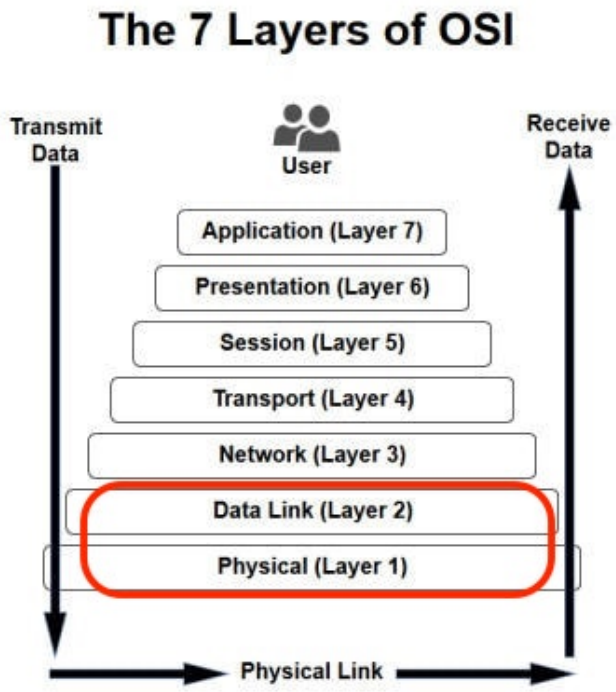
Lecture 3



1 Review of Physical Layer

1.1 IQ Sampling and Modulation

We can transmit data by inverting the carrier frequency according to the data bits, resulting in a modulated signal.
These signals can be used to encode **packets** (the unit of information in the data link layer). Example:

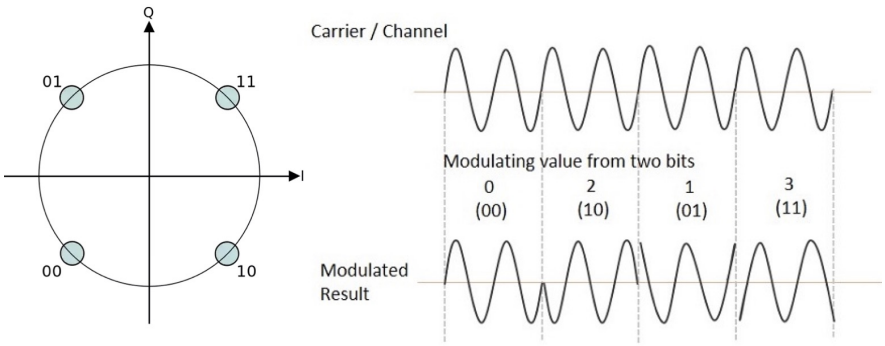


Figure 1: Packet encoded on a carrier frequency

Lecture 3

In this example, the packet encoded by the wave is **00100111**.
This modulation is generated by the following QPSK (Quadrature phase-shift keying) IQ samples (I,Q): [**(-1, 1), (1, 1), (-1, 1), (1, 1)**].

1.2 Baud/Symbol Rate

The **Baud rate** is the rate of communication over a network representing *symbols per second*. For example, assume in Figure 1 that the period between vertical dashed lines represent 1 ms. Then 2 bits are transmitted every ms, or 2000 bits per second (2000 baud).

Again, assume instead of a QPSK modulation scheme, we used BPSK (binary phase shift keying) modulation with a unit interval time of 1 ms. Then the rate of communication is 1000 baud.
One reason we would prefer this slower method of communication is because it performs better in noisy environments.

1.3 Security

At the physical layer, any sequence of bits can be crafted and transmitted through the network. We can generate any packet that we want and inject into into the physical medium. Therefore, nothing can be trusted at this level.

2 Data Link Layer

The goals of our study of the Link layer is to understand the principles behind the services it provides, these include:

- error detection, correction
- sharing a broadcast channel: multiple access
- link layer addressing
- local area networks: Ethernet VLANs

The data link layer has the responsibility of transferring datagram (in the form of packets) from one node to another physically adjacent node over a link. This is implemented individually in each and every host through a network interface card (NIC) or on a chip; this attaches into host's system buses.

