation in protocol design would require significant violation of the layered architectures.

Chapter 2 has set a stage on which questions pertaining to cross-layer design can be meaningfully debated. Questions like whether cross-layer design *is*

the right way to go for wireless networks; or what kinds of cross-layer design proposals promise the most benefit in terms of architecture and performance; or what kind of information needs to be shared between the layers etc.; are important and challenging questions in their own right. However, to be able to answer these questions, one needs to be clear about what exactly cross-layer design implies. That is where Chapter 2 of this thesis has made a contribution. In light of Chapter 2, answering the aforementioned questions about cross-layer design—along with addressing the open challenges mentioned in Section 2.7—remain avenues for potential future work.

Chapter 3 looked at a specific instance of design coupling between layers without the creation of new interfaces – a category of cross-layer design – whereby the physical layer design for a Stop-and-Wait ARQ system was considered. The metric of interest was the link layer average service time. Guidelines for physical layer processing such that the metric of interest would be favorably affected were developed and illustrated with examples. Our guidelines enabled us to define Service Time Improving (STI) codes as those forward error correction codes that reduce the average service time with respect to the uncoded transmission, with the symbol-rate and constellation size kept unchanged. We came up with a necessary and sufficient condition for a rate R_c code to be an STI code. We also studied the impact of physical layer processing on the link-layer average delay and illustrated our results using numerical examples.

In Chapter 4, we merged the necessary and sufficient condition for being an STI code from Chapter 3 with the VG bound and the sphere-packing bound, and studied the existence of binary STI codes. We also considered the asymptotic regime of large packet length, and obtained sufficient conditions for the existence of STI codes from two viewpoints, namely, the channel capacity viewpoint and the asymptotic VG bound viewpoint. Interpreting the queueing inspired

conditions in light of results from coding theory forged a connection between the disciplines of queueing theory and coding theory, besides revealing some rich behavior which the application of ideas from either of the disciplines alone could not have.

One clear avenue for extension of the work in Chapter 3 and Chapter 4 is to consider other higher layer metrics. For example, one could make use of a similar methodology for designing physical layer processing with the view of optimizing TCP performance. While vertical calibrations involving TCP have been reported in the literature, our approach of comparing an original system to a modified system (Chapter 3) and the use of results from coding theory (Chapter 4) can potentially lead to new insights. Also, more elaborate channel models, for example time-varying or multi-user channels, can be incorporated into such studies too.

All in all, this thesis covered a wide range of topics, all tied to the common theme of cross-layer design. Thus, this thesis highlighted the different viewpoints and tools that are relevant to the different aspects of protocol design, and also found connections between the different viewpoints.

Bibliography

[1] H. Rheingold, *Smart Mobs : The Next Social Revolution*. Perseus Publishing, 2002. [Online]. Available: www.smartmobs.com

[2] J. Geier, *Wireless LANs*, 2nd ed. SAMS, 2001.

[3] A. Jamalipour, *The Wireless Mobile Internet*. John Wiley & Sons, 2003.

[4] W. C. Y. Lee, *Mobile Communications Engineering*, 2nd ed. McGraw-Hill Telecommunications, 1998.

[5] A. J. Goldsmith and S. B. Wicker, "Design Challenges for Energy-Constrained Ad Hoc Wireless Networks," *IEEE Wireless Communications Magazine*, pp. 8–27, Aug. 2002.

[6] K. Xu, M. Gerla, and S. Bae, "How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks?" in *Proc. IEEE GlobeCom'02*, 2002.

[7] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[8] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-hoc Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

[9] T. E. Hunter and A. Nosratinia, "Performance Analysis of Coded Cooperation Diversity," in *Proc. IEEE International Conference on Communications (ICC'03)*, Anchorage, 2003.

[10] V. Kawadia and P. R. Kumar, "A Cautionary Perspective on Cross-Layer Design," *IEEE Wireless Communications Magazine*, vol. 12, no. 1, pp. 3–11, Feb. 2005.

[11] L. L. Xie and P. R. Kumar, "A Network Information Theory for Wireless Communication: Scaling Laws and Optimal Operation," *IEEE Transactions on Information Theory*, vol. 50, no. 5, pp. 748 – 767, 2004.

[12] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw Hill, 2000.

[13] R. A. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[14] H. A. Tuan and M. Motani, "Buffer and Channel Adaptive Modulation for Transmission over Fading Channels," in *Proc. IEEE International Conference on Communications (ICC'03)*, Anchorage, 2003.

[15] A. Goldsmith and P. Varaiya, "Capacity of fading channel with channel side information," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1986–1992, Nov. 1997.

