Kevin Cui, Matt Yundt, Yajun Chen

COEN 233 Computer Networks Spring 2013

HIGH SPEED ETHERNET: 40/100GE TECHNOLOGIES

TABLE OF CONTENTS

2 INTRODUCTION

2.1 Objective

Nowadays, driven by high definition video and the penetration of high-speed broadband access, the rising volume of consumer IP traffic is bolstering the overall IP growth rate. So, High speed LAN interfaces such as 40Gbitls Ethernet (40GbE) and 100Gbit/s Ethernet (l00GbE) have been standardized to support huge traffic demands.

Here we are trying to compare two FEC, Reed-Solomon (RS) and Binary Cyclic code as error correction encoding scheme candidates for 802.3ap's FEC layer. We try to simulate FEC performance in Matlab since it is an ideal tool for simulating digital signal and communication system due to its easy scripting language, environment setup due to its abundant simulink block library, and its data plotting capabilities. We are evaluating the performance by using its Bertool, a bit-error-rate testing and plotting tool.

2.2 What is the problem

To addresses critical challenges facing technology providers today:

The growing number of applications with demonstrated bandwidth needs far more exceeding existing Ethernet capabilities, by providing a larger, more durable bandwidth pipeline.

“The capacity constraints posed by the rapid growth of video rich Ethernet traffic”

According to Sterling Perrin, Senior Analyst, Heavy Reading, “heavy Reading network operator surveys have consistently shown strong and immediate operator demand for 100 Gigabit Ethernet, driven by the rapid increase in global IP traffic and exhaustion of existing 10 Gigabit networks.”

For certain research project, such as those related to weather, energy, probing our universe etc, “leveraging petascale data and information exchange is essential.”

More interests and deployment in metro and regional networks are growing.

Need a standard that is as compatible as to many existing application as possible.

2.3 Why This is a Project Related to the This Class

This class covers Ethernet. 40/100G Ethernet just a amendment to Ethernet.

2.4 Why Other Approach is No Good

At this point we don't know if the approach we propose are better. We should know the answer when we finish the project defense.

2.5 Why You Think Your Approach is Better

We will know why when the results and analysis come out if the approach we proposed is really better.

3 THEORETICAL BASES AND LITERATURE REVIEW

3.1 What is Ethernet?

Ethernet is the most popular and competing family of wired LAN technologies. Logically, Ethernet works at layer 1 and layer 2 of the OSI model as is illustrated in Figure 3-1. Ethernet was commercially introduced in 1980 and standardized in 1985 as IEEE 802.3. Ethernet has largely replaced competing wired LAN technologies.

The original 10BASE5 Ethernet used coaxial cable as a shared medium. Later the coaxial cables were replaced by twisted pair and fiber optic links in conjunction with hubs or switches. Data rates were periodically increased from the original 10 megabits per second to 100 gigabits per second, which you will see in the following chapters of this article.

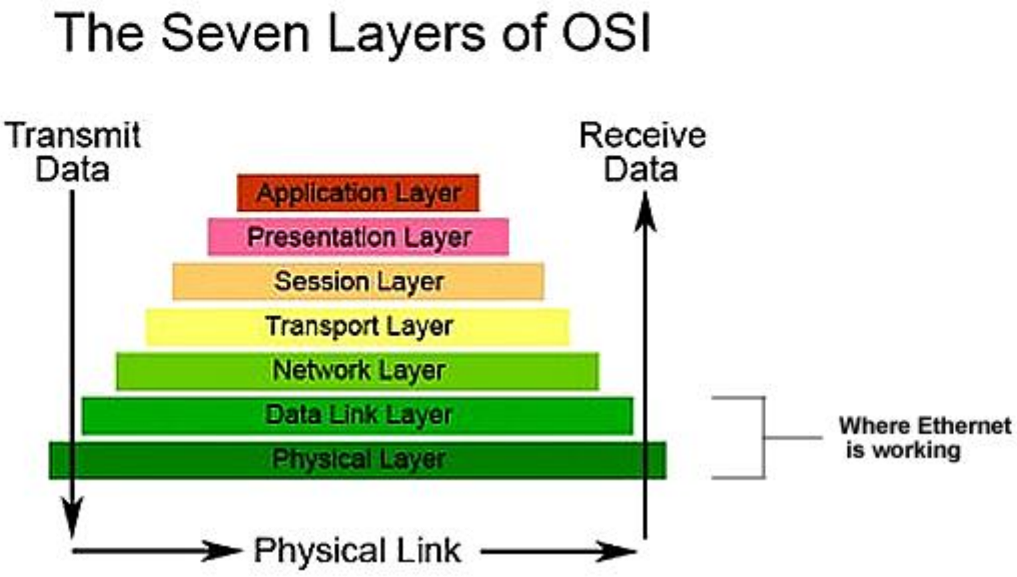


Figure 3-1 Where Ethernet is Working in the OSI Model

3.2 IEEE 802.3 Ethernet Architecture

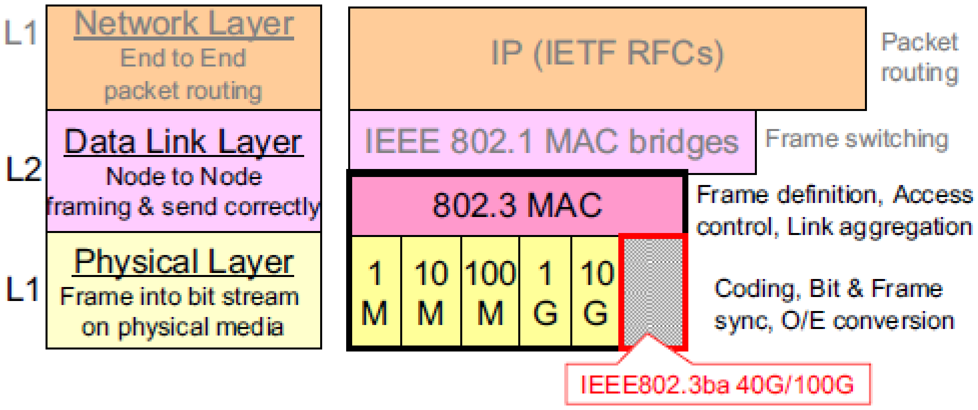

L1
Network Layer
End to End packet routing
L2
Data Link Layer
Node to Node framing & send correctly
L1
Physical Layer
Frame into bit stream on physical media
IP (IETF RFCs)
Packet routing
IEEE 802.1 MAC bridges
Frame switching
802.3 MAC
Frame definition, Access control, Link aggregation
1 M
10 M
100 M
1 G
10 G
Coding, Bit & Frame sync, O/E conversion
IEEE802.3ba 40G/100G

Figure 3-2 Ethernet Evolution

As we stated in section 2.1, Ethernet is working at layer 1 and layer 2 of the OSI model. Here we have a more precise description of the features of Ethernet architecture. Ethernet is located at:

- Lower half of L2: common MAC sub-layer
- L1: multiple PHYs dependent on speeds and physical media

Besides, it is now composed of 1Mbps, 10Mbps, 100Mbps, 1Gbps, 10Gbps as well as 40/100Gbps which we are introducing in this article. You can find more details about Ethernet Architecture in Fugure 3-2.

3.2.1 Ethernet MAC Basics

In IEEE802.3, MAC frame format is a kind of “bible”. Historically, the1500 Bytes frame size was restricted by CSMA/CD protocol. It is rarely extended since backward compatibility is considered the first priority.

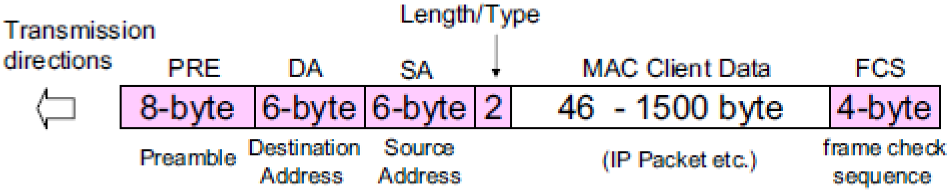


Figure 3-3 Ethernet Frame

An Ethernet frame uses a unique 48-bits destination address and another unique 48-bits source address to find its destination device and to tell the device from which source device it comes from. Accordingly, these addresses are called MAC addresses and are basically classified into three categories, unicast MAC addresses, multicast MAC addresses and broadcast MAC addresses, which are briefly explained in Table 3-1. MAC addresses are most often assigned by the manufacturer of a network interface card (NIC) and are stored in its hardware, the card's read-only memory, or some other firmware mechanism. To keep the integration of a frame, a 32-bit error-checking data, called FCS is put at the end of a frame calculated using some algorithm. Figure 3-3 shows the basic content of an Ethernet frame.

MAC address Type	Description & Function	Example
Unicast MAC Address	If the least significant bit of the most significant octet of a 48-bit destination MAC address is set to 0, it is an unicast address and if it were put in a frame as a destination address the frame will transmit to all devices in a collision domain and only the device with this address will accpet the frame.	0x100000001135

Multicast MAC Address	If the least significant bit of the most significant octet of a 48-bit destination MAC address is set to 1, it is a multicast address and if it were put in a frame as a destination address the frame will transmit to all devices in a collision domain and one or more devices will selectively accept it based on some criteria.	0x010000001135
Broadcast MAC Address	If all of a 48-bit destination MAC address is set to 1, it is an broadcast address and if it were put in a frame as a destination address the frame will transmit to all devices in a collision domain and all devices will accept the frame.	Should be always 0xFFFFFFFFFFFF

Table 3-1 Descriptions and functions of three types of Ethernet MAC address

3.2.2 Ethernet PHY Basics

PHY is an abbreviation for the physical layer of the OSI model. An instantiation of PHY connects a link layer device, eg. MAC as we introduced above, to a physical medium such as an optical fiber or copper cable. A PHY device typically includes a Physical Coding Sublayer (PCS) and a Physical Medium Dependent (PMD) layers. The PCS encodes and decodes the data that is transmitted and received. The purpose of the encoding is to make it easier for the receiver to recover the signal. So, we have the following functions of PHY:

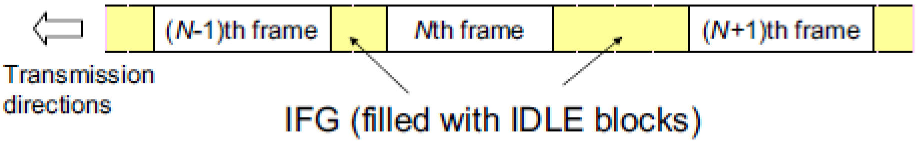


Figure 3-4

- Generating a nB/mB block-encoded data stream: Sending n Binary bit via m Binary bit 4B/5B for FE, 8B/10B for GbE, and 64B/66B for 10GbE and above
- Provide at least 12-byte inter-frame spacing
- Inter-frame gap (IFG), depicted in Figure 3-4 is filled with IDLE blocks IDLE is indispensable for

  1) continuous bit-stream generation, and

  2) block & frame delineation by using an IDLE as a marker, and

  3) asynchronous clock domain crossover by inserting / deleting an IDLE

3.2.2 MII and PHY Sublayers

Media Independent Interface (MII)

The Media Independent Interface (MII) was originally defined as a standard interface used to connect a Fast Ethernet (i.e., 100 Mbit/s) MAC-block to a PHY chip.

Being media independent means that different types of PHY devices for connecting to different media (i.e. Twisted pair copper, fiber optic, etc.) can be used without redesigning or replacing the MAC hardware. The MII bus (standardized by IEEE 802.3u) connects different types of PHYs (Physical Transceivers) to Media Access Controllers (MAC). Thus any MAC may be used with any PHY, independent of the network signal transmission media. The MII bus transfers data using 4-bit words (nibble) in each direction (4 transmit data bits, 4 receive data bits). The data is clocked at 25 MHz to achieve 100 Mbit/s speed.

Physical Coding Sublayer (PCS)

The Physical Coding Sublayer (PCS) helps to define physical layer specifications (speed and Duplex modes, etc.) for networking protocols like Fast Ethernet, Gigabit Ethernet and 10 Gigabit Ethernet. This sublayer performs auto-negotiation and coding such as 8b/10b encoding.

Physical Medium Attachment (PMA)

Physical Medium Attachment (PMA) sublayer performs PMA framing, octet synchronization/detection, and scrambling/descrambling.

Physical Medium Dependent (PMD)

Physical Medium Dependent (PMD) sublayer helps to define the physical layer of computer network protocols. They define the details of transmission and reception of individual bits on a physical medium. These responsibilities encompass bit timing, signal encoding, interacting with the physical medium, and the properties of the cable, optical fiber, or wire itself. Common examples are specifications for Fast Ethernet, Gigabit Ethernet and 10 Gigabit Ethernet defined by the Institute of Electrical and Electronics Engineers (IEEE).

Figure 3-5 illustrates the MII and the Sublayers.