Lecture 3

2.1 Introduction, services

2.1.1 Definitions

Nodes Hosts and routers are referred to as nodes in the link layer, some examples include routers, cells, laptops, etc

Links Communication channels that connect adjacent nodes along a communication path, some examples are wired links, wireless links, and LANs

Frame A layer-2 packet, encapsulates datagram

2.1.2 Analogy

- trip from Charlottesville to Manhattan
  - limo: Charlottesville to CHO
  - jet: CHO to LGA
  - train: LGA to Manhattan

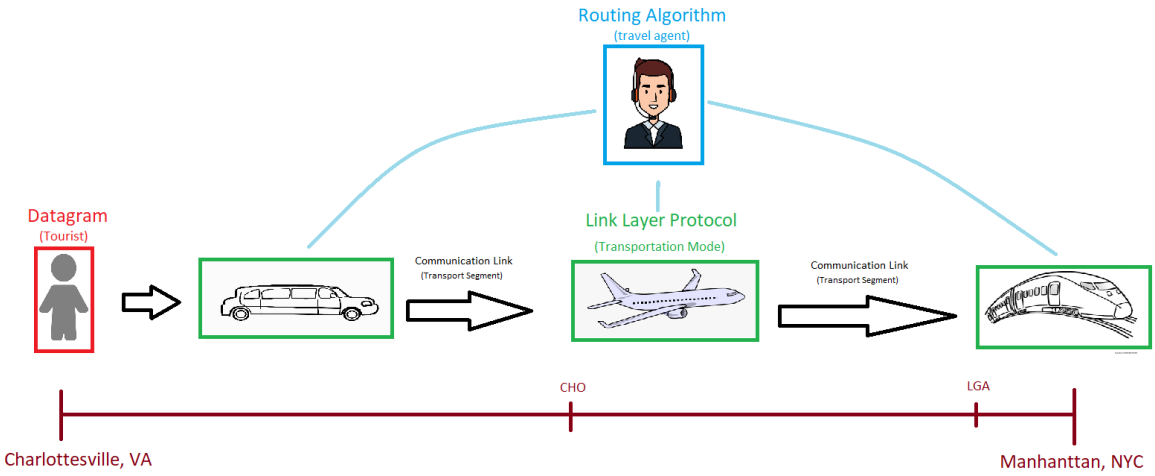


Figure 2: A real world analogy to represent the Link Layer

In this analogy, our tourist represents the datagram, the information we are transmitting over the link layer. The car, plane, and train represent the link later protocol, our method of getting from one node to another. The transportation segment is the communication link connecting the nodes. The locations such as CHO, LGA, and Manhattan are the nodes. And finally, the travel agent booking the modes for the tourist the the routing algorithm, telling the datagram where to go next.

2.1.3 Services

Framing, Link Access Encapsulate datagram into frame, adding header and trailer and channel access if it is across a shared medium. MAC addresses are used in the header to identify source and destination

Reliable Delivery Deliver the datagram with reduced bit error

Flow Control Pacing between adjacent sending and receiving nodes

Error Detection Errors caused by noise, the receiver is tasked with detecting these. Upon detection, must signal sender for retransmission or drop frames

Error Correction Receiver identifies errors and corrects the bit errors without retransmission

