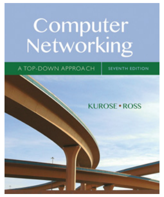
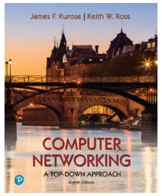
Chapter 6
The Link Layer and LANs

Courtesy to the textbooks' authors and Pearson Addison-Wesley because many slides are adapted from the following textbooks and their associated slides.



Jim Kurose, Keith Ross, "Computer Networking: A Top Down Approach", 7th Edition, Pearson, 2016.



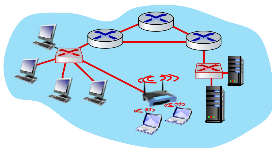
Jim Kurose, Keith Ross, "Computer Networking: A Top Down Approach", 8th Edition, Pearson, 2020.

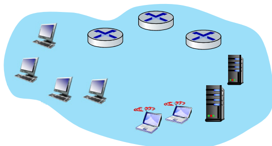
Link layer, LANs: roadmap

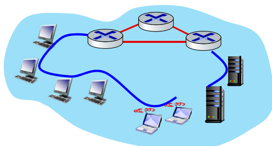
- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking
- a day in the life of a web request

Link layer: introduction

- Terminology
  - host and router are layer-3 nodes
    - switch (and bridge) is layer-2
  - frame is layer-2 packet
- link layer is responsible to transfer packets from one layer-3 node to another via a link (subnet)
  - wired
  - wireless
  - LAN

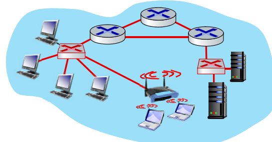






Link layer: context

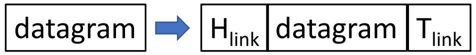
- datagram transferred by different link-layer protocols over different links:
  - e.g., WiFi on first link, Ethernet on next link

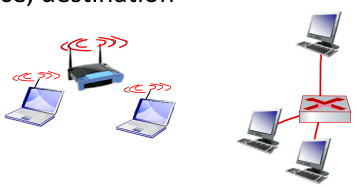


- each link protocol provides different services
  - e.g., may or may not provide reliable data transfer over link

Link layer: services

- framing
  - encapsulate datagram into frame
    - adding header, trailer
  - "MAC" addresses in frame headers identify source, destination
    - MAC address is different from IP address!
- channel access (multiple access, MAC)
  - if shared medium
- reliable delivery in link layer
  - we already know how to do this!
  - seldom used on low bit-error links
  - wireless links: high error rates
  - *Q:* why both link-level and end-to-end reliability?