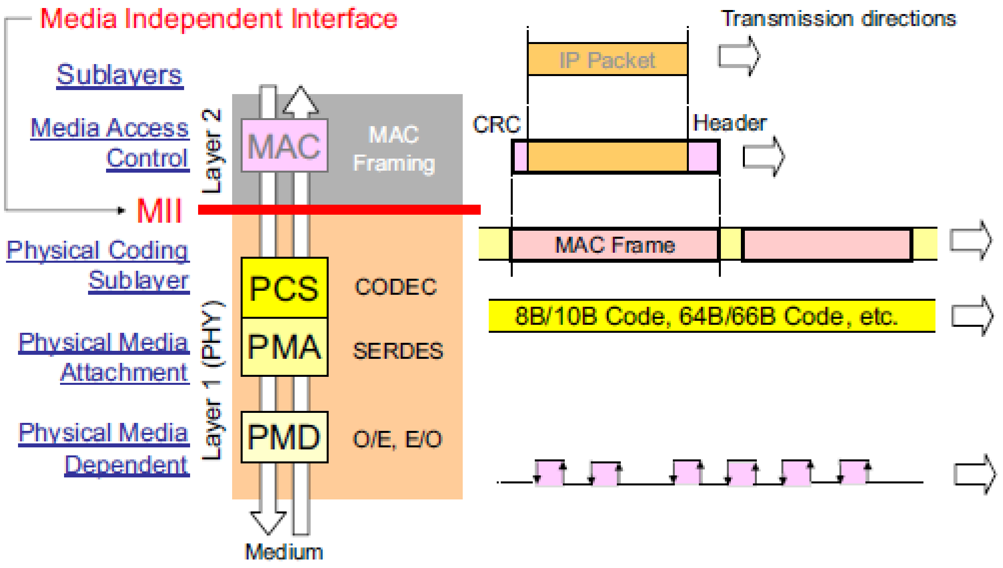


Figure 3-5

3.3 Why We Need 40/100GbE?

Driven by high definition video and the penetration of high-speed broadband access, the rising volume of consumer IP traffic is bolstering the overall IP growth rate. The growth rate per year ranges from 20 to 35%. If the growth of the traffic continues over 10 years, the resultant traffic volume is about 20 times the current traffic.

The effort to develop the next generation of Ethernet began in July 2006, when the IEEE 802.3 Working Group formed the Higher Speed Study Group (HSSG) to investigate Ethernet operation beyond the maximum rate of 10 Gigabit per second.

The initial focus of the HSSG was on the bandwidth requirements for core networking and aggregation applications in data centers, Internet exchanges, and service provider peering points, which are being driven beyond existing capabilities by high bandwidth applications such as video on demand (VOD). However, it was found that while bandwidth requirements for network aggregation applications are doubling approximately every 18 months, server I/O bandwidth requirements are on a different growth curve, doubling approximately every 24 months. Figure 3-6 depicts this trend.

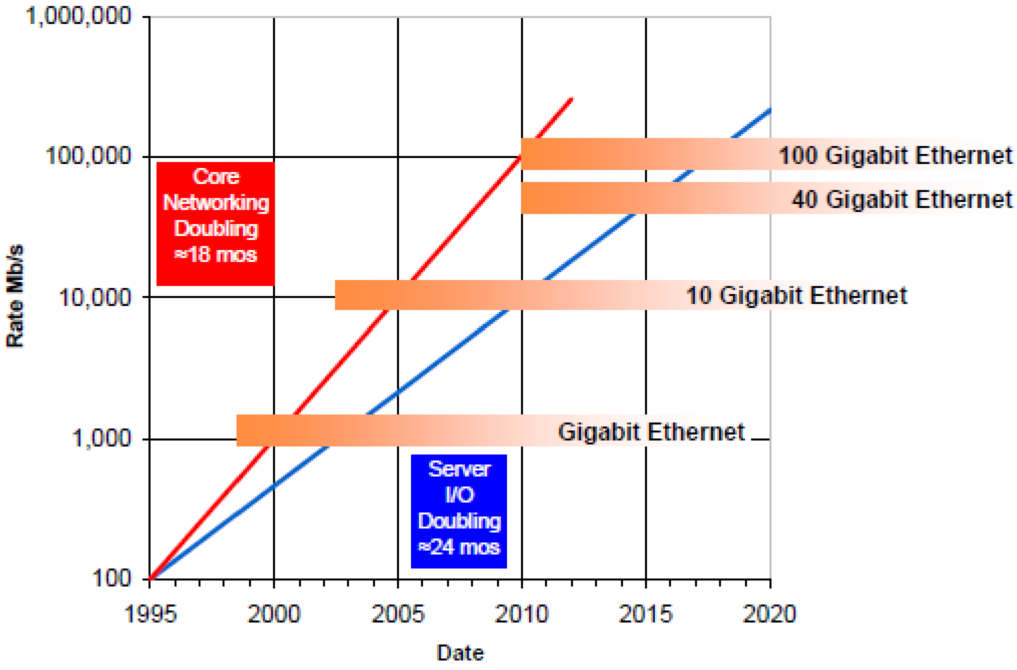


Figure 3-6

3.4 Why Both 40GbE and 100GbE?

Based on these findings, the HSSG determined that two new rates of operation for Ethernet were needed: 40 Gigabit per second for computing and server applications and 100 Gigabit per second for network aggregation applications. Physical layer specifications, appropriate for each application space, were chosen.

- For computing and server applications at 40 Gigabit per second, three distance objectives were selected: at least 1m over a backplane; at least 10m over a copper cable assembly; and at least100m on OM3 multimode fiber (MMF). Figure 3-7 shows the x86 server demands for the 40GbE in the history and following years.

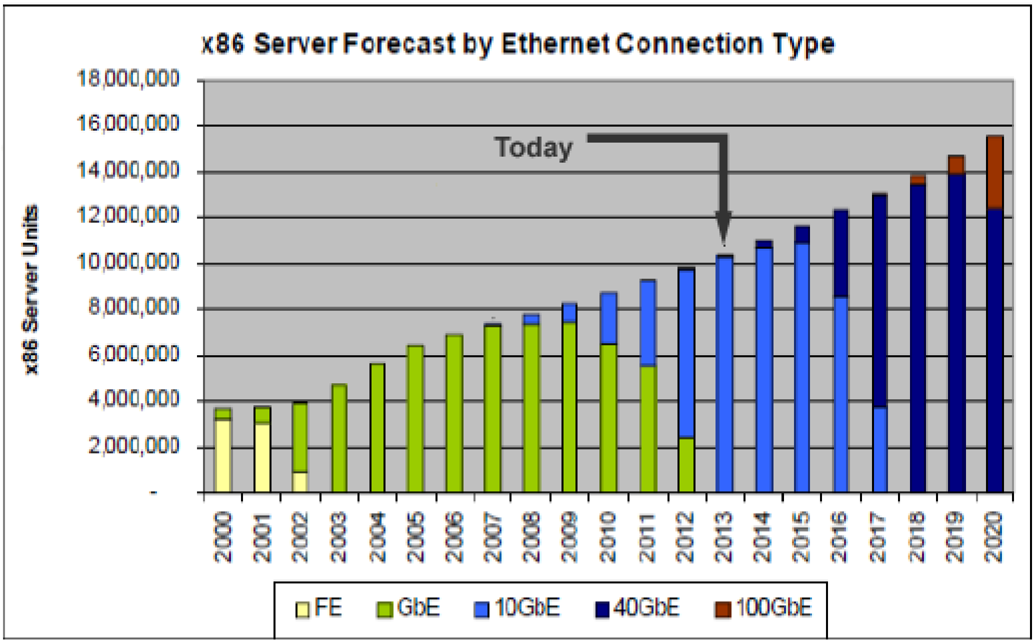


Figure 3-7

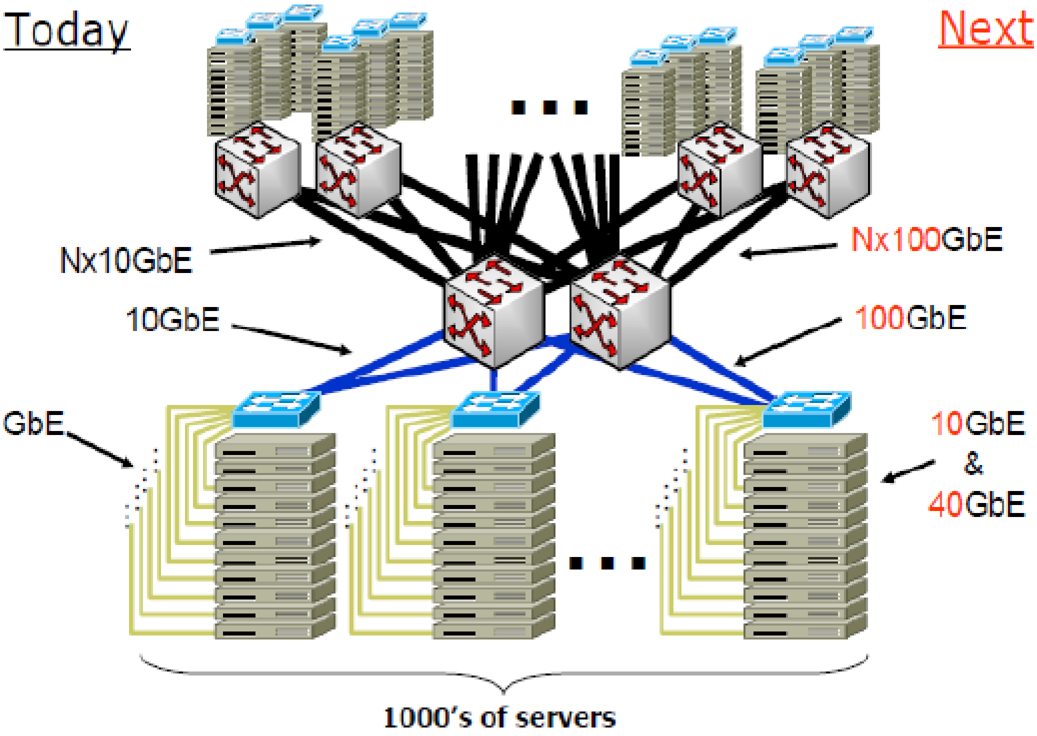


Figure 3-8

- For core networking and aggregation applications at 100 Gigabit per second, four distance objectives were selected: at least10m over a copper cable assembly; at least 100m on OM3 MMF; at least 10km on single mode fiber (SMF); and at least 40km on

SMF. Subsequently, a new objective to do 40 Gigabit per second over at least 10km on SMF was added. Figure 3-8 illustrates the role of 100GbE in the future.

3.5 IEEE 802.3ba 40/100GbE Objectives

The first objective is to add new PHYs. As is illustrated in Figure 3-9, 40G and 100G PHYs will be supported in new specifications. Table 3-1 is a summary of physical layer specifications.

- Provide physical layer specifications which support 40 gigabit per second operation over:

  at least 10km on single mode fiber (SMF)

  at least 100m on OM3 multi-mode fiber (MMF)

  at least 10m over a copper cable assembly

  at least 1m over a backplane

- Provide physical layer specifications which support 100 gigabit per second operation over:

  at least 40km on SMF

  at least 10km on SMF

  at least 100m on OM3 MMF

  at least 10m over a copper cable assembly

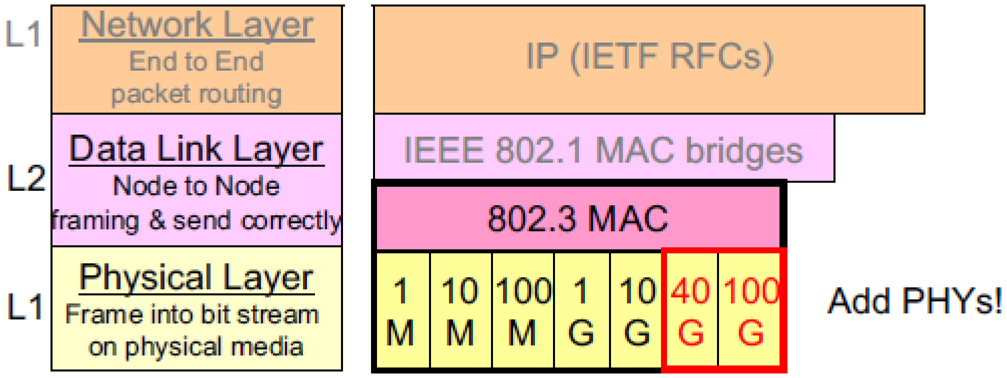


Figure 3-9

	40 Gigabit Ethernet	100 Gigabit Ethernet
At least 1m backplane	V	
At least 10m copper cable	V	V
At least 100m OM3 MMF	V	V
At least 10km SMF	V	V
At least 40km MMF		V

Table 3-1

Other objectives include:

- Support full-duplex operation only
- Preserve the 802.3 / Ethernet frame format utilizing the 802.3 media access controller (MAC)
- Preserve minimum and maximum frame size of current 802.3 standard
- Support a bit error rate (BER) better than or equal to 10-12 at the MAC/ physical layer service
- interface
- Provide appropriate support for optical transport network (OTN)
- Support a MAC data rate of 40 gigabit per second
- Support a MAC data rate of 100 gigabit per second

3.6 IEEE 802.3ba 40/100GbE Standardization Timeline

In July 2006, the IEEE 802.3 working group formed the High Speed Study Group (HSSG) to investigate new standards for high speed Ethernet.

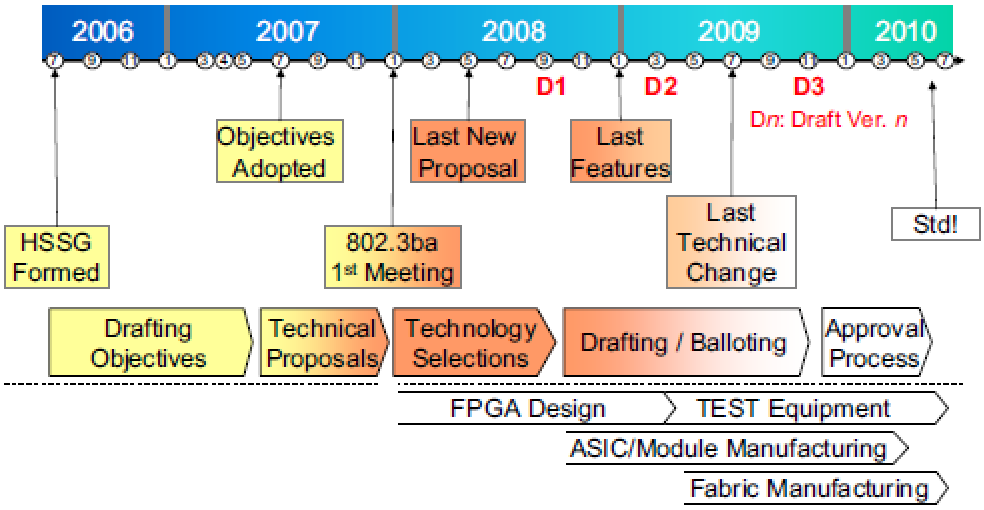


Figure 3-10

In June 2007, a trade group called "Road to 100G" was formed after the NXTcomm trade show in Chicago. Official standards work was started by IEEE 802.3 Higher Speed Study Group. In December 2007 a Project Authorization Request (PAR) was approved and the 802.3ba

Ethernet Task Force commenced on December 5, 2007 with the following project authorization request:

The purpose of this project is to extend the 802.3 protocol to operating speeds of 40 Gb/s and 100 Gb/s in order to provide a significant increase in bandwidth while maintaining maximum compatibility with the installed base of 802.3 interfaces, previous investment in research and development, and principles of network operation and management. The project is to provide for the interconnection of equipment satisfying the distance requirements of the intended applications.