[16] R. Knopp and P. A. Humblet, "Information Capacity and Power Control in single-cell multiuser communications," in *Proc. IEEE International Conference on Communications (ICC'95)*, 'Seattle, 1995.

[17] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-Layer Design for Wireless Networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, Oct. 2003.

[18] D. Tse and S. Hanly, "Multi-Access Fading Channels: Part I: Polymatroid Structure, Optimal Resource Allocation and Throughput Capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2796–2815, Nov. 1998. [Online]. Available: http://degas.eecs.berkeley.edu/∼dtse/opp_comm.html

[19] ——, "Multi-Access Fading Channels: Part II: Delay-Limited Capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2816–2831, Nov. 1998. [Online]. Available: http://degas.eecs.berkeley.edu/∼dtse/opp_comm.html

[20] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic Beamforming using Dumb Antennas," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1277–1294, June 2002. [Online]. Available: http://degas.eecs.berkeley.edu/∼dtse/opp_comm.html

[21] J. Day, "The Reference Model for Open Systems Interconnection," in *Computer Network Architectures and Protocols*, 2nd ed., C. A. Sunshine, Ed. New York: Plenum Press, 1989.

[22] D. E. Perry and A. L. Wolf, "Foundations for the Study of Software Architecture," *ACM SIGSOFT Software Engineering Notes*, vol. 17, no. 4, pp. 40–52, Oct. 1992.

[23] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. New Jersey: Prentice Hall, 1992.

[24] B. M. Leiner, R. Cole, J. Postel, and D. Mills, "The DARPA Internet protocol suite," *IEEE Communications Magazine*, vol. 23, pp. 29–34, 1985.

[25] W. Stallings, *Networking Standards : A Guide to OSI, ISDN, LAN and MAN Standards*. Addison-Wesley, 1993.

[26] D. D. Clark, "The Design Philosophy of the DARPA Internet Protocols," in *Proc. ACM Symposium on Communications Architectures and Protocols (SIGCOMM'88)*, 1988.

[27] C. A. Sunshine, "A Brief History of Computer Networking," in *Computer Network Architectures and Protocols*, 2nd ed., C. A. Sunshine, Ed. New York: Plenum Press, 1989.

[28] A. S. Tanenbaum, *Computer Networks*, 3rd ed. Prentice-Hall, Inc., 1996.

[29] J. D. Day, "The (Un)Revised OSI Reference Model," *ACM Computer Communication Review*, vol. 25, pp. 39–55, Oct. 1995.

[30] V. T. Raisinghani and S. Iyer, "Cross-Layer Design Optimizations in Wireless Protocol Stacks," *Computer Communications*, vol. 27, pp. 720 –724, 2004.

[31] G. H. Cooper, "An Argument for Soft Layering of Protocols," Master's thesis, Masachussets Institute of Technology, 1983.

[32] D. D. Clark and D. L. Tennenhouse, "Architectural Considerations for a New Generation of Protocols," in *Proc. ACM Symposium on Communications Architectures and Protocols (SIGCOMM'90)*, Philadelphia, 1990.

[33] L. Svobodova, "Implementing OSI Systems," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 1115–1130, 1989.

[34] W. R. Stevens, *TCP/IP Illustrated, Volume 1.* Addison Wesley Longman, Inc., 1994.

[35] H. Balakrishnan and V. N. Padmanabhan and S. Seshan and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, Dec. 1997.

[36] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC protocol for Multi-hop Wireless Networks," in *Proc. ACM Annual International Conference on Mobile Computing and Networking (MobiCom'01)*, Rome, July 2001.

[37] L. Tong, Q. Zhao, and G. Mergen, "Multipacket Reception in Random Access Wireless Networks: From Signal Processing to Optimal Medium Access Control," *IEEE Communications Magazine*, vol. 39, no. 11, pp. 108–112, Nov. 2001.

[38] X. Yu, "Improving TCP Performance over Mobile Ad Hoc Networks by Exploiting Cross-Layer Information Awareness," in *Proc. ACM Annual International Conference on Mobile Computing and Networking (MobiCom'04)*, Philadelphia, Oct. 2004.

[39] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic Media Access for Multirate Ad-hoc Networks," in *Proc. ACM Annual International Conference on Mobile Computing and Networking (MobiCom'02)*, Atlanta, Sept. 2002.

[40] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia, "Exploiting Medium Access Diversity in Rate Adaptive Wireless LANs," in *Proc. ACM Annual International Symposium on Mobile Computing and Networking (MobiCom'04)*, Philadelphia, Oct. 2004.

[41] D. Qiao and S. Choi, "Goodput Enhancement of 802.11a Wireless LAN via Link Adaptation," in *Proc. IEEE International Conference on Communications (ICC'01)*, Helsinki, 2001.