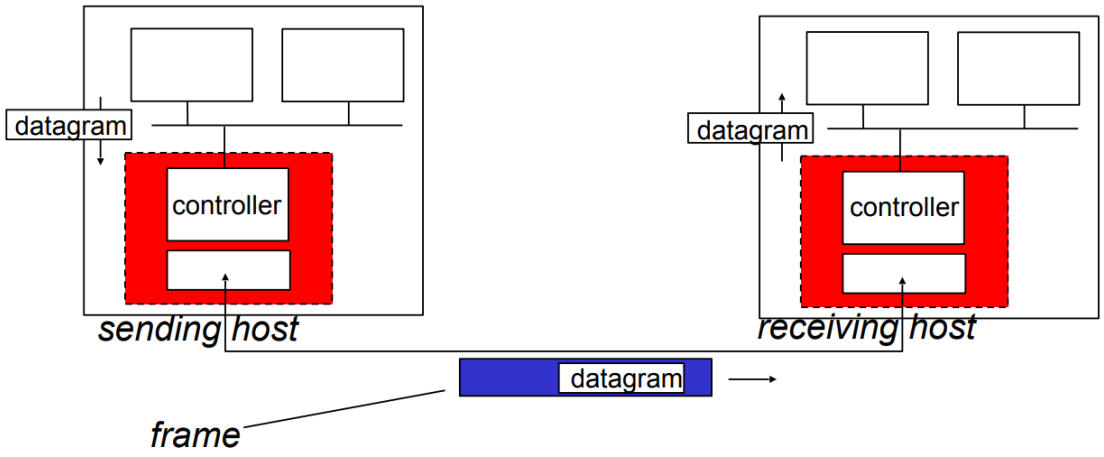


Figure 3: Sender-Receiver relationship outlined

The above figure shows how the sender and receiver communicate the datagram. The sending side encapsulates the datagram in a frame with the proper error checking bits, rdt (real data transport), flow control, etc. The receiving side looks for these communicated errors, rdt, etc. and extracts the datagram, passing it to upper later at receiving side upon validation.

2.2 Error Detection

EDC Error detection and correction bits (redundancy)

D Data protected by error checking, including header fields

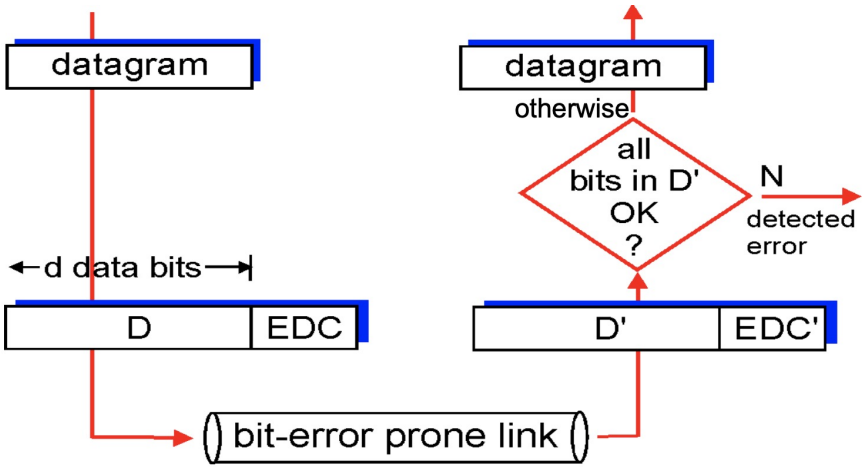


Figure 4: Data flow through an unreliable medium

Because every bit of data (including the EDC bits) flows through a bit-error prone link (Figure 4), error detection can never be 100% reliable. We can add EDC bits to the tail of the packet to achieve error detection and/or correction.

2.2.1 Single bit parity

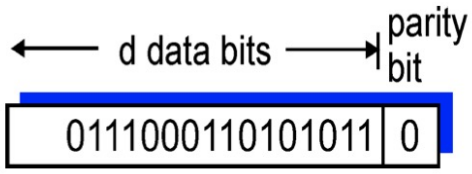


Figure 5: Single bit parity check

However, we can append a single bit to the message to detect single bit errors.
The **parity bit** can be 0 if the number of 1s in the binary message is even and 1 if the number of 1s in the message is odd. However, this method can only detect an error if only a single bit (or more accurately, an odd number of bits) is flipped.

2.2.2 Cyclic Redundancy Check

This method of error-detection is more powerful than the ones discussed above, breaks up the datagram into data bits (D) and crc (R) bits as illustrated below:

This uses modulo-2 arithmetic where addition and subtraction is equivalent to the traditional XOR function and multiplication and divisions are equivalent to base-2 arithmetic.