January 2008 the HSSG was renamed and met as the "IEEE 40Gb/s and 100Gbs Ethernet Task Force," moving the process to the next stage of formalization. This standard was approved at the June 2010 IEEE Standards Board meeting under the name IEEE Std 802.3ba-2010.

Figure 3-10 shows the timeline of IEEE802.3ba.

3.7 40/100GbE PHY Architectures

The IEEE 802.3ba specifies a single architecture that accommodates 40 Gigabit Ethernet and 100 Gigabit Ethernet and all of the physical layer specifications under development.

The PHY device consists of a physical medium dependent (PMD) sublayer, a physical medium attachment (PMA) sublayer, and a physical coding sublayer (PCS). The backplane and copper cabling PHYs also include an auto-negotiation (AN) sublayer and a forward error correction (FEC) sublayer. Figure 4-1 is an overview of 40/100GbE PHY architectures, which we will cover in detail by following sections.

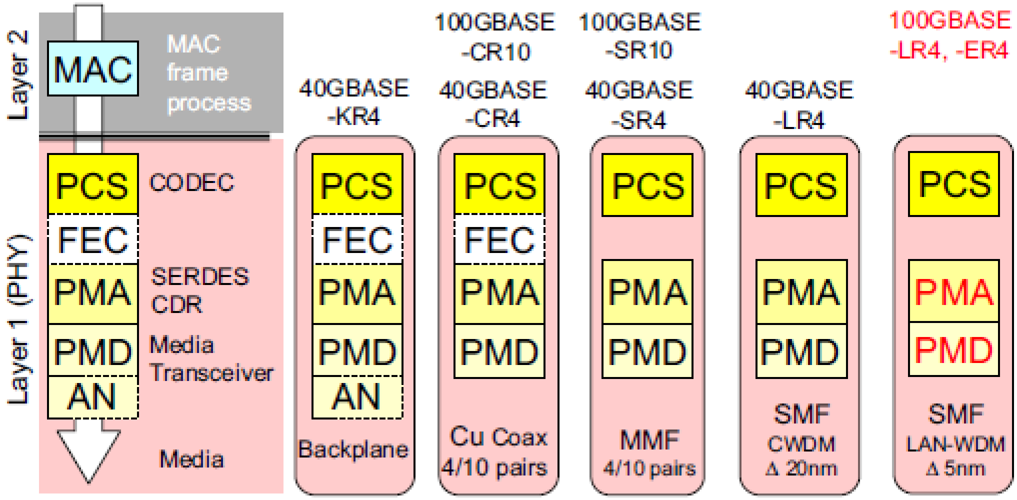


Figure 3-11, An Overview of 40/100GbE PHY Architectures

3.7.1 Physical Coding Sublayer (PCS)

As shown in Figure 3-11, the PCS translates between the respective media independent interface (MII) for each rate and the PMA sublayer. The PCS is responsible for the encoding of data bits into code groups for transmission via the PMA and the subsequent decoding of these code groups from the PMA. The Task Force developed a low overhead scheme, referred to as “Multilane Distribution (MLD),” as the basis for the PCS for 40 Gigabit Ethernet and 100 Gigabit Ethernet.

The MLD scheme in the PCS has been designed to support all PHY types for both 40 Gigabit Ethernet and 100 Gigabit Ethernet. It is flexible and scalable, and will be able to support all PHY types currently under development in the IEEE 802.3ba project. We are covering MLD in later sections. Furthermore, the PCS will support future PHY types that may be developed that will be fueled by continuous advances in electrical and optical transmission. The PCS layer also performs the following functions:

- Provide frame delineation
- Transportation of control signals
- Ensure necessary clock transition density as needed by the physical optical and electrical technology
- Stripe and re-assemble the information across multiple lanes

The PCS leverages the 64B/66B coding scheme that was used in 10 Gigabit Ethernet. It provides a number of useful properties including low overhead and sufficient code space to support necessary codewords, which are also consistent with 10 Gigabit Ethernet.

3.7.2 Multi-Lane Distribution (MLD)

The Multi-Lane Distribution (MLD) scheme implemented in the PCS is fundamentally based on a striping of the 66-bit blocks across multiple lanes.

MLD ensures that PCS can scale with technology:

- 100GbE: 10-lane CAUI, 10 or 4 (or 2 or 1)-lane PMD
- 40GbE: 4-lane XLAUI, 4 (or 2 or 1)-lane PMD

And it provides in-band de-skew mechanism:

- TX: Insert a lane marker in each PCS lane every 16,384 blocks
- RX: Alignment and skew compensation by searching the marker

The mapping of the lanes to the physical electrical and optical channels that will be used in any implementation is complicated by the fact that the two sets of interfaces are not necessarily coupled. Technology development in either chip interfaces or optical interface is not always tied together, and it was necessary to develop the concept of PCS lanes to allow the decoupling of the evolution of the optical interface widths from the evolution of the electrical interface widths. Figure 3-12 shows the concept of MLD.

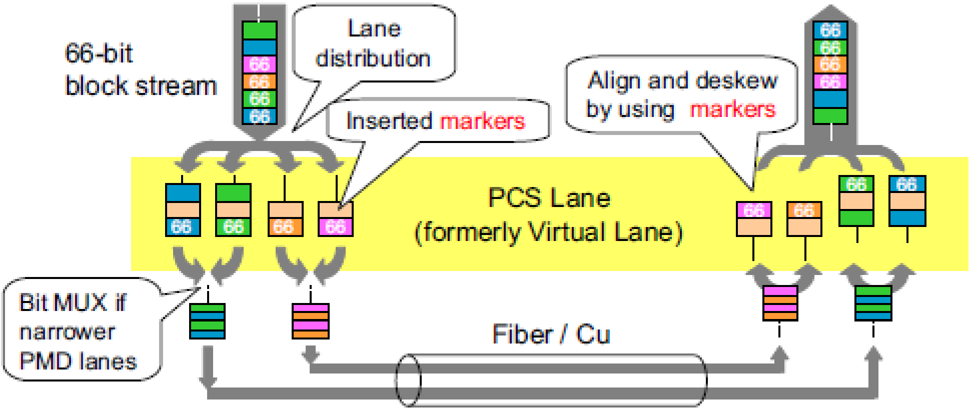

66-bit block stream
Lane distribution
Inserted markers
Align and deskew by using markers
PCS Lane (formerly Virtual Lane)
Bit MUX if narrower PMD lanes
Fiber / Cu
66

Figure 3-12

The transmit PCS, therefore performs the initial 64B/66B encoding and scrambling on the aggregate channel (40 gigabit per second or 100 gigabit per second) before distributing 66-bit block in a round robin basis across the PCS lanes, as illustrated in Figure 4-3.

The number of PCS lanes needed is the least common multiple of the expected widths of optical and electrical interfaces. For 100 Gigabit Ethernet 20 PCS lanes has been chosen. The number of electrical or optical interface widths supportable in this architecture is equivalent to the number of factors of the total PCS lanes. Therefore, 20 PCS lanes support interface widths of 1, 2, 4, 5, 10 and 20 channels or wavelengths. For 40 Gigabit Ethernet 4 PCS lanes support interface widths of 1, 2, and 4 channels or wavelengths.

Once the PCS lanes are created they can then be multiplexed into any of the supportable interface widths. Each PCS lane has a unique lane marker, which is periodically inserted. All multiplexing is done at the bit-level. The round-robin bit-level multiplexing can result in multiple PCS lanes being multiplexed into the same physical channel. The unique property of the PCS lanes is that no matter how they are multiplexed together, all bits from the same PCS lane follow the same physical path, regardless of the width of the physical interface. This enables the receiver to be able to correctly re-assemble the aggregate channel by first demultiplexing the bits to re-assemble the PCS lane and then re-align the PCS lanes to compensate for any skew. The unique lane marker also enables the deskew operation in the receiver. Bandwidth for these lane markers is created by periodically deleting inter-packet gaps (IPG). These alignment blocks are also shown in Figure 3-13.

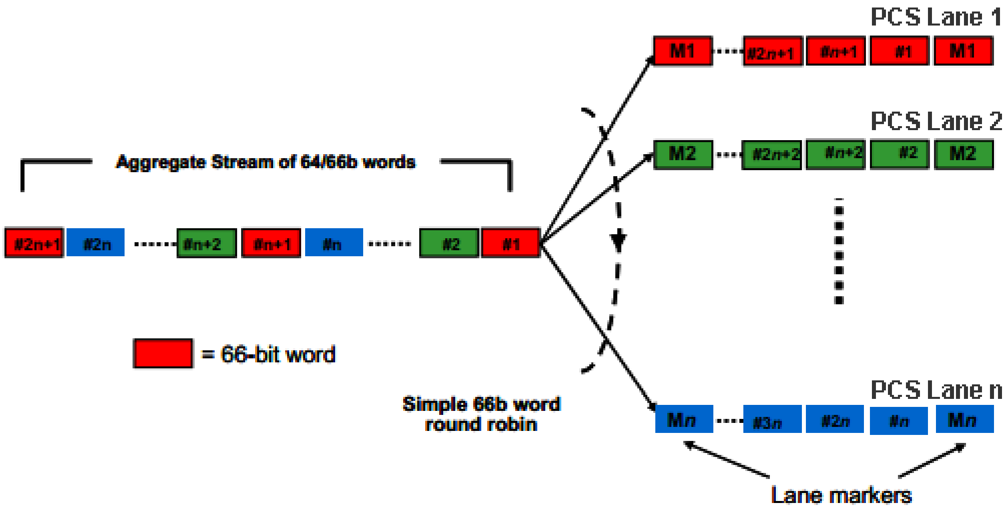


Figure 3-13 Round Robin Distribution Algorithm

The receiver PCS receives all these multiple PCS lanes, realigns them using the embedded lane markers and then re-order the lanes into their original order to reconstruct the aggregate signal.

Figure 3-14 and 3-15 illustrates the MLD of 40GbE and 100GbE separately.

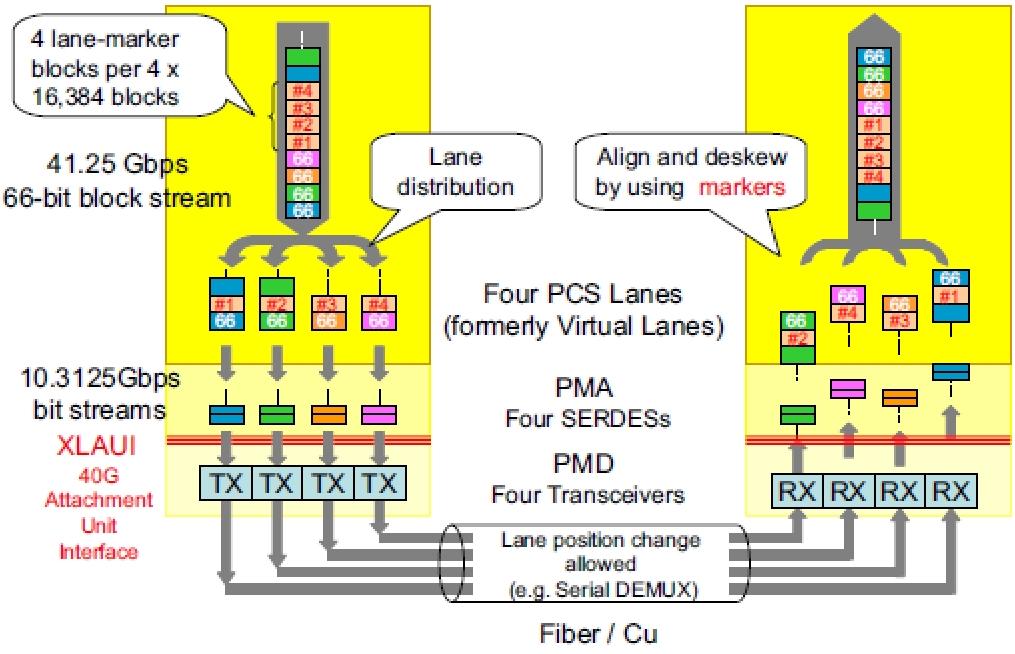

4 lane-marker blocks per 4 x 16,384 blocks
41.25 Gbps
66-bit block stream
Lane distribution
Align and deskew by using markers
Four PCS Lanes (formerly Virtual Lanes)
10.3125Gbps bit streams
XLAUI
40G Attachment Unit Interface
PMA
Four SERDESs
PMD
Four Transceivers
TX
RX
Lane position change allowed (e.g. Serial DEMUX)
Fiber / Cu

