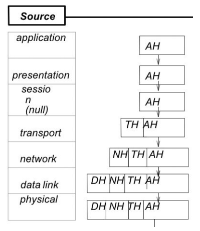
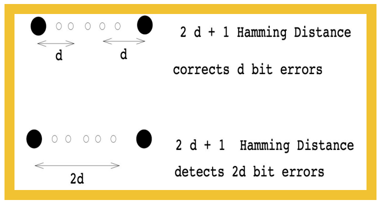
Layering
5. Application Layer - i.g Web Browser (HTTP HyperText Transfer Protocol) (FTP File Transfer Protocol) (SMTP Simple Mail Transfer Protocol)
4. Transport Layer -TCP/UDP - <u>Segment/Datagram</u>
3. Network Layer - IP (Every time connected with network, has its own IP address) - Routing Protocols - <u>packet</u>
2. Data Link Layer - Wi-Fi/Ethernet - <u>frame</u>
1. Physical Layer (PHY) - <u>bits</u>
<u>Protocol :</u> Same layer levels, different computers
<u>Interface:</u> Same computer, different layer levels
<u>PDU (Protocol Data Unit):#</u> Messages that are exchanged between peer entities实体 #N-PDU between layer N entities #I.e. TPDU = Segment. NPDU = Packet. DPDU = Frame
<u>SDU (Service Data Unit) :</u> #Data unit across an interface #N-SDU: Data unit between Layer N and Layer N+1
<u>Strict Layering:</u> Each layer only look at its own header and interface data to do its job. ⇒ changes to one layer doesn't cause other layers to be reimplemented

The idea behind **strict layering**, which is implicit in the ISO model, is that *each layer should only look at its header and interface data to do its job.* Thus lower layers strip off their headers before passing up messages, and lower layers should not "peek" into higher layer headers. There is an obvious software engineering reason for this: we do not want changes to one layer do not cause other layers to be reimplemented. Imagining retrofitting millions of TCP implementations when Microsoft Outlook changes its mail protocol.





Physical Layer

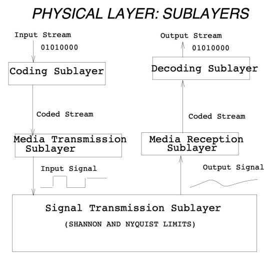
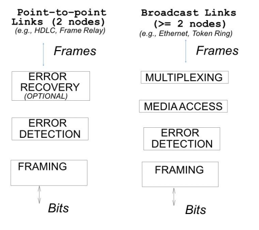


Figure 2: Physical Layer Sublayers


Point-to-point Links (2 nodes)
(e.g., HDLC, Frame Relay)
Frames
ERROR RECOVERY (OPTIONAL)
ERROR DETECTION
FRAMING
Bits
Broadcast Links (>= 2 nodes)
(e.g., Ethernet, Token Ring)
Frames
MULTIPLEXING
MEDIA ACCESS
ERROR DETECTION
FRAMING
Bits
QUASI-RELIABLE 1 HOP FRAME PIPE

- **<u>Coding and decoding layer</u>**
  - Clock Recovery
    - Asynchronous Coding

The actually coding consists of the following rules. First, low amplitudes are used to encode a 1, and high amplitudes are used to encode a 0 (this is the oppposite of the convention we have followed so far!). We add a parity bit to each character and then "frame" the character with an extra start bit and 1-2 stop bits. The start bit is a 0 and the stop bit is a 1. Thus regardless of the value of the data character, the first bit sent is a 0 and the last bit is a 1. This is shown in Figure 9. Notice that the start bit is a high amplitude because of the peculiar way we code bits!

    - Synchronous Coding

The so-called *synchronous coding* techniques attacks both these problems by using a frame size that is (typically) much larger than a character, in the order of thousands of bits. In this case, the overhead of the preamble (i.e., start bits) and postamble (i.e., stop bits) is amortized over a large number of bits and can become small. In addition, the phase information learnt from the preamble is used to clock the remaining bits in the frame as well. Thus the inter-frame time needed for reliable phase detection is again paid once for a large group of bits. The bits within a frame can now be sent at very high speeds.