[42] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, pp. 118–133, 1997.

[43] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Enhancing throughput over wireless LANs using Channel State Dependent Packet Scheduling," in *Proc. IEEE Infocom'96*, 1996.

[44] S. Hara, A. Ogino, M. Araki, M. Okada, and M. Morinaga, "Throughput performance of SAW-ARQ protocol with adaptive packet length in mobile packet data transmission," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 3, pp. 561–569, Aug. 1996.

[45] L. Larzon, U. Bodin, and O. Schelen, "Hints and Notifications," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'02)*, Orlando, 2002.

[46] G. Xylomenos and G. C. Polyzos, "Quality of service support over multi-service wireless internet links," *Computer Networks*, vol. 37, no. 5, pp. 601–615, 2001.

[47] G. Dimić, N. D. Sidiropoulos, and R. Zhang, "Medium Access Control - Physical Cross-Layer Design," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 40–50, 2004.

[48] T. Elbatt and A. Ephremides, "Joint Scheduling and Power Control for Wireless Ad Hoc Networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 74 – 85, 2004.

[49] M. Chiang, "To Layer or Not To Layer: Balancing Transport and Physical Layers in Wireless Multihop Networks," in *Proc. IEEE Infocom'04*, Hong Kong, Mar. 2004.

[50] U. C. Kozat, I. Koutsopoulos, and L. Tassiulas, "A Framework for Cross-Layer Design of Energy-efficient Communication with QoS Provisioning in Multi-hop Wireless Networks," in *Proc. IEEE Infocom'04*, Hong Kong, Mar. 2004.

[51] T. A. Elbatt, S. V. Krishnamurthy, D. Connors, and S. Dao, "Power Management for Throughput Enhancement in Wireless Ad-Hoc Networks," in *Proc. IEEE International Conference on Communications (ICC'2000)*, 2000.

[52] A. Muqattash and M. Krunz, "A Single-Channel Solution for Transmission Power Control in Wireless Ad Hoc Networks," in *Proc. ACM Annual International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc'04)*, Tokyo, 2004.

[53] V. Kawadia and P. R. Kumar, "Principles and Protocols for Power Control in Ad Hoc Networks." *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 76–88, Jan. 2005.

[54] K. Sundaresan and R. Sivakumar, "A Unified MAC Layer Framework for Ad-Hoc Networks with Smart Antennas," in *Proc. ACM Annual International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc'04)*, Tokyo, 2004.

[55] L. Tong, V. Naware, and P. Venkitasubramaniam, "Signal Processing in Random Access," *IEEE Signal Processing Magazine*, vol. 21, no. 5, Sept. 2004.

[56] V. Srivastava and M. Motani, "Combining Communication and Queueing with Delay Constraints in Wireless Ad-Hoc Networks," in *Proc. International Conference on Information and Communication Systems (ICICS'03)*, Singapore, Dec. 2003.

[57] ——, "Coding Meets Queueing: Quantifying the impact of Forward Error Correction on Higher Layers," in *Proc. International Symposium on Information Theory and its Applications (ISITA)'04*, Parma, 2004.

[58] D. Barman, I. Matta, E. Altman, and R. E. Azuozi, "TCP Optimization through FEC, ARQ and Transmission Power Tradeoffs," in *Proc. International Conference on Wired/Wireless Internet Communications*, Frankfurt, 2004.

[59] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-Layer Combining of Adaptive Modulation and Coding with Truncated ARQ Over Wireless Links," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1746 – 1755, 2004.

[60] Q. Wang and M. A. Abu-Rgheff, "Cross-Layer Signalling for Next-Generation Wireless Systems," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'03)*, New Orleans, 2003.

[61] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-Layering in Mobile Ad Hoc Network Design," *IEEE Computer Magazine*, pp. 48–51, Feb. 2004.

[62] R. Braden, T. Faber, and M. Handley, "From Protocol Stack to Protocol Heap - Role-Based Architecture," in *Proc. Hot Topics in Networking (HOTNETS I)*, Princeton, NJ, 2002.

[63] F. Bai, G. Bhaskara, and A. Helmy, "Building the Blocks of Protocol Design and Analysis - Challenges and Lessons Learned from Case Studies on Mobile Ad Hoc Routing and Micro-Mobility Protocols," *ACM SIGCOMM Computer Communications Review*, vol. 34, no. 3, pp. 57–69, July 2004.

[64] C. Barrett, M. Drozda, A. Marathe, and M. V. Marathe, "Characterizing the Interaction Between Routing and MAC Protocols in Ad-Hoc Networks," in *Proc. ACM Annual International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc'02)*, Lausanne, June 2002.