Figure 3-14 40 GbE MLD

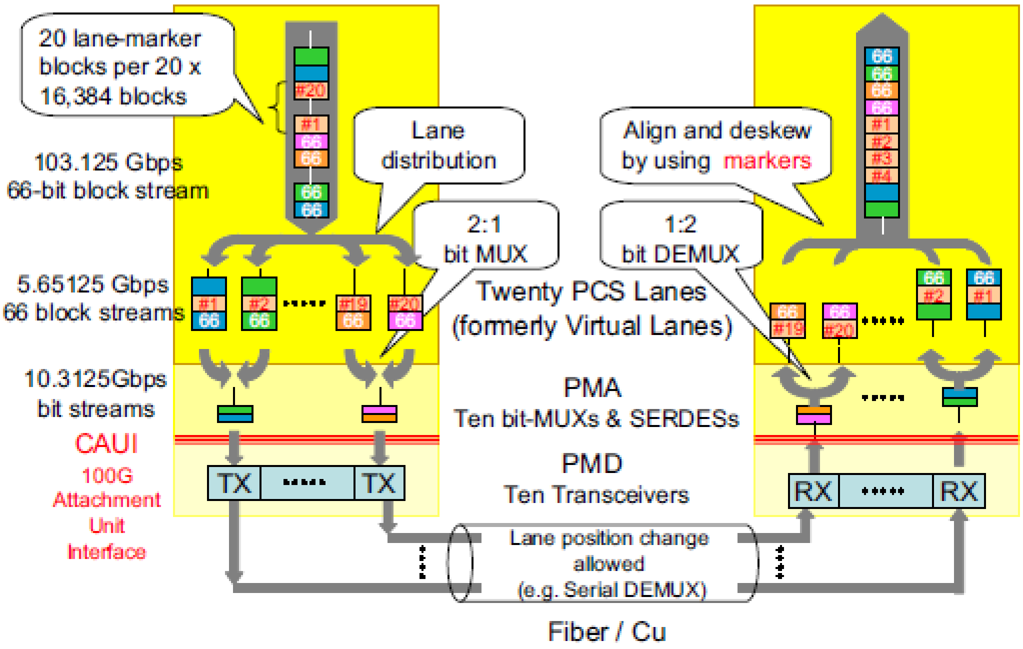

20 lane-marker blocks per 20 x 16,384 blocks
103.125 Gbps
66-bit block stream
Lane distribution
Align and deskew by using markers
2:1 bit MUX
1:2 bit DEMUX
5.65125 Gbps
66 block streams
Twenty PCS Lanes (formerly Virtual Lanes)
10.3125Gbps bit streams
CAUI
100G Attachment Unit Interface
PMA
Ten bit-MUXs & SERDESs
PMD
Ten Transceivers
TX
RX
Lane position change allowed (e.g. Serial DEMUX)
Fiber / Cu

Figure 3-15 100 GbE MLD

Two key advantages of the MLD methodology are that all the encoding, scrambling and deskew functions can all be implemented in a CMOS device, which is expected to reside on the host device, and minimal processing of the data bits other than bit muxing happens in the high speed electronics embedded with an optical module. This will simplify the functionality and ultimately lower the costs of these high-speed optical interfaces.

3.7.3 Physical Medium Attachment (PMA)

The PMA sublayer interconnects the PCS to the PMD sublayer, and contains the functions for transmission, reception, and (depending on the PHY) collision detection, clock recovery and skew alignment. Within this section, the description of the PMA sublayer will focus on the transmission, reception, and clock recovery aspects of the PMA function. The wide range of supportable interfaces and implementation options requires that, to fully explain the PMA function, it is necessary to explode the PMA function into some PMA sub-layers.

Figure 4-6 illustrates the general architecture for 100 Gigabit Ethernet, as well as examples of two other architectural implementations:

- 100GBASE-LR4, which is defined as 4 wavelength at 25 gigabit per second per wavelength on SMF
- 100GBASE-SR10, which is defined as 10 wavelengths across 10 parallel fiber paths at 10 gigabit per second on MMF

These two implementations will be used to illustrate the flexibility needed by the PMA sublayer to support the multiple PMDs being developed for 40 Gigabit Ethernet and 100 Gigabit Ethernet.

As described in the previous sections, for 100 Gigabit Ethernet the PCS creates 20 PCS lanes. In the example implementation shown in Figure 3-16, the PMA functionality is split between two PMA devices that are interconnected via an electrical interface, known as the 100 gigabit per second attachment unit interface (CAUI), which is based on a 10 wide interface at 10 gigabit per second per lane. In this implementation the PMA sublayer at the top of the CAUI multiplexes the 20 PCS lanes into 10 physical lanes. The PMA sublayer at the the bottom of the CAUI performs three functions. First, it retimes the incoming electrical signals. After the retiming the electrical lanes are then converted back to 20 PCS lanes, which are then multiplexed into the 4 lanes needed for the 100GBASE-LR PMD.

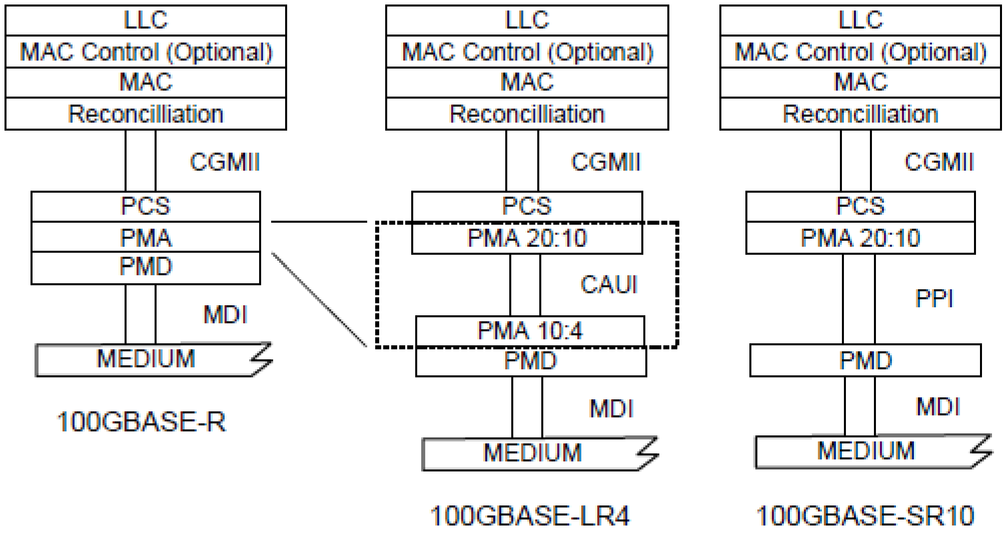


Figure 3-16 the Illustrations of 100GBASE-R Architectures

However the implementation of the 100GBASE-SR10 architecture is different. In this implementation a host chip is directly connected to an optical transceiver that is hooked up to 10 parallel fiber paths in each direction. The PMA sublayer resides in the same device as the PCS sublayer, and multiplexes the 20 PCS lanes into the ten electrical lanes of the Parallel Physical Interface (PPI), which is the non-retimed electrical interface that connects the PMA to the PMD.

In summary, the high level PMA functionality of multiplexing and clock recovery still exists but the actual implementation is dependent on the specific PMD being used.

3.7.4 Physical Media Dependent (PMD)

Different physical layer specifications for computing and network aggregation applications are being developed. For computing applications, physical layer solutions will cover distances inside the data center for up to 100m for a full range of server form factors including blade, rack, and pedestal configurations. For network aggregation applications, the physical layer solutions include distances and media appropriate for data center networking, as well as service provider inter-connection for intra-office and inter-office applications. A summary of the physical layer specifications being developed for each MAC rate is shown in Table 4-1.

	40 Gigabit Ethernet	100 Gigabit Ethernet
At least 1m backplane	40GBASE-KR4	
At least 10m copper cable	40GBASE-CR4	100GBASE-CR10
At least 100m OM3 MMF	40GBASE-SR4	100GBASE-SR10
At least 10km SMF	40GBASE-LR4	100GBASE-LR4
At least 40km MMF		100GBASE-ER4

Table 3-2. A summary of the 40/100GbE physical layer specifications

3.8 40/100GbE PHYs Added to IEEE 802.3ba

3.8.1 BASE-CR and 40BASE-KR4

The 40GBASE-KR4 PMD supports backplane transmission while the 40GBASE-CR4 and 100GBASE-CR10 PMD support transmission across copper cable assemblies. All three of the PHYs leverage the Backplane Ethernet 10GBASE-KR architecture, developed channel requirements and PMD.

The architecture for the PHY types is shown in Figure 3-17. All three PHYs use the standard 40GBASE-R and 100GBASE-R PCS and PMA sublayers. The BASE-CR and 40GBASE-KR4 PHYs also include an auto-negotiation (AN) sublayer and an optional FEC sublayer.

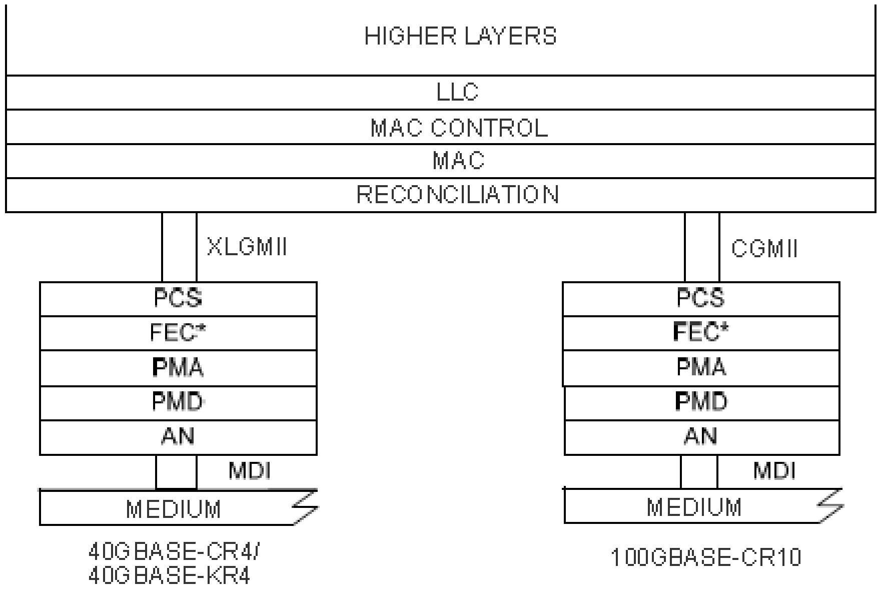


Figure 3-17 Backplane and Copper Cable Architecture

The BASE-CR and 40GBASE-KR4 specifications also leverage the channel development efforts of the Backplane Ethernet project. The channel specifications for 10GBASE-KR were developed to ensure robust transmission at 10 gigabit per second. The 40 Gigabit Ethernet and 100 Gigabit Ethernet PHYs apply these channel characteristics to 4 lane and 10 lane solutions. The BASE-CR specifications will also leverage the cable assembly specifications developed in support of 10GBASE-CX4. For 40GBASE-CR4, two connectors have been selected: The QSFP connector which will support a module footprint that can support either copper-based or Gigabit per second optic based modules. The 10GBASE-CX4 connector has also been selected, which will enable an upgrade path for those applications that are already invested in 10GBASE-CX4.

The effective data rate per lane is 10 gigabit per second, which when 64B/66B encoded results in a signaling rate of 10.3125 gigabaud per second.. Thus, the 40GBASE-KR4 and 40GASE-CR4 PMDs support transmission of 40 Gigabit Ethernet over 4 differential pair in each direction over either a backplane or twin axial copper cabling medium, while the 100GBASE-CR10 PMD will support the transmission of 100 Gigabit Ethernet over 10 differential pair in each direction for at least 10m over a twin axial copper cable assembly.

3.8.2 BASE-SR, BASE-LR, and BASE-ER

All of the optical PMDs being developed share the common architecture shown in Figure 3-18. While they share a common architecture, the PMA sublayer plays a key role in transmitting and receiving the number of PCS lanes from the PCS sublayer to the appropriate number of physical lanes that are required per the PMD sublayer and medium.

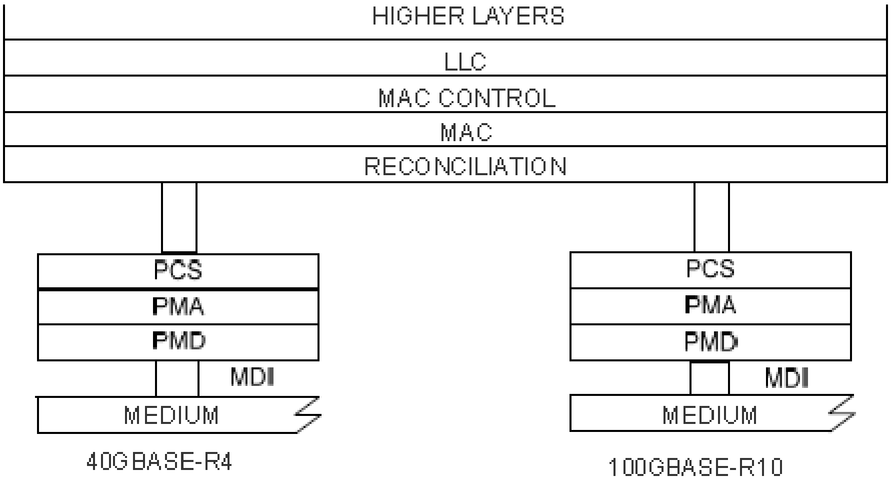


Figure 3-18 40GBASE-R and 100GBASE-SR Architecture

Below is a description of each of the different optical PMD's:

- 40GBASE-SR4 and 100GBASE-SR10 PMD - based on 850nm technology and supports transmission over at least 100m OM3 parallel gigabit per second. The effective date rate per lane is 10 gigabit per second, which when 64B/66B encoded results in a signaling rate of 10.3125 gigabaud per second... Therefore, the 40GBASE-SR4 supports transmission of 40 Gigabit Ethernet over a parallel gigabit per second medium consisting of 4 parallel OM3 fibers in each direction, while the 100GBASE-SR10 PMD will support the transmission of 100 Gigabit Ethernet over a parallel gigabit per second medium consisting of 10 parallel OM3 fibers in each direction.