*<u>NRZ: 0 by another(not 0), 1 is 1</u>
*<u>4-5 Codes: used in higher speed physical layer.4 data bits are encoded as 5 transmitted bits. e.g. 000000001</u>
*<u>Manchester encoding</u>
#-1.5->1.5 is 0. 1.5->-1.5 is 1 #(-) Two coded bits one <u>real bits</u>. ⇒ efficiency is poor. (50% efficiency) #(+) self-clocking #This refers to the fact that every bit, regardless of its value, provides a transition that can be used to sample the bit value. #(+) DC balanced #Still need preamble (typically 0101010101… trailing 11 in Ethernet)
*<u>AMI</u> #A 0 is encoded as 0V, but a 1 is encoded as alternating between 1.5V and -1.5V. #This guarantees (+) DC balance but does (-)not guarantee transitions #(-) poorer immunity to noise (lower signal to noise ratio) than, say, NRZ or Manchester. #This is because, using AMI, a noise signal of amplitude 0.75V can confuse the receiver. In Manchester and NRZ, it takes double the amount of noise (1.5 V) to confuse the receiver.
⇒ baseband coding (is coding using energy levels (say voltage or light)
Broadband coding (the information is modulated on a carrier wave of a certain frequency) 移线: use the signal to modulate a high frequency signal(carrier) + can be viewed as the product of the two signals

- **<u>Media dependent sublayer</u>**

<u>Media Affects Protocols</u>: The choice of media affects the way protocols have been designed or are evolving. We give examples below.
<u>Each Type of Media has its Range of Applicability</u>: It is important to study and understand the pros and cons of different media in order to design a network
<u>Wired:</u> (-) not easy in crowded cities (-) expensive in offices and campuses
<u>Twisted Pair Copper (standard telephone wire)</u>
(+) Low Bandwidth(+) Cheap(+) Easy to install (-) unsuitable不适用 for sending data at high rates. Hard to send data faster than 50 Kbps(without compression), possible slightly over 100 kbps (w. compression) #Better loading cables (telephone company) ⇒ higher rates (bps bits per second) #Remove cables ⇒ (-Reduce length) ⇒ ADSL (Asymmetric digital subscriber line)#Easily available in the office and the home #UTP (Unshield Twisted Pair无屏蔽双绞线): bunch of 8 wires twisted together in 4 pairs to reduce electrical interference
<u>Coaxial cable 1.</u>Baseband #Smaller distance than broadband signalling 2.Broadband (-) security problem(+) cheap broadcast protocols #High bandwidth. Still used in cable networks for cable TV and for data via cable modems调制解调器
<u>Fiber:</u> Huge bandwidth (+) light/thin and easy to install (+) excellent noise immunity to electrical noise and disturbances干扰 as it carries light (+) secure. #Almost impossible to tap窃听 #Point−to−point Secure(-) expensive(-) unidirectional

the transmission rate by reducing x. To reduce or eliminate modal dispersion, the fiber width is reduced till there the only way (or mode) for light to travel is "straight down the middle". This is called *single mode* fiber as opposed to *multimode* fiber. Single mode fiber is often at most 1 micron in diameter and more expensive than multimode fiber. To eliminate chromatic dispersion, we use more accurate sources, like lasers, that emit light in a very small frequency range. Many high speed

<u>Wireless #</u>Widely varying channel bandwidths/distances(-) Extremely vulnerable易受攻击 to noise and interference (+) excellent bypass in developing countries get around lack of infrastructure基础设施 and the need for right of way 先行权
<u>Microwave #</u>Extremely microwaves(-) can be affected by rain #(?) avoid the need for right of way#(+) Maybe cheaper than installing cable
<u>Satellite</u>(+) cheap broadcast protocols(-) security problem with broadcast(?) avoid right of way(+) good bandwidth 500Mhz(+) World wide(-) Large latency(-) Antenna天线 cost
<u>Infrared红外线</u>(-) easily absorbed by obstacles干扰 (+) useful for wireless computing and laptops because do not require wires and much power

Medium	Speed	Distance Span	Pros	Cons
Twisted Pair	1 Mps -1 G (Cat 1 – Cat 5)	1 – 2 Km	Cheap, easy to install	Low distance
Digital Coax	10-100 Mbps	1- 2 km	broadcast	Hard to install in building
Analog Coax	100-500 Mbps	100 Km	Cable companies Use it now	Expensive amplifiers
Fiber	Terabits	100 km	Security, low noise, BW	No broadcast, Needs digging
Microwave	10-100 Mbps	100 km	Bypass, no right Of way need	Fog outages
Satellite	100-500 Mbps	worldwide	Cost independent of distance	250 msec delay Antenna size
RF/Infrared	1 – 100 Mbps, < 4 Mbps	1 km 3 m	wireless	Obstacles for infrared