[65] ——, "Analyzing Interaction Between Network Protocols, Topology and Traffic in Wireless Radio Networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'03)*, New Orleans, Mar. 2003.

[66] R. Jiang, V. Gupta, and C. V. Ravishankar, "Interactions Between TCP and the IEEE 802.11 MAC Protocol," in *Proc. DARPA Information Survivability Conference and Exposition*, 2003.

[67] Z. Fu, P. Zefros, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," in *Proc. IEEE Infocom'03*, San Francisco, 2003.

[68] S. Bansal, R. Shorey, and A. A. Kherani, "Performance of TCP and UDP Protocols in Multi-Hop Multi-Rate Wireless Networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'04)*, 2004.

[69] G. Carneiro, J. Ruela, and M. Ricardo, "Cross-layer design in 4G wireless terminals," *IEEE Wireless Communications*, vol. 11, no. 2, pp. 7–13, Apr. 2004.

[70] M. Agarwal, J. H. Cho, L. Gao, and J. Wu, "Energy Efficient Broadcast in Wireless Ad Hoc Networks with Hitch-hiking," in *Proc. IEEE Infocom'04*, 2004.

[71] C. Chien, M. B. Srivastava, R. Jain, P. Lettieri, V. Aggarwal, and R. Sternowski, "Adaptive Radio for Multimedia Wireless Links," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 793–813, 1999.

[72] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Eaglewood Cliffs, New Jersey: Prentice Hall, 1995.

[73] B. Zhao and M. C. Valenti, "Distributed turbo coded diversity for the relay channel," *IEE Electronics Letters*, vol. 39, no. 10, pp. 786 – 787, 2003.

[74] ——, "Practical relay Networks: A Generalization of Hybrid-ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 7–18, Jan. 2005.

[75] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[76] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Eaglewood Cliffs, New Jersey: Prentice Hall, 1983.

[77] C. Barakat and E. Altman, "Bandwidth tradeoff between TCP and link-level FEC," *Computer Networks*, vol. 39, no. 2, pp. 133–150, 2002.

[78] L. Gallucio, G. Morabito, and S. Pallazo, "An Analytical Study of a Trade-off Between Transmission Power and FEC for TCP Optimization in Wireless Networks," in *Proc. IEEE Infocom'03*, San Francisco, 2003.

[79] N. Celandroni and F. Potortì, "Maximising Single Connection TCP Goodput By Trading Bandwidth for BER," *International Journal for Communication Systems*, vol. 16, no. 1, pp. 63–79, 2003.

[80] M. M. Krunz and J. G. Kim, "Fluid Analysis of Delay and Packet Discard Performance for QoS Support in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 2, pp. 384–395, 2001.

[81] A. Chockalingam, M. Zorzi, and V. Tralli, "Wireless TCP performance with Link Layer FEC/ARQ," in *Proc. International Conference on Communications (ICC'99)*, 1999, pp. 1212– 1216.

[82] A. Stamoulis and N. Al-Dhahir, "Impact of Space-Time Block Codes on 802.11 Network Throughput," *IEEE Transactions on Wireless Communications*, vol. 2, no. 5, pp. 1029–1039, Sept. 2003.

[83] A. L. Toledo and X. Wang, "Effect of MIMO Wireless Channels on TCP," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'04)*, Atlanta, Mar. 2004.

[84] S. Haykin, *An Introduction to Analog and Digital Communications*, 1st ed. John Wiley & Sons, Inc., 1989.

[85] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. Prentice-Hall, Inc., 2001.

[86] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, 2nd ed. Cambridge, Mass.: MIT Press, 1972.

[87] F. Vacirca, A. D. Vendictis, and A. Baiocchi, "Investigating Interactions between ARQ Mechanisms and TCP over Wireless Links," in *Proc. European Wireless 2004*, Barcelona, Feb. 2004.

[88] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., 1994.

[89] E. N. Gilbert, "A Comparison of Signalling Alphabets," in *Algebraic Coding Theory: History and Development*, I. J. Blake, Ed. Dowden, Hutchingson and Ross Inc., 1973.

[90] R. R. Varshamov, "Estimate of the Number of Signals in Error Correcting Codes," in *Algebraic Coding Theory: History and Development*, I. J. Blake, Ed. Dowden, Hutchingson and Ross Inc., 1973.

[91] E. Berlekamp, "The Performance of Block Codes," *Notices of the AMS*, vol. 49, no. 1, pp. 17–22, 2002.

[92] R. G. Gallager, *Information Theory and Reliable Communication*. John Wiley and Sons, 1968.

[93] A. R. Calderbank, "The Art of Signaling: Fifty Years of Coding Theory," *IEEE Transactions on Information Theory*, vol. 10, pp. 2561 – 2595, Oct. 1998.