- 40GBASE-LR4 - based on 1310nm, Coarse Wave Division Multiplexing (CWDM) technology and supports transmission over at least 10km over SMF. .The grid is based on the ITU G.694.2 specification, and the wavelengths used are 1270, 1290, 1310, and 1330nm. The effective data rate per lambda is 10 gigabit per second, which when 64B/66B encoded results in a signaling rate of 10.3125 gigabaud per second. which will help provide maximum re-use of existing 10G PMD technology. Therefore, the 40GBASE-LR4 PMD supports transmission of 40 Gigabit Ethernet over 4 wavelengths on each SMF in each direction.
- 100GBASE-LR4 - based on 1310nm, Dense Wave Division Multiplexing (WDM) technology and supports transmission over at least 10km over single mode gigabit per second. The grid is based on the ITU G.694.1 specification, and the wavelengths used are 1295, 1300, 1305, and 1310nm. The effective data rate per lambda is 25 gigabit per second, which when 64B/66B encoded results in a signaling rate of 28.78125 gigabaud per second. Therefore, the 100GBASE-LR4 PMD supports transmission of 100 Gigabit Ethernet over 4 wavelengths on each SMF in each direction.

100GBASE-ER4 - based on 1310nm, WDM technology and supports transmission over at least 40km over single mode gigabit per second. The grid is based on the ITU G.694.1 specification, and the wavelengths used are 1295, 1300, 1305, and 1310nm. The effective data rate per lambda is 25 gigabit per second, which when 64B/66B encoded results in a signaling rate of 28.78125 gigabaud per second. Therefore, the 100GBASE-LR4 PMD supports transmission of 100 Gigabit Ethernet over 4 wavelengths on each SMF in each direction. To achieve the 40km reaches, it is anticipated that implementations will include semiconductor optical amplifier (SOA) technology.

3.9 40/100GbE Interfaces

The various chip interfaces in the IEEE 802.3ba amendment are illustrated in Figure 4. The IEEE 802.3ba amendment will specify some interfaces as logical, intra-chip, interfaces, as opposed to a physical, interchip, interfaces as they have been in the past. A logical interface specification only specifies the signals and their behavior. A physical interface specification also specifies the electrical and timing parameters of the signals.

The inclusion of logical interfaces supports system on a chip (SoC) implementations where various cores, implementing the different sublayers, are supplied by different vendors. The provision of an open interface specification through the IEEE 802.3ba amendment will help these cores to be integrated into a SoC in the same way that chips from different vendors can be integrated to build a system. While a physical interface specification is sufficient to specify a logical interface, there are cases where the interfaces are unlikely to ever be implemented as a physical interface so the provision of electrical and timing parameters are unnecessary.

There are three chip interfaces defined that have a common architecture for both speeds. The MII is a logical interface that connects the MAC to a PHY and the AUI is a physical interface that extends the connection between the PCS and the PMA. The naming of these interfaces follows the convention found in 10 Gigabit Ethernet, IEEE Std 802.3ae, where the 'X' in XAUI and XGMII represents the Roman numeral 10. Since the Roman numerals for 40 are 'XL' and the Roman numeral for 100 is 'C', the same convention yields XLAUI and XGMII for 40 gigabit per second and CAUI and CGMII for 100 gigabit per second. The final interface is the Parallel Physical Interface (PPI), discussed in further detail below, which is the physical interface for the connection between the PMA and the PMD for 40GBASE-SR4 and 100GBASESR10 PMDs.

3.9.1 40 and 100 Gigabit Media Independent Interface (XLGMII and CGMII)

The XLGMII, which supports the 40 gigabit per second data rate, and the CGMII, which supports the 100 gigabit per second data rate, are defined as logical interfaces between the MAC and the PCS which share a common interface specification, the only differentiation being the specified clock rate.

The interface provides 64 bit wide transmit and receive data paths. These 64 bit data paths are grouped nto 8 lanes of 8 bits, with a control bit associated with each lane indicating if it is data or control information such as. delimiters or idle being transferred during that clock cycle. There is a single clock associated with transmit and a single clock associated with the receive path. These clocks operate at one one-sixty-fourth of the data rate resulting in a 625 megahertz clock for 40 gigabit per second operation and a 1.5625 gigahertz clock for 100 gigabit per second operation.

Since this is a wide high speed interface it is not expected to be physically implemented as a physical, inter-chip, interface so is only specified as a logical, intra-chip, interface.

3.9.2 40 and 100 Gigabit Attachment Unit Interface (XLAUI and CAUI)

The XLAUI, which supports the 40 gigabit per second data rate, and CAUI, which supports the 100 gigabit per second data rate, are low pin count physical interfaces that enables partitioning between the MAC and sublayers associated with the PHY in a similar way to XAUI in 10 Gigabit Ethernet. They are self-clocked, multi-lane, serial links utilizing 64B/66B encoding. Each lane operates at an effective data rate of 10 gigabit per second, which when 64B/66B encoded results in a signaling rate of 10.3125 gigabaud per second.

The lanes utilize low-swing AC-coupled balanced differential signaling to support a distance of approximately 25 cm. In the case of XLAUI, there are four transmit and four receive lanes of 10 gigabit per second, resulting in a total of 8 pairs or 16 signals, in the case of CAUI there are ten transmit lanes and ten receive lanes of 10 gigabit per second resulting in a total of 20 pairs or 40 signals.

These interfaces serve primarily as chip to chip interfaces, for example to partition system design between the largely digital based system chip and more analogue based portions of the PHY chip which are often based on different technology. In addition, while there is no mechanical connector specified for XLAUI and CAUI in the IEEE 802.3ba amendment, these interfaces are also candidate interfaces for pluggable form factor specifications, enabling a single host system to support the various PHY types through pluggable modules. Due to this, these interfaces are being specified based on a channel of approximately 25 cm on FR4 printed circuit boards (PCB) strip line with one connector.

The pluggable form factor specifications themselves are beyond the scope of IEEE 802.3 and are developed by other industry organizations.

3.9.3 Parallel Physical Interface (PPI)

The PPI is a physical interface for short distances between the PMA and PMD sub-layers. It is common to both 40 Gigabit Ethernet and 100 Gigabit Ethernet, with the only differentiation being the number of lanes. The PPI is a self-clocked, multilane, serial links, utilizing 64B/66B encoding. Each lane operates at an effective data rate of 10 gigabit per second, which when 64B/66B encoded results in a signaling rate of 10.3125 gigabaud per second. In the case of the 40 Gigabit Ethernet, there are four transmit and four receive lanes of 10 gigabit per second, in the case of the 100 Gigabit Ethernet there are ten transmit lanes and ten receive lanes of 10 gigabit per second.

3.10 Our Solution to This Problem

We like to propose a different FEC method to lower the BER than the one specified in 802.3ba, which is binary burst error correction code (2112, 2080), a type of cyclic code. We will select from Hamming code, RS or Viterbi.

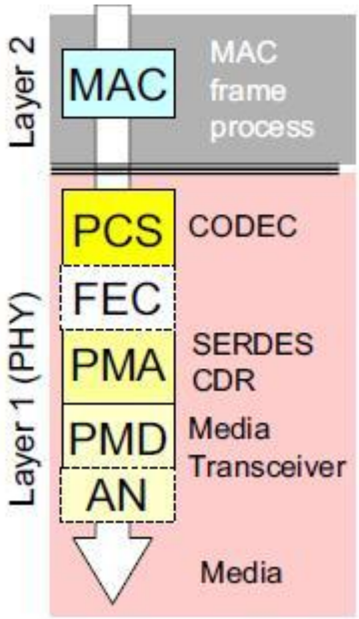


Figure 3-19

3.11 Where Our Solution is Different from Others

The coding scheme used in FEC is a type of cyclic code. We like to try some different cyclic code such as RS or even some block code such as Hamming code.

4 HYPOTHESIS (GOAL)

There exists some type of FEC methods that can lower BER at no less transmit speed and more expensive hardware cost than the one in 802.3ba.

5 METHODOLOGY

5.1 How to generate input data

There are two source of input data, each generated using pseudo-random generator.

- Data passed down from PCS, the layer above
- Crosstalk simulation

5.2 How to solve the problem

- Research the different FEC code, study the advantages and disadvantages, select the best one and find and study the algorithm, implements as encoder and decoder
- Java and/or C
- IDE to develop the simulation and charting program to compare the results

5.3 How to Generate Output

- the output from PCS layer simulator goes through 2 paths, one directly to BER monitor and the other goes to FEC encoder, backplane model, joined with crosstalk signal, to receiver, to decoder, then to BER monitor.

5.4 How to Test against Hypothesis

Use different FEC code algorithm in the encoder and decoder and compare the results in BER monitor.

6 IMPLEMENTATION

We created two models, one for simulating FEC using RS coding, one for Binary Cyclic coding. Each model accepts a parameter EbNo. We use Bertool to run the model multiple times, each time passing in EbNo ranging from 1 to 20 dB. As result, each model simulation will produce two series BER corresponding to the EbNo passed, one with coding and one without.

In actual 802.3ap FEC layer, the (2112, 2080) burst error correction code is used. It is constructed by shortening of cyclic code (42987, 42955). In our model, any codeword length longer than 31 will freeze Matlab, so we used binary cyclic code (7,4) instead.

6.1 Model Setup

6.1.1 Binary Cyclic Code (7, 4)

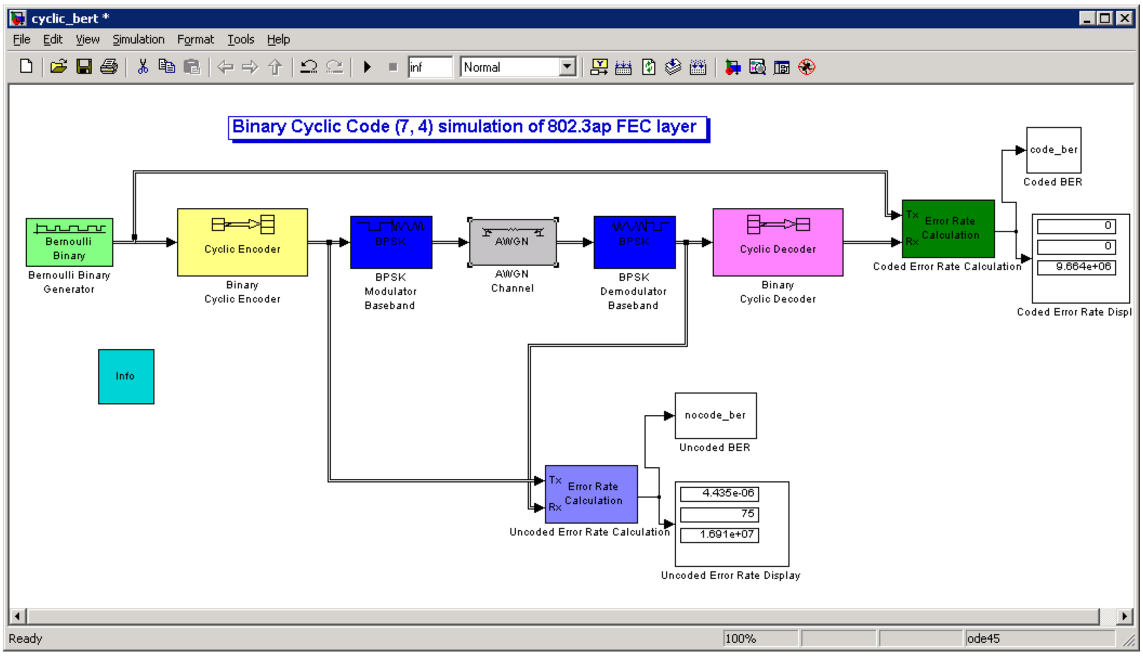
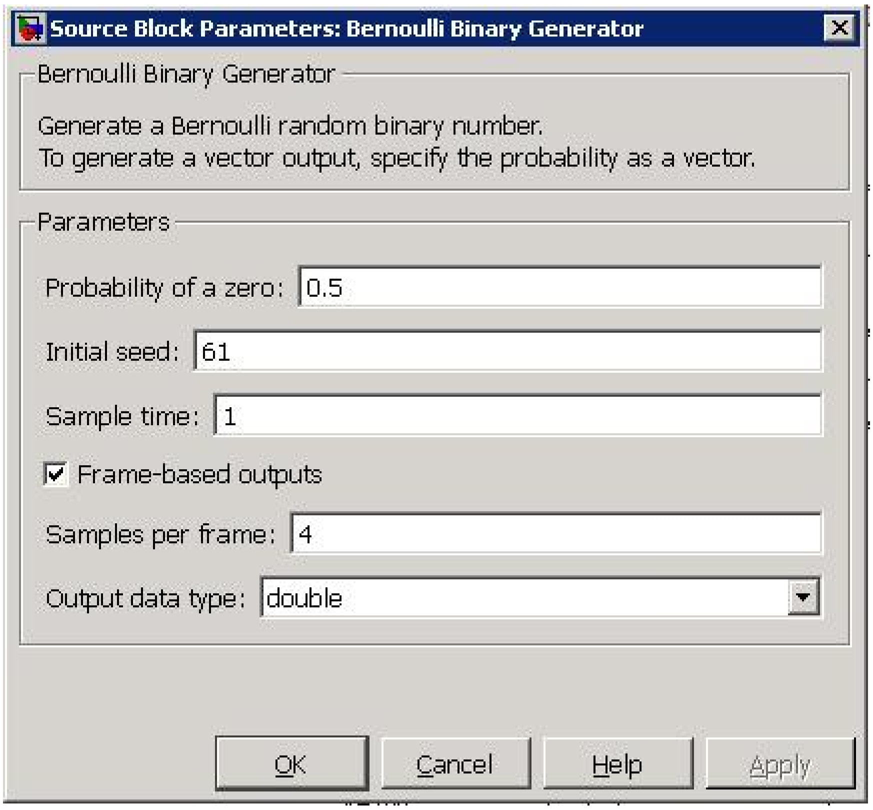


Figure 6-1

We use the Bernoulli Binary Generator block to simulate signal source. It generates random binary number using a Bernoulli distribution. The mean value is 0.5 and variance is 0.25.