- **Transmission sublayer**

1.Fourier Transform (harmonic)
2.Nyquist limit- sluggishness affects the max signaling rates: Sending signal at rate of 2B signals/sec (max baud rate) without causing intersymbol interference (* ⇒ low bandwidth let to tight encoding (media) *)
3.Shannon theorem -may use diff voltage for diff output - noise affects the max bit rates:
Bit rate = #bit per symbol * baud rate
Naive bound: log(S/2N) bits per symbol * 2Bl, S: Maximum signal amplitude
Shannon Bound: B log(1+S/2N)
Max noise tolerate: smallest vertical interval / 2

Data Link

However, if the physical links have very low bit error rates, then Data Link overhead can actually decrease performance. This is because part of the physical layer bandwidth is wasted due to acks. Also, sending Data Links have to buffer frames for possible retransmission; this adds complexity and slows down the Data Link processing. This tradeoff is illustrated by the following data. On telephone links with poor bit error rates (say 10−4)^2, it is often a good idea to run a reliable Data Link protocol like HDLC. However, on fibre links or local area networks that have good bit error rates, it pays to use unreliable Data Links (i.e., Data Links that do only error detection) like ATM, Frame Relay, and Ethernet.

- Framing
  - Flags and Bit Stuffing
    - Stuffer -> add flag -> remove flag -> destuff
    - HDLC, 01111110, stuff 0 after 11111
    - a flag cannot occur between stuffed bits.

+the stuffed bit cannot be part of a flag
+a spurious flag cannot be formed by the last few data bits and the first few bits of the real final flag

  - Start Flags and character count: Use flags to indicate the start of frame and then a length field at the start of frame to indicate how many bits to the end of the frame
  - Start and End Flags supplied by Physical Layer
- Hop by hop only performance.

When transfering a file, the end-to-end argument implies that we should add end-to-end checksums to guarantee the end-to-end integrity of the data and not rely on the hop-by-hop data integrity provided by Data Links. In fact, the most common transport protocol on the Internet is the Transmission Control Protocol (TCP) that allows a transport level checksum. However, since TCP checksums increase processing time significantly, this option is often not used. This increases the reponsibility of Data Links to provide good per hop integrity (i.e., probability of undetected errors should be extremely low). However, the result is a violation of the end-to-end argument.

Error detection

Hamming Distance

What is the significance of the Hamming Distance? It represents the smallest number of errors needed for codeword b to be detected (incorrectly) as b′ or vice versa. Define the Hamming Distance of a coding scheme to be the smallest Hamming Distance between any two codewords. For error detection, we want the Hamming Distance of the code to be as large as possible.
This is because if the Hamming Distance is d, it takes a d bit error for one valid codeword to be mistaken for another. An arbitrary (i.e., random) d − 1 bit error will result in an invalid codeword that can be discarded. Thus a codeword with Hamming distance d can detect all d − 1 bit errors; in other words to detect d bit errors, we need a code with distance of d + 1. Parity codes have a Hamming Distance of 2 (since all codewords have an even number of 1's) and so can detect 1 bit errors.

Detecting single bits and odd bits error

Single bit errors result in the addition of an x^i term. x^i will not divide $G(x)$ if the CRC polynomial $G(x)$ has at least two terms. To see this just think of your normal polynomial intuition: can you multiply a two term polynomial by something and get a one term polynomial? No way. The result will have at least two terms. In particular, it will have one distinct term corresponding to the product of the two greatest powers in the two factors, and one distinct term corresponding to the product of the two least powers.

Similarly, two bit errors correspond to adding $x^i + x^j$ which is the same as $x^i(1 + x^{j-i})$. This will not divide $G(x)$ if $G(x)$ does not divide $x^k + 1$ for sufficiently large k that is larger than the message length in bits. Most CRC polynomials like CRC-16 and CRC-32 are *chosen* to satisfy this property. This an important reason to choose special polynomials.

Finally, odd bit error polynomials are never divisible by $x + 1$. Suppose E is a polynomial caused by an odd number of bit errors. Then E has an odd number of terms. Thus if we substitute $x = 1$ in E, we will get 1. Now suppose E was divisible by $x + 1$. Then $E = Y(x + 1)$ for some polynomial Y. Then if we substitute $x = 1$ we would get $E = Y(1 + 1) = Y(0) = 0$. But we have just seen that we cannot get 0 when we substitute $x = 1$ in E. This contradiction can only be resolved if $x + 1$ does not divide E.

Using this observation, we can make the CRC catch all odd bit errors if $x + 1$ is a factor of G. If $G = Y(x + 1)$ for some Y, then clearly G cannot divide any polynomial that does not divide $x + 1$. Thus the CRC will catch all odd bit errors if G is a multiple of $x + 1$. (This is not as big a deal as it sounds because a single parity bit can detect all odd bit errors!).