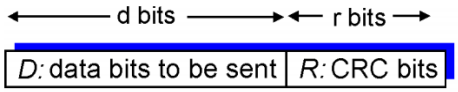


Figure 6: Datagram breakdown for cyclic redundancy check

No operation has carries and borrows, further simplifying the process. In general, we observe that multiplication by 2^k left shifts a bit pattern by k bits.

The sender and receiver first agree on a $|r|+1$ wide pattern known as the generator, which we will denote by G. The validity of the message then, relies on $G|(D*2^r) \oplus R$, meaning that the previously agreed upon generator must divide the XOR of the transmitted message, resulting in a remainder of 0. Otherwise, the receiver is able to deduce that some error exists. This method of error detection breaks down when bit flips occur to form an error in the datagram but the resulting datagram is some multiple of the generator. Although this is a less likely scenario than the detecting methods of single bit parity, the flaw still exists.

The following is a worked out example of CRC, with $G = 1001$, meaning that $|r| = |G| - 1 = 4 - 1 = 3$ and $D = 101110$. This example lets us know what R should be our last r bits of the datagram in order to make it valid.

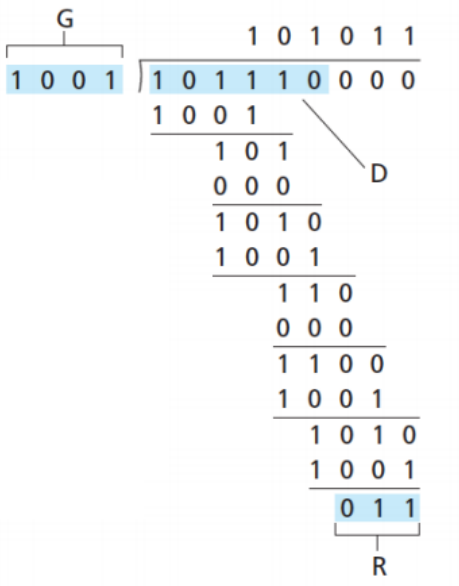


Figure 7: Bitwise long division to determine R

Thus, the sender is concerned with finding the correct R to send and find it using the equation $R = remainder[\frac{D*2^r}{G}]$. Whereas the receiver wants to figure out if the transmitted datagram is valid and does so by confirming the equivalency $nG = (D * 2^r) \oplus R \; \forall n \in \mathbb{N}$.

2.3 Error Correction

Single bit parity checking is proven to be a somewhat effective method of error detection. We can expand this concept to **two-dimensional bit parity**, which allows us to detect the exact bit that was flipped by breaking up the datagram into lines forming a matrix and having parity checks for every row and column. This type of detection allows us to correct the datagram without the need for a retransmission by the sender.

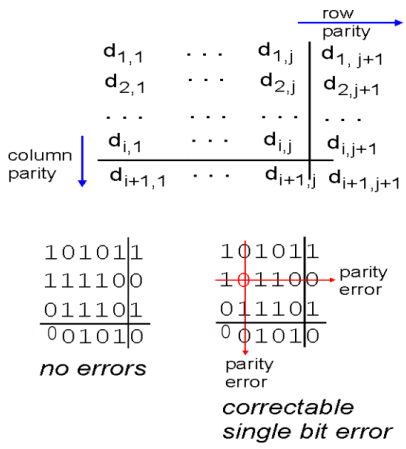


Figure 8: Two-dimensional Bit Parity

2.4 Multiple Access Protocols

There are two types of network links.

Point-to-point link Consists of a single sender at one end of the link and a single receiver at the other end of the link.
(e.g. PPP for dial-up access, point-to-point link between Ethernet switch, host)

Broadcast link Can have multiple sending and receiving nodes all connected to the same, single, shared broadcast channel.
(e.g. old-fashioned Ethernet, upstream (Hybrid fiber-coaxial), 802.11 LAN)

Multiple access problem There is a **collision** if a node receives two or more signals at the same time.

In order to regulate nodes' transmission into a shared broadcast channel, computer network have **multiple access protocols**. This protocol must be shared using the channel itself.