Source Block Parameters: Bernoulli Binary Generator

Bernoulli Binary Generator

Generate a Bernoulli random binary number.
To generate a vector output, specify the probability as a vector.

Parameters

Probability of a zero: 0.5

Initial seed: 61

Sample time: 1

☑ Frame-based outputs

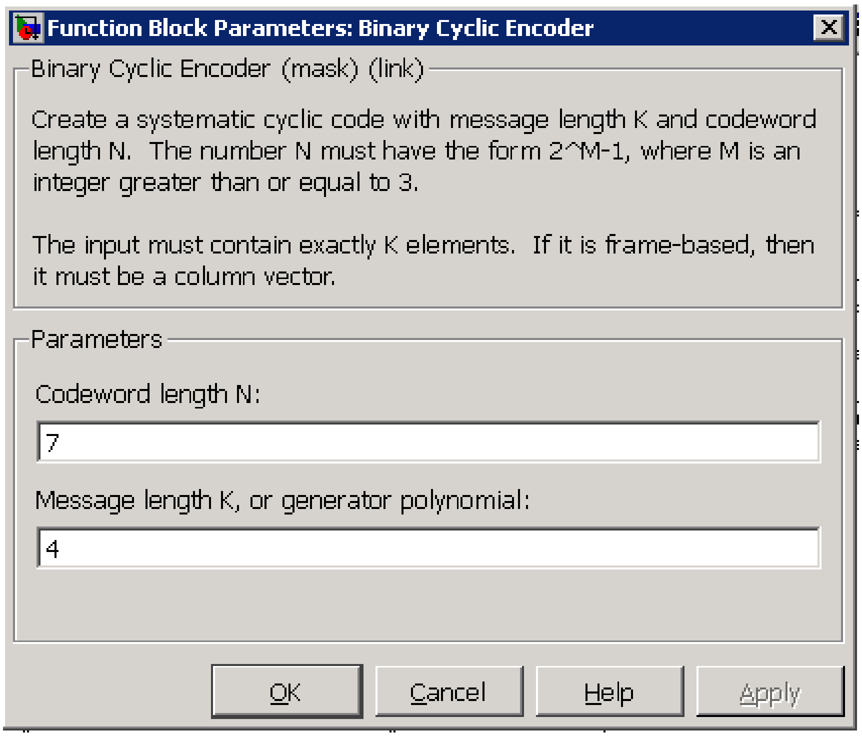
Samples per frame: 4

Output data type: double

Figure 6-2

Frame-based outputs are checked and Samples per frame is set to 4 so that the messages match the input requirement of our Binary Cyclic Encoder Block.

The codeword Length N is set to 7, same to the decoder block.



Function Block Parameters: Binary Cyclic Encoder

Binary Cyclic Encoder (mask) (link)

Create a systematic cyclic code with message length K and codeword length N. The number N must have the form 2^M-1, where M is an integer greater than or equal to 3.

The input must contain exactly K elements. If it is frame-based, then it must be a column vector.

Parameters

Codeword length N:

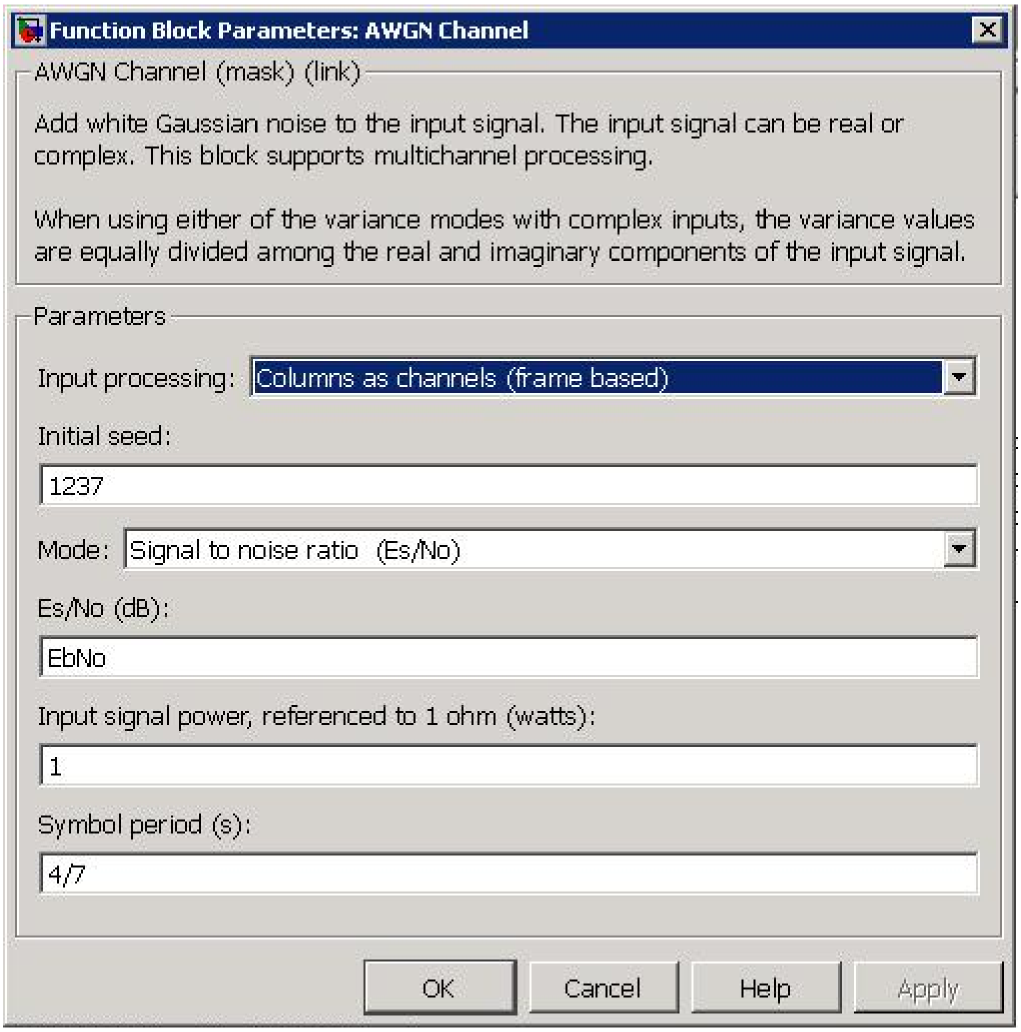
7

Message length K, or generator polynomial:

4

Figure 6-3

In AWGN Channel block, Symbol period is set to 4/7. This is so that the channel produces the same amount of noise per symbol as in the BPSK model without coding. And this allows us to evaluate the improvement in bit error rate due to channel coding. Es/No (dB): is set to variable EbNo so that Bertool will automatically run the simulation multiple times with different value for EbNo to chart the BER vs EbNo plot.



Function Block Parameters: AWGN Channel

AWGN Channel (mask) (link)

Add white Gaussian noise to the input signal. The input signal can be real or complex. This block supports multichannel processing.

When using either of the variance modes with complex inputs, the variance values are equally divided among the real and imaginary components of the input signal.

Parameters

Input processing: Columns as channels (frame based)

Initial seed: 1237

Mode: Signal to noise ratio (Es/No)

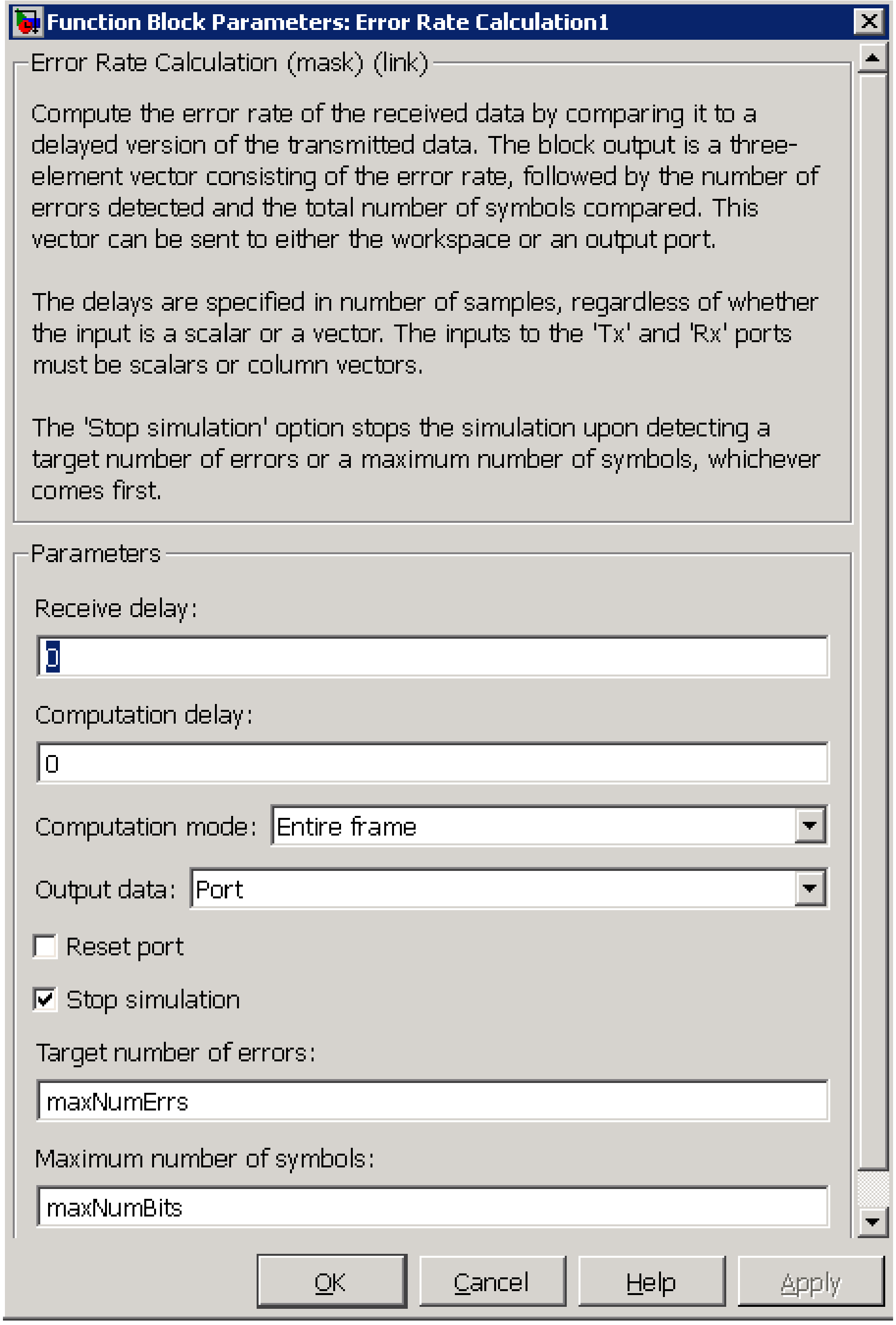
Es/No (dB): EbNo

Input signal power, referenced to 1 ohm (watts): 1

Symbol period (s): 4/7

Figure 6-4

In the error rate calculation block computation mode is set to Entire frame, Output data is to port so that the data is handled in other block, to a workspace block. The simulation is stopped when either of Target number of error or Maximum number of symbols reached the specified variables.



Error Rate Calculation (mask) (link)

Compute the error rate of the received data by comparing it to a delayed version of the transmitted data. The block output is a three-element vector consisting of the error rate, followed by the number of errors detected and the total number of symbols compared. This vector can be sent to either the workspace or an output port.

The delays are specified in number of samples, regardless of whether the input is a scalar or a vector. The inputs to the 'Tx' and 'Rx' ports must be scalars or column vectors.

The 'Stop simulation' option stops the simulation upon detecting a target number of errors or a maximum number of symbols, whichever comes first.

Parameters

Receive delay: 0

Computation delay: 0

Computation mode: Entire frame

Output data: Port

□ Reset port

☑ Stop simulation

Target number of errors: maxNumErrs

Maximum number of symbols: maxNumBits

Figure 6-5

There are two error rate calculation blocks, one for coded error rate and one is for without.