Error Recovery

Correct Code of Stop and Wait

```
-----------Sender Code--------------------------
Sender keeps state variable SN, initially 0 and
repeats following loop

1) Accept a new packet if available from higher layer and
store it in buffer B
2). Transmit a frame Send (SN, B)
3). If error-free (ACK, R) frame received and R != SN then
    SN = R
    Go to Step 1
Else if the previous condition does not occur after T sec
    Go to Step 2

----------- Receiver Code -----------------------------------
    Receiver keeps state variable RN, initially 0

When an error free data frame (S, D) is received
  On receipt:
    If S = RN then
      Pass D to higher layer
      RN = RN + 1;
      Deliver data m to client.
    Send (ACK, RN) // Send ack unconditionally!
```

1

Code of Alternating Bit

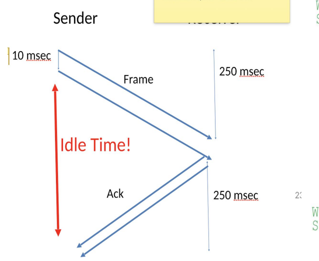
```
-----------Sender Code--------------------------
Sender keeps state bit SN, initially 0 and
repeats following loop

1) Accept a new packet if available from higher layer and
store it in buffer B
2). Transmit a frame Send (SN, B)
3). If error-free (ACK, R) frame received and R != SN then
    SN = R
    Go to Step 1
  Else if the previous condition does not occur after T
   Go to Step 2

----------- Receiver Code -----------------------------------
    Receiver keeps state bit RN, initially 0

When an error free data frame (S, D) is received
  On receipt:
    If S = RN then
      Pass D to higher layer
      RN = ~ RN ; //flip bit!
      Deliver data m to client.
    Send (ACK, RN) // Send ack unconditionally!
```

1

Stop and wait on Satellite Link

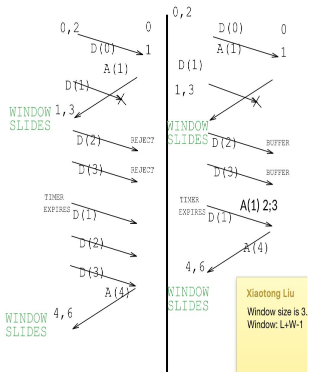
- 1-way Propagation Delay: 250 msec
- Transmission speed: 100 kbit/sec
- Frame size: 1000 bits.
- What is throughput?

Xiaotong Liu Thu 10:57 AM
Pipe size= round propagation delay(0.5) * Transmission speed(10^5) = 50000 bits
Frame # = Pipe size/Frame size = 50
10msec = Frame size / transmission speed
Sends 1000 bits(1 frame) every 500 msec
Throughput = 2000 bits/sec
2%percent efficiency = throughput/ transmission speed = 20000/100k



23

GO BACK 3 SELECTIVE REJECT



Xiaotong Liu
Window size is 3.
Window: L+W-1

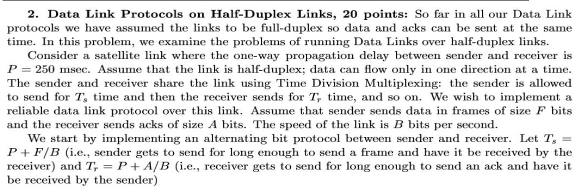
- **Throughput**: measures the number of jobs completed per second.
- **Latency**: measures the worst-case time to complete a job.
- **Transmission Rate**: at Datalink, is the rate at physical sends bits.
    Bandwidth at phy layer Transmission rate at datalink
- **Propagation Delay**: limited by the speed of light to at least 3usec/km   *Propagation = Distance / speed of lights*
- Pipeline: crucial that they use pipelined error recovery protocol.
Since the speed of light constant, transmission rates keep improving bandwidth-delay product keep increasing. ## *(pipe size = round propagation delay * Transmission speed ## frame # = pipe size / frame size ## throughput = frame size/*

*2*propagation delay* ## utilize at *throughput/speed %* of link bandwidth)

Pipe size= round propagation delay(0.5) * Transmission speed(10^5) = 50000 bits
Frame # = Pipe size/Frame size = 50
10msec = Frame size / transmission speed
Sends 1000 bits(1 frame) every 500 msec
Throughput = 2000 bits/sec
2%percent efficiency = throughput/transmission speed = 20000/100k

