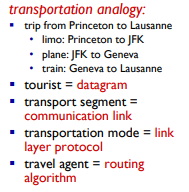
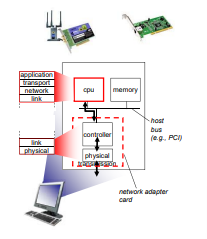
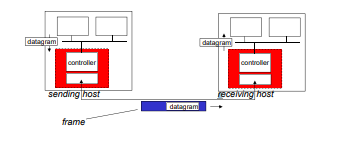
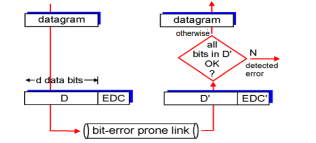
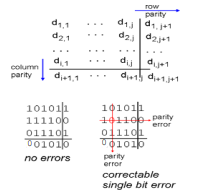
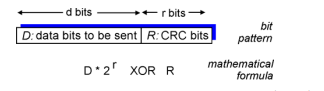
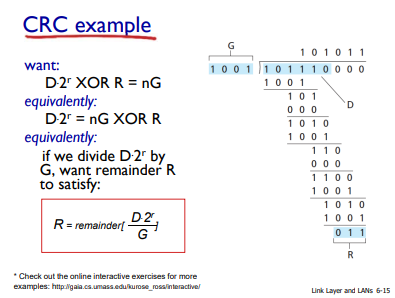
Chapter 6: The Link Layer and LANs

* Link Layer: Introduction
  + terminology:
    - hosts and routers: **nodes**
    - communication channels that connect adjacent nodes along communication path: **links** 
      * • wired links
      * • wireless links
      * • LANs
    - layer-2 packet: **frame**, encapsulates datagram
  + data-link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

Message > Segment > Datagram > Frame (each time you add a header)

* Link Layer: Context
  + datagram transferred by different link protocols over different links:
    - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
  + each link protocol provides different services
    - e.g., may or may not provide rdt over link
  + 
* Link Layer Services
  + framing, link access:
    - encapsulate datagram into frame, adding header, trailer
    - channel access if shared medium
    - “MAC” addresses used in frame headers to identify source, destination
      * different from IP address!
  + reliable delivery between adjacent nodes
    - we learned how to do this already (chapter 3)!
    - seldom used on low bit-error link (fiber, some twisted pair)
    - wireless links: high error rates
      * Q: why both link-level and end-end reliability?
  + flow control:
    - pacing between adjacent sending and receiving nodes
  + error detection:
  + • errors caused by signal attenuation, noise.
    - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
  + error correction:
    - receiver identifies and corrects bit error(s) without resorting to retransmission
  + half-duplex and full-duplex
    - with half duplex, nodes at both ends of link can transmit, but not at same time
* Where is the link layer implemented?
  + in each and every host
  + link layer implemented in “adaptor” (aka network interface card NIC) or on a chip
    - • Ethernet card, 802.11 card; Ethernet chipset
    - • implements link, physical layer
  + attaches into host’s system buses
  + combination of hardware, software, firmware
* 
* Adaptors Communicating
  + Sending:
    - encapsulates datagram in frame
    - adds error checking bits, rdt, flow control, etc.
  + receiving side
    - looks for errors, rdt, flow control, etc.
    - extracts datagram, passes to upper layer at receiving side
  + 
* Error Detection
  + EDC= Error Detection and Correction bits (redundancy) D = Data protected by error checking, may include header fields
  + Error detection not 100% reliable!
    - protocol may miss some errors, but rarely
    - larger EDC field yields better detection and correction
  + 
* Parity Checking
  + Single bit parity: detect single bit errors
    - 
  + Two-Dimensional Bit Parity:
    - Detect and correct single bit errors



* Internet Checksum
  + goal: detect “errors” (e.g., flipped bits) in transmitted packet (note:used at transport layer only)
  + sender:
    - treat segment contents as sequence of 16-bit integers
    - checksum: addition (1’s complement sum) of segment contents
    - sender puts checksum value into UDP checksum field
  + receiver:
    - compute checksum of received segment
    - check if computed checksum equals checksum field value: •
      * NO - error detected •
      * YES - no error detected. But maybe errors nonetheless?
* Cyclic Redundancy Check
  + more powerful error-detection coding
  + view data bits, D, as a binary number
  + choose r+1 bit pattern (generator), G
  + goal: choose r CRC bits, R, such that
    - • exactly divisible by G (modulo 2)
    - • receiver knows G, divides by G. If non-zero remainder: error detected!
    - • can detect all burst errors less than r+1 bits
  + widely used in practice (Ethernet, 802.11 WiFi, ATM)
  + 
* \*\*\*BE CAREFUL. Use XOR, not traditional subtraction while divided to get remainder.
* D = 101110, d= 6, G = 1001, r = 3, R =?
* 
* Multiple access links, protocols
  + two types of “links”:
  + point-to-point
    - • PPP for dial-up access
    - • point-to-point link between Ethernet switch, host
  + broadcast (shared wire or medium)
    - • old-fashioned Ethernet
    - • upstream HFC
    - • 802.11 wireless LAN
  + 
* single shared broadcast channel
* two or more simultaneous transmissions by nodes: interference
  + collision if node receives two or more signals at the same time
* multiple access protocol
  + distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
  + communication about channel sharing must use channel itself! • no out-of-band channel for coordination
* Ideal Multiple Access Protocol
  + given: broadcast channel of rate R bps
  + desiderata:
    - 1. when one node wants to transmit, it can send at rate R.
    - 2. when M nodes want to transmit, each can send at average rate R/M
    - 3. fully decentralized: • no special node to coordinate transmissions • no synchronization of clocks, slots
    - 4. simple