6.1.2 RS code

Here is screenshot of the model setup for RS code simulation in Matlab:

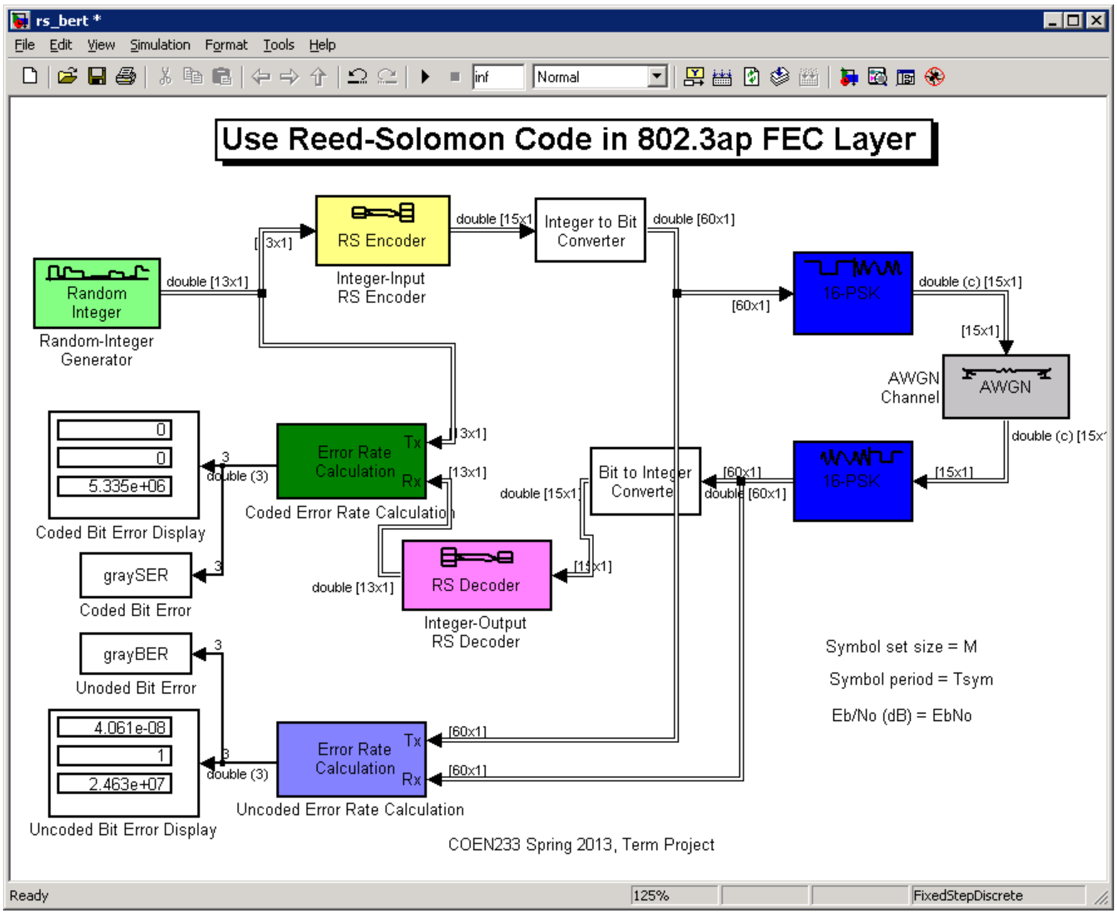
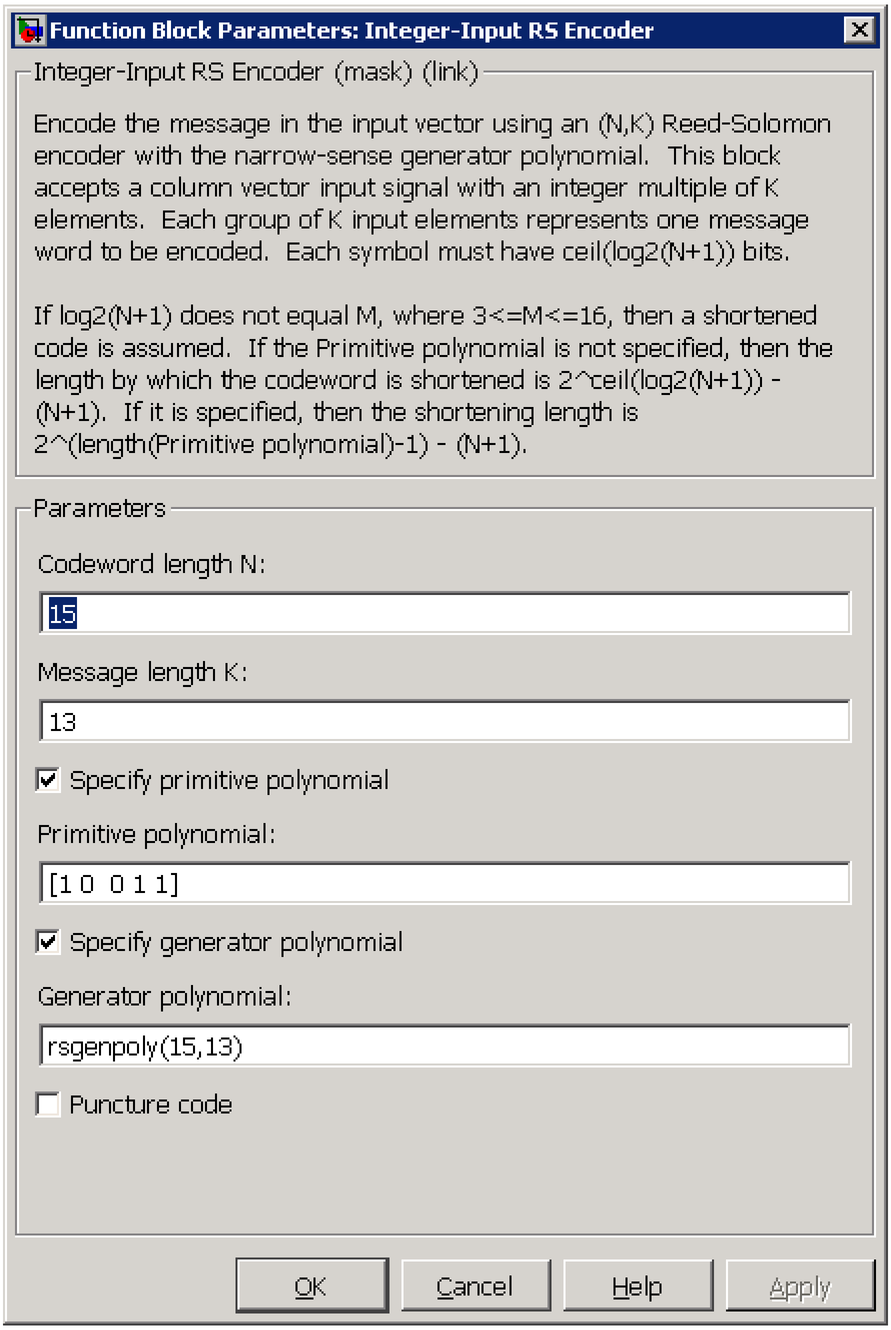


Figure 6-6

The parameter we used for RS encoder block:



Function Block Parameters: Integer-Input RS Encoder

Integer-Input RS Encoder (mask) (link)

Encode the message in the input vector using an (N,K) Reed-Solomon encoder with the narrow-sense generator polynomial. This block accepts a column vector input signal with an integer multiple of K elements. Each group of K input elements represents one message word to be encoded. Each symbol must have ceil(log2(N+1)) bits.

If log2(N+1) does not equal M, where 3<=M<=16, then a shortened code is assumed. If the Primitive polynomial is not specified, then the length by which the codeword is shortened is 2^ceil(log2(N+1)) - (N+1). If it is specified, then the shortening length is 2^(length(Primitive polynomial)-1) - (N+1).

Parameters

Codeword length N: 15

Message length K: 13

☑ Specify primitive polynomial

Primitive polynomial: [1 0 0 1 1]

☑ Specify generator polynomial

Generator polynomial: rsgenpoly(15,13)

☐ Puncture code

Figure 6-7

6.2 Design Document

The The simulation is done in Simulink© by building the model with blocks from toolbox for easy visualization. the model is also done in Matlab's scripting language. the simulation consists of a transmitter, a reciver, and a channel. we use a source block to generate a long sequence of random bits or integers (for RS code, for the reason that will be explained in the later part of the paper). The transmitter modulates these bits onto some form of digital signaling, which we will send through a simulated channel. We simulate the channel by adding a controlled amount of noise to the transmitted signal. This noisy signal then becomes the input to the receiver. The receiver demodulates the signal, producing a sequence of recovered bits. Finally, we compare the received bits to the transmitted bits, and tally up the errors.

Since this simulation try to simulate ethernet backplane channel, AWGN channel model is used here. It produces white noise with a constant spectral density (expressed as watts per hertz of bandwidth) and a Gaussian distribution of amplitude. so it is used to simulate background noise of the channel under study here.

For RS code simulation, the RS encoder and decoder block support use of m-bit symbols instead of bits. A message for an [n,k] Reed-Solomon code must be a k-column Galois array in the field GF(2^m). Each array entry must be an integer between 0 and 2^m-1. The code corresponding to that message is an n-column Galois array in GF(2^m). The codeword length n must be between 3 and 2^m-1. the RS coding we have implemented use RS(15,13)

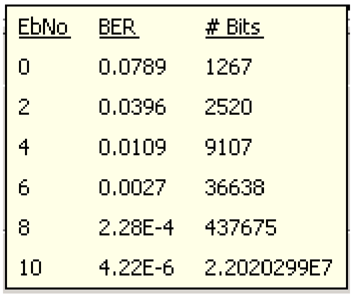
For Binary Cyclic code, the codeword length N must have the form 2^M-1, where M is an integer greater than or equal to 3. We are using (7,3).

7 DATA ANALYSIS AND DISCUSSION

7.1 Output Generation

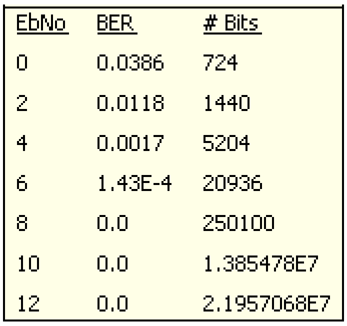
Simulation 1: Binary Cyclic Code

Uncoded



EbNo	BER	# Bits
0	0.0789	1267
2	0.0396	2520
4	0.0109	9107
6	0.0027	36638
8	2.28E-4	437675
10	4.22E-6	2.2020299E7

Coded



EbNo	BER	# Bits
0	0.0386	724
2	0.0118	1440
4	0.0017	5204
6	1.43E-4	20936
8	0.0	250100
10	0.0	1.385478E7
12	0.0	2.1957068E7

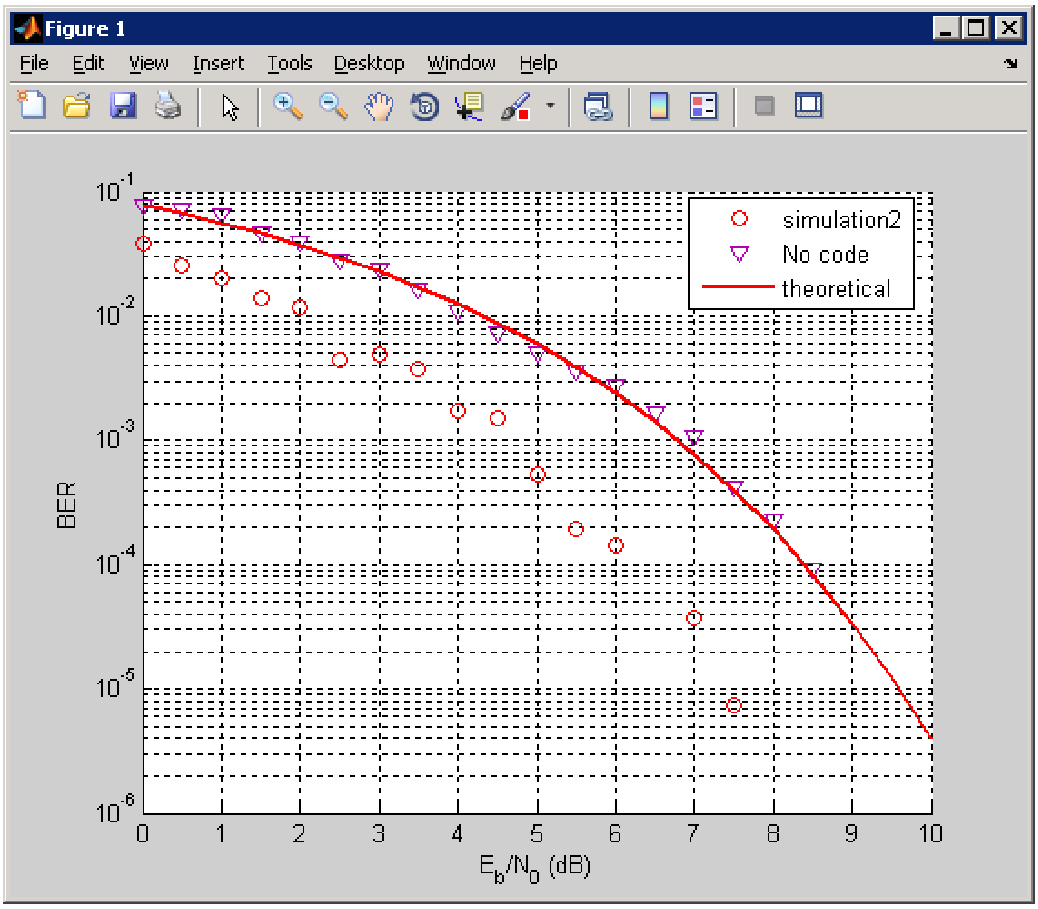
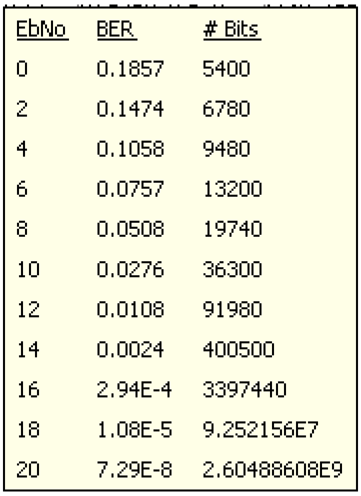


Figure 7-1 BER vs. EbN0 over AWGN channel for BPSK modulation scheme with and without Binary Cyclic

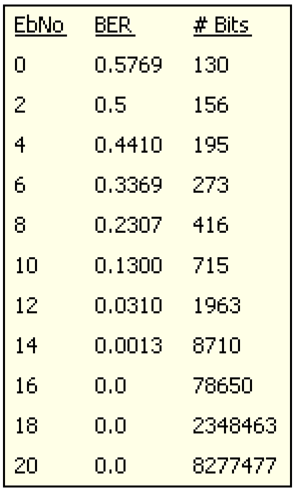
Simulation 2: RS code

Uncoded:



EbNo	BER	# Bits
0	0.1857	5400
2	0.1474	6780
4	0.1058	9480
6	0.0757	13200
8	0.0508	19740
10	0.0276	36300
12	0.0108	91980
14	0.0024	400500
16	2.94E-4	3397440
18	1.08E-5	9.252156E7
20	7.29E-8	2.60488608E9