---

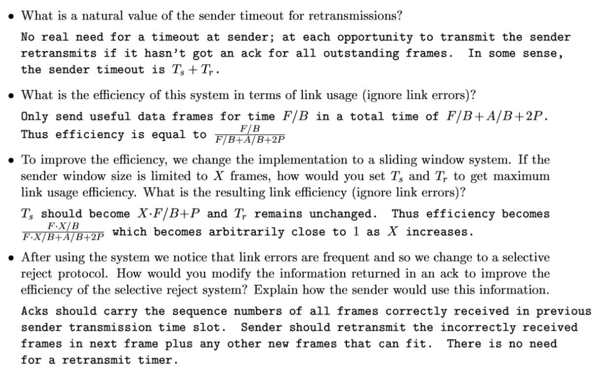
Layering and Interfaces: Many of today's routers implement fireworks. Then the routers at the boundaries of DEC networks will allow email from the outside world but not other applications such as Telnet. Why Peter is unhappy about fireworks? Because looking at email headers in routers is a layer violation.
Clock Recovery: Why eye patterns are often used by real communication engineers in a lab? Because they provide a quick visual check of the quality of the link, in particular the amount of intersymbol interference, and the sampling margin.
Shannon Limit: Hugh Hopeful is working with a transmitter that can only send at two amplitude levels. Hugh assumes that he can never send faster than Nyquist rate. Why is Hugh wrong? Because there are other ways to send more than 1 bit per symbol such as using multiple phases or frequencies.
Media: Why lasers and single-mode fibre, though more expensive, are sometimes used to send light over a fibre link? Because they avoid chromatic and modal dispersion which reduces ISI which in turn allows sending at higher bit rates.
Coding, Preambles and Transitions: Why 11111111 is a reasonable preamble for AMI coding but not for Ethernet. Because in AMI the 1 levels alternate at clock boundaries allowing the receiver to determine bit boundaries, but in Manchester 11111111 and 00000000 are identical except for 90 degree phase shift.
CRCs and Hamming Distance: Why frames, after adding CRC-32, have a Hamming Distance of at least 4 for reasonable frame sizes. Because we argues that CRC can catch all odd errors and it can catch even errors(by making x^k not divisible by CRC) and so it catches up to 3 random errors; Hence its Hamming Distance must be at least one larger or 4.
Data Link: Why undetected error rates on Data Links should be several orders of magnitude lower than the lost frame rate? Because undetected errors at each hop can lead to router errors and (worse) even undetected errors at the receiver if there is no end-t- end checksum
Data Link: Why it is sometimes not worth doing error recovery at the Data Link? Because it is expensive to buffer packets for retransmission at routers and to send ack on each hop(which costs extra bandwidth).
LANs and multiplexing: Why most LANs do statistical multiplexing instead of strict multiplexing? Because data traffic is bursty and the number of active users is typically smaller than the total number of users. Thus stat muxing divides bandwidth among active users, giving each active user typically a larger share.
LANs and wide area networks: Why its reasonable to have high bandwidths for Local Area Networks and lower bandwidths for wide area networks? Because traffic exhibits locality: there is more traffic local to LAN than outside a LAN.



2. Data Link Protocols on Half-Duplex Links, 20 points: So far in all our Data Link protocols we have assumed the links to be full-duplex so data and acks can be sent at the same time. In this problem, we examine the problems of running Data Links over half-duplex links.

Consider a satellite link where the one-way propagation delay between sender and receiver is $P = 250$ msec. Assume that the link is half-duplex; data can flow only in one direction at a time. The sender and receiver share the link using Time Division Multiplexing: the sender is allowed to send for T_s time and then the receiver sends for T_r time, and so on. We wish to implement a reliable data link protocol over this link. Assume that sender sends data in frames of size F bits and the receiver sends acks of size A bits. The speed of the link is B bits per second.

We start by implementing an alternating bit protocol between sender and receiver. Let $T_s = P + F/B$ (i.e., sender gets to send for long enough to send a frame and have it be received by the receiver) and $T_r = P + A/B$ (i.e., receiver gets to send for long enough to send an ack and have it be received by the sender)



- What is a natural value of the sender timeout for retransmissions?

  No real need for a timeout at sender; at each opportunity to transmit the sender retransmits if it hasn't got an ack for all outstanding frames. In some sense, the sender timeout is $T_s + T_r$.

- What is the efficiency of this system in terms of link usage (ignore link errors)?

  Only send useful data frames for time F/B in a total time of $F/B + A/B + 2P$. Thus efficiency is equal to $\frac{F/B}{F/B+A/B+2P}$

- To improve the efficiency, we change the implementation to a sliding window system. If the sender window size is limited to X frames, how would you set T_s and T_r to get maximum link usage efficiency. What is the resulting link efficiency (ignore link errors)?

  T_s should become $X \cdot F/B + P$ and T_r remains unchanged. Thus efficiency becomes $\frac{F \cdot X/B}{F \cdot X/B+A/B+2P}$ which becomes arbitrarily close to 1 as X increases.

- After using the system we notice that link errors are frequent and so we change to a selective reject protocol. How would you modify the information returned in an ack to improve the efficiency of the selective reject system? Explain how the sender would use this information.

  Acks should carry the sequence numbers of all frames correctly received in previous sender transmission time slot. Sender should retransmit the incorrectly received frames in next frame plus any other new frames that can fit. There is no need for a retransmit timer.

Layering and Interfaces: When the data link software in computer receives a packet P, it directly calls the transport layer (through an interface) to ask where P should be placed in memory. The software copies P to the specified buffer, and then hands it over to the routing layer for normal processing. a) This is not a layer violation because neither transport or data link is looking at each other's header b) This is a layer violation because calling between layers should only flow downwards c) This is a layer violation but can easily be fixed by the Data Link calling routing who in turn calls Transport. d) This is a layer violation and cannot be fixed because two non-adjacent layers should not pass information to each other
Media: Most laptops use 802.11 (WiFi) which uses radio transmission. What would go wrong (or right) if the 802.11 standard decided to use infrared instead of radio? a) The 802.11 bit rate would reduce because Infrared has lower frequency b) The 802.11 bit error rate because Infrared has more noise c) 802.11 would have less interference because neighboring Access Points would not interfere with each other just as 2 infrared keyboards do not interfere with each other in adjacent cubicles. d) 802.11 would not work if there was any obstacle between the sender and receiver
Bit Stuffing and Transitions: Bit stuffing and physical layer coding to ensure transitions are both forms of coding. If you use bit stuffing at the Data Link can you ensure sufficient transitions at the physical layer? a) Yes because we ensure that for every 5 1's we add a 0 b) No, because the data could be all zeroes c) No, because the data could be 010101 . . and this will confuse the receiver about clock boundaries. d) Yes, because we can rely on the data to be sufficiently random
Clock Recovery: Phased Locked loops have the following property: a) They quickly correct drift in the receiver clock b) They are fooled by noise spikes that appear to be transitions c) They do not need a preamble to set up synchronization when the frame first starts. d) They are slow to adapt but resilient to noise
Latency vs Throughput: Many modems do time-consuming compression algorithms before

they send data so that: a) The latency increases and the throughput increases

Multiplexing: When a voice call is made, 64kbps of data bandwidth is reserved across any digital line in the path of the call. Why is strict multiplexing reasonable for a voice call instead of statistical multiplexing? a) Voice calls are not bursty b) Voice calls are bursty c) Voice calls need predictable bandwidth guarantees compared to say email d) Both a) and c) e) Both b) and c)

Error recovery versus error correction: On a satellite link, sometimes engineers use error correction instead of error recovery because: a) Satellites have a high error rate and it's always good to do error recovery when the error rate is high b) It reduces the latency to recover from a small error c) It costs too much bandwidth to retransmit messages when there is an error as in the example of CDs a) It reduces the overhead for the checksum

Ethernet and the End-to-end argument: Ethernets retransmit collided frames hop-by-hop, but let frames lost (due to bit errors) be retransmitted end-to-end. This can be explained as follows in terms of the end-to-end argument. a) Collisions are frequent but bit errors are rare b) Retransmission is an artifact of an older technology (like HDLC) but its not worth it because errors are much less frequent today c) Retransmission of collided frames is needed for correctness. d) Doing retransmission at the Data Link removes the need for a TCP end to end acknowledgement

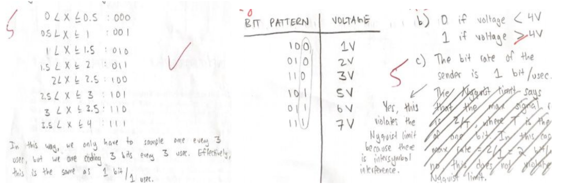
Nyquist and Shannon Limits Reconsidered: Suppose the sender sends bit 1 using 4 Volts for 1 usec, and sends bit 0 by using 0 Volts for 1 usec. Unfortunately, the channel is quite sluggish and the response to a 1 bit takes 3 usec to die out. As shown in the figure, the output of the channel (when a 1 is sent) stays at 4V for 1 usec, goes to 2 Volts for 1 usec, then stays at 1 Volt for 1 usec, before dying to 0 Volts after 3 usec. Sending 0 Volts will result in an output of 0 Volts. Just to go against tradition, Henry Nyquist (the grandson of Harry Nyquist) decides to send bits once every usec. Thus Henry is deliberately incurring intersymbol interference! However, he feels that a controlled amount of Intersymbol Interference can be detected at the receiver. Assume the sender is sending a bit every 1 usec. When the voltage of the current bit's output interferes with the voltages of the previous bits, assume that the resulting output is just the sum of the voltages.

a) By writing down all possible combinations of the current bit and the previous bits that can intefere, show the resulting voltage for each such combination. (8 points)

b) How can the receiver use the results of a) to correctly detect the current bit despite intersymbol interference. (2 points)

c) What is the effective bit rate of the sender. Does this violate the Nyquist limit? (5 points)

d) Cyril Shannon, the grandson of Claude Shannon, hears about Harry's new scheme and is not very impressed. He claims that his grandfather could have done the same thing in a much simpler way. Explain.