Coded:



EbNo	BER	# Bits
0	0.5769	130
2	0.5	156
4	0.4410	195
6	0.3369	273
8	0.2307	416
10	0.1300	715
12	0.0310	1963
14	0.0013	8710
16	0.0	78650
18	0.0	2348463
20	0.0	8277477

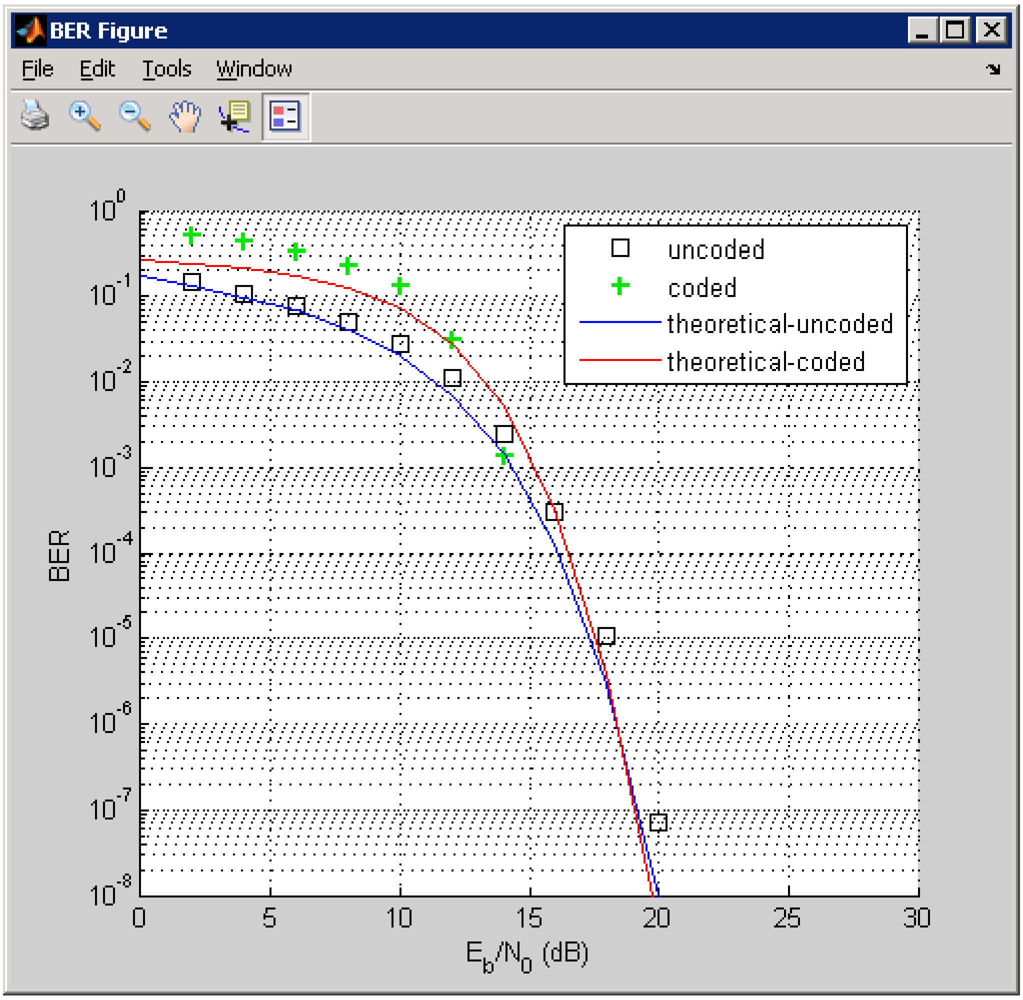


Figure 7-2 BER vs. EbN0 over AWGN channel for 16-MPSK modulation scheme with and without RS coding

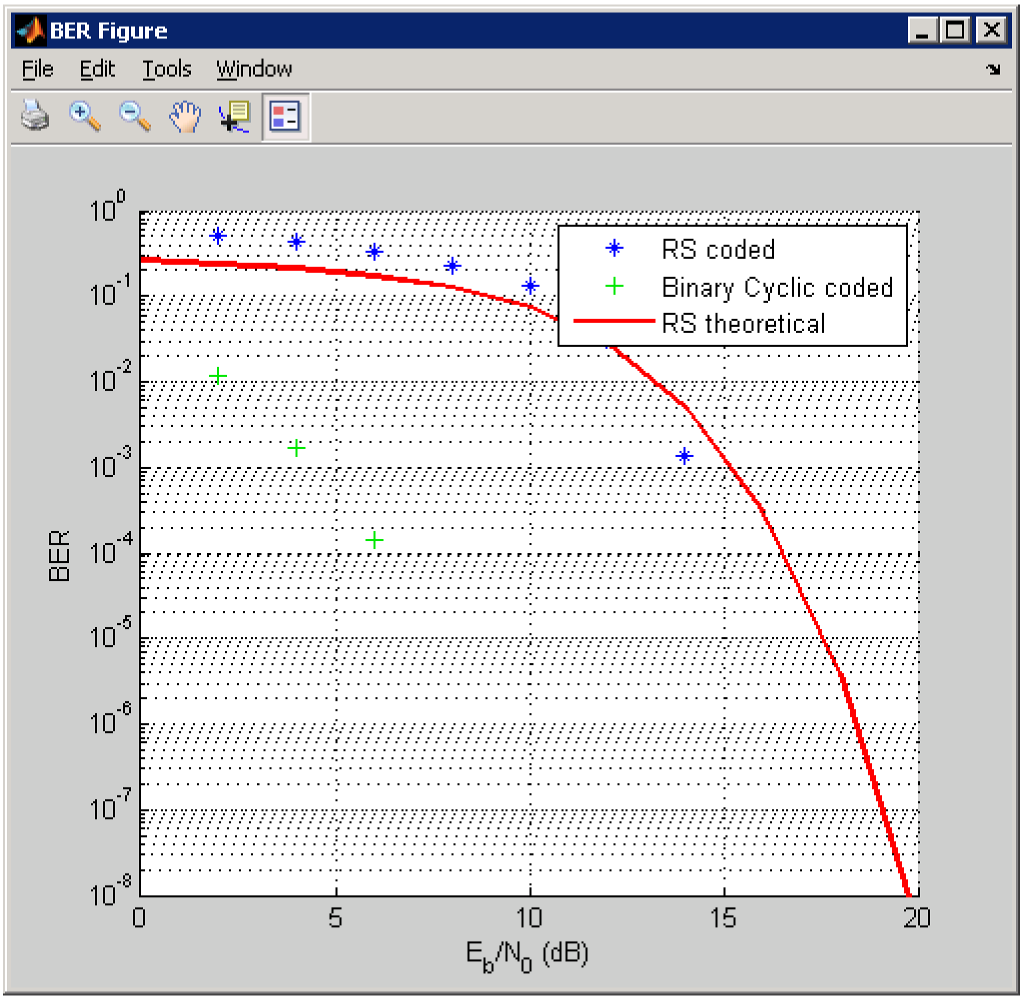


Figure 7-3 BER vs. EbN0 of RS coded and Binary Cyclic coded

7.2 Output Analysis

Based on data generated by computer simulation for BER calculation we obtained the following results:

1. BER vs. EbN0 over AWGN channel for BPSK modulation scheme with and without Binary Cyclic coding (Figure 7-1)

The coded BER, represented by the red circles, has lower BER than the uncoded curve throughout the EbN0 range. The goal of 802.3ap's FEC BER is 10-12 but since we were unable to acquire data at that high EbN0, we see at 10-5 around 2 dB coding gain was achieved.

2. BER vs EbN0 over AWGN channel for 16-MPSK modulation scheme with and without RS coding (Figure 7-2)

The coded signal, represented by the green plus, has sharper downward slope than that of uncoded signal, represented by the black squires. Before 13dB, uncoded signal has better BER, after that, coded signal has better BER. Again due to the limited measured data, it is hard to tell the coding gain but if the curve is extrapolated, the coding gain is about in 2dB range

3. In Figure 7-3 we simply plotted BER vs. EbN0 of RS coded and Binary Cyclic coded graph. And it is obvious Binary Cyclic coding has much better BER than RS coding.

Note: Theoretical coded and uncoded curves are included for each comparison except there is no theoretical Binary Cyclic coded curve because we couldn't come out a formula for it and Matlab doesn't have a build-in formula for that. Otherwise our measured data closely matches that of theoretical curves.

7.3 Compare Output against Hypothesis

Originally we made hypothesis that there are coding scheme that has better BER than the one actually used in 802.3ap FEC layer, and we picked RS coding as candidate. The result of the simulation shows RS coding has much lower BER than Binary Cyclic coding that's used in actual FEC layer.

7.4 Statistic Regression

When When the bit-error-rate is high, many bits will be in error. The worst-case bit-error-rate is 50 percent, at which point, it is essentially useless. The goal of 802.3ap standard for the error rate of FEC layer is to be lower than 10-12. And the theoretical SNR of the actual FEC code used for that error rate is around 15dB. We want to plot a curve of the bit-error-rate as a function of the SNR, and include enough points to cover a wide range of bit-error-rates. At such high SNRs, this becomes difficult, since the bit-error-rate becomes very low. For example, a bit-error-rate of 10−6 means only one bit out of every million bits will be in error. If our test signal only contains under million bits, we will most likely not see an error at this bit-error-rate. In order to be statistically significant, each simulation we run must generate some number of errors. If a simulation generates no errors, it does not mean the bit-error-rate is zero; it only means we did not have enough bits in our transmitted signal. As a rule of thumb, we need about 100 (or more) errors in each simulation, in order to have confidence that our bit-error-rate is statistically valid. That is why we set maxErrors variable to be 100 for the simulation to stop. But at 20dB EbNo for RS coded simulation and at 10dB EbNo for Cyclic coded simulation, billions of bits had been processed (been hours running) and there were no errors. Much longer time is needed to run the simulation to get enough errors to get the BER beyond 20dB and 10 dB respectfully, which is the time we didn't have.

7.5 Discussion

The simulation was built and run using Matlab 2012a/b on PC at design center (dcts2/dcts3). The specification for the PCs is:

Intel Xeon 5160@3.00GHz

16GB of RAM

Windows Server 2003 R2 Enterprise Edition

The assumption we made using the model:

FEC block shifts due to lost synchronization

Encoder and decoder latency

The only impairment is white noise.

8 CONCLUSIONS AND RECOMMENDATIONS

8.1 Summary and Conclusions

Both RS and Binary Cyclic code are widely used in digital communication system for detecting and correcting errors in received signal message bits. In this paper, the BER performances are obtained from the simulation of RS coded and Binary Cyclic coded transmission systems in the AWGN channels. It shows that the improvement of BER using both coding is better than that without coding, and also compared the performance of the two different coding, that is Binary Cyclic code has much better BER than that of RS code.

9 BIBLIOGRAPHY

IEEE 802.3ba Standard Specification

http://en.wikipedia.org/wiki/Media_access_control

http://en.wikipedia.org/wiki/PHY

http://en.wikipedia.org/wiki/Media_Independent_Interface

http://en.wikipedia.org/wiki/Physical_Coding_Sublayer

http://en.wikipedia.org/wiki/Physical_Medium_Dependent

http://standards.ieee.org/news/2010/ratification8023ba.html

10 APPENDICES

Program Source Code

Represent words for RS codes:

```
n = 7; k = 3; % Codeword length and message length
m = 3; % Number of bits in each symbol
x = 0:3; % for creating Galois field array

a = gf(x,m)

    msg = a; % Message is a Galois array.
    obj = comm.RSEncoder(n, k);

c1 = step(obj, msg(1,:)');
c2 = step(obj, msg(2,:)');
c = [c1 c2].

r = rsgenpoly(15,13) % for RS model

r= cyclpoly(15,5) % for cyclic code model


EbN0dB = 0:9;

% Loop over the vector of EbNo values.
berVec = zeros(3,numEbNos); % Reset
for jj = 1:numEbNos
  EbNo = EbNovec(jj);
  snr = EbNo; % Because of binary modulation
  reset(hErrorCalc)
  hChan.SNR = snr; % Assign Channel SNR
  % Simulate until numerrmin errors occur.
  while (berVec(2,jj) < numerrmin)
    msg = randi([0,M-1], siglen, 1); % Generate message sequence.
    txsig = step(hMod, msg); % Modulate.
    hChan.SignalPower = (txsig'*txsig)/length(txsig);   % Calculate and
    % assign signal power
    rxsig = step(hChan,txsig); % Add noise.
    decodmsg = step(hDemod, rxsig); % Demodulate.
    if (berVec(2,jj)==0)
      % The first symbol of a differentially encoded transmission
      % is discarded.
      berVec(:,jj) = step(hErrorCalc, msg(2:end),decodmsg(2:end));
    else
      berVec(:,jj) = step(hErrorCalc, msg, decodmsg);
    end
  end
  % Error rate and 98% confidence interval for this EbNo value
  [ber(jj), intv1] = berconfint(berVec(2,jj),berVec(3,jj)-1,.98);
  intv{jj} = intv1; % Store in cell array for later use.
  disp(['EbNo = ' num2str(EbNo) ' dB, ' num2str(berVec(2,jj)) ...
    ' errors, BER = ' num2str(ber(jj))])
end
```

Sample code for plotting the BER results if Bertool is not used

```
% Use BERFIT to plot the best fitted curve,
% interpolating to get a smooth plot.
fitEbNo = EbNomin:0.25:EbNomax; % Interpolation values
berfit(EbNovec,ber,fitEbNo,[],'exp');

% Also plot confidence intervals.
hold on;
for jj=1:numEbNos
 semilogy([EbNovec(jj) EbNovec(jj)],intv{jj},'g-+');
end
hold off;
```