5

0 ≤ X ≤ 0.5	: 000
0.5 ≤ X ≤ 1	: 001
1 ≤ X ≤ 1.5	: 010
1.5 ≤ X ≤ 2	: 011
2 ≤ X ≤ 2.5	: 100
2.5 ≤ X ≤ 3	: 101
3 ≤ X ≤ 3.5	: 110
3.5 ≤ X ≤ 4	: 111

✓

In this way, we only have to sample once every 3 usec, but we are coding 3 bits every 3 usec. Effectively, this is the same as 1 bit / 1 usec.

BIT PATTERN	VOLTAGE
100	1V
010	2V
110	3V
101	5V
011	6V
111	7V

b) 0 if voltage < 4V
1 if voltage ≥ 4V

5

c) The bit rate of the sender is 1 bit/usec.

Yes, this violates the Nyquist limit because there is intersymbol interference.

~~The Nyquist limit says that the max signal is 2/T, where T is the of one bit. In this case max rate = 2/1 = 2 bit/ no, this does not violate Nyquist limit.~~

Spam and layering:ISFS use anti-spam devices at the network that check for the content of every packet to see if it is likely spam and drop those packets. The ISP claims that looking at emails content(not header) is not a layer violation. More specifically, they claim that if the routing or transport layer change, they would not have to reprogram their anti-spam device. explain what is wrong. [sol]Need to unpack. Looking for path.

Media impairment: cat 5 cable has nyquist bandwidth of 100 Mhz, in order to transmit at 100 Mbps Over cat 5 cable. why can you not use Manchester encoding [sol]Nyquist bandwidth =nyquist limit. thus can send 100*10^6 signal/sec. however, each bit in manchester encoding need two signals. thus can only transfer 50 Mhz

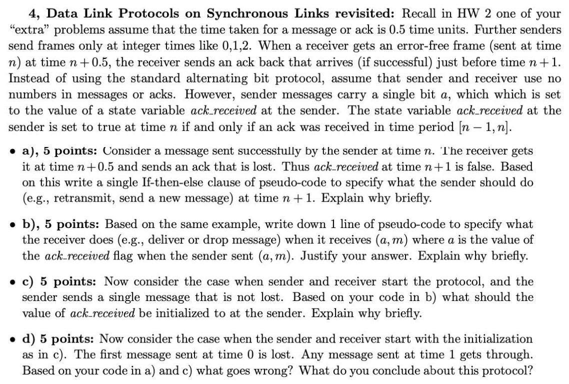
Clock recovery: What happened to an eye pattern when bandwidth becomes too small? small

preambles and transitions: why is 01011 01011 01011 a good preamble for 4-5 encoding whereas 0101010101 is a good preamble manchester encoding (hint 4-5 encoding besides bit sync what other sync is needed) [sol] 010101 is good because it ensure transition only happens half time unit,4-5 send 4 data bit for encoding. this guarantee transition 01011...ensure that receiver also in sync regarding each bit sent

Media: Because radio ranges are approximate, it is hard to ensure that the 802.11 networks of neighboring homes do not overlap. What property of 802.11 should the home owners exploit to minimize the interference? Multiple channels. Diff people dff channels.

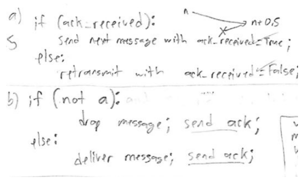
Latency vs throughput: In the old days of batch processing, a number of jobs were run at the same time on IBM 360. Explain in terms of latency and throughput what batch processing is trade off. Throughput increases. Latency increase.

End-to-end argument: In the internet, vendors sometimes require that their packets be encrypted. Why better to have packet encrypted by the src and decrypted at the destination instead of hop-by-hop? Overhead. Security

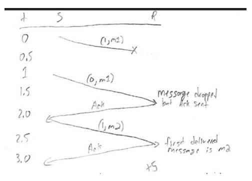


4, Data Link Protocols on Synchronous Links revisited: Recall in HW 2 one of your "extra" problems assume that the time taken for a message or ack is 0.5 time units. Further senders send frames only at integer times like 0,1,2. When a receiver gets an error-free frame (sent at time n) at time $n+0.5$, the receiver sends an ack back that arrives (if successful) just before time $n+1$. Instead of using the standard alternating bit protocol, assume that sender and receiver use no numbers in messages or acks. However, sender messages carry a single bit a, which which is set to the value of a state variable *ack_received* at the sender. The state variable *ack_received* at the sender is set to true at time n if and only if an ack was received in time period $[n-1, n]$.

- **a), 5 points:** Consider a message sent successfully by the sender at time n. The receiver gets it at time $n+0.5$ and sends an ack that is lost. Thus *ack_received* at time $n+1$ is false. Based on this write a single If-then-else clause of pseudo-code to specify what the sender should do (e.g., retransmit, send a new message) at time $n+1$. Explain why briefly.
- **b), 5 points:** Based on the same example, write down 1 line of pseudo-code to specify what the receiver does (e.g., deliver or drop message) when it receives (a, m) where a is the value of the *ack_received* flag when the sender sent (a, m). Justify your answer. Explain why briefly.
- **c) 5 points:** Now consider the case when sender and receiver start the protocol, and the sender sends a single message that is not lost. Based on your code in b) what should the value of *ack_received* be initialized to at the sender. Explain why briefly.
- **d) 5 points:** Now consider the case when the sender and receiver start with the initialization as in c). The first message sent at time 0 is lost. Any message sent at time 1 gets through. Based on your code in a) and c) what goes wrong? What do you conclude about this protocol?



```
a) if (ack_received):
5      send next message with ack_received=True;
   else:
       retransmit with ack_received = False;
b) if (not a):
       drop message; send ack;
   else:
       deliver message; send ack;
```



c) true.

7. There are 2^{n-2} burst errors of length n. There are 2^{n-k-2} polynomials that are multiples of a generator with degree k and also correspond to a burst error of length n. Thus, the fraction of undetectable burst errors is $\frac{2^{n-k-2}}{2^{n-2}} = \frac{1}{2^k}$

3. Data Link Protocols on Synchronous Links: So far in all our Data Link protocols we have assumed the links to be asynchronous in that the delay of a frame or ack could be arbitrary. Now we consider the case that the time taken for a message or ack is 0.5 time units. Further senders send frames only at integer times like 0,1,2. When a receiver gets an error-free frame (sent at time n) at time $n + 0.5$, the receiver sends an ack back that arrives (if successful) just before time $n + 1$. Suppose we use the standard alternating bit protocol except that the sender also waits to send at integer times.

- Does the sender need to number the data frames? If your answer is yes give a counterexample to show what goes wrong when it does not. (10 points)
- Does the receiver need to number the ack frames? If your answer is yes give a counterexample to show what goes wrong when it does not. (10 points)
- Describe a simple protocol for the sender to initialize the receiver state after a crash? (5 points)

a) Yes, the sender needs to number the data frames

Counterexample:
Receiver will get duplicate copies.
I--Sender sends a frame which the receiver gets successfully. Receiver sends an ack which is lost.
sender will assume packet is lost when it does not get an ack and retransmits . Receiver will accept this frame without realising it is a duplicate.

b)No, the receiver does not need to renumber the acks.
This is because an ack received at time n+1 could only be an ack for the frame sent at time n.
Since we transmit at discrete time intervals no confusion with old acks is possible.

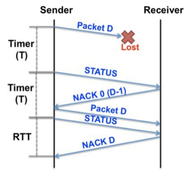
c) A simlple protocol is as follows
Sender waits for 1 time unit for all older acks to die out . It then sends a reset message
Receiver initializes number when it gets reset and sends back a reset ack .

(1) The STATUS packet is needed in Peter Protocol because there are some cases that the sender does not know packet lost happens. For example, if all packets are lost, the sender will no receive any NACK packet from the receiver. Also, the same thing can happen when the last data packet is lost.

(2) Since the sender keeps sending STATUS packets as long as it has data packets to send, it guarantees retransmission if the data packets are lost.

(3) The number of remaining data packets in a sender's queue. If there are data packets to send, then start the timer. Otherwise, stop the timer.

(4) The worst-case latency is '2*T + 1*RTT'. Denote that T is time for STATUS timer and RTT is a sum of latencies between the sender and the receiver.


Sender
Receiver
Packet D
Lost
Timer (T)
STATUS
Timer (T)
NACK 0 (D-1)
Packet D
STATUS
RTT
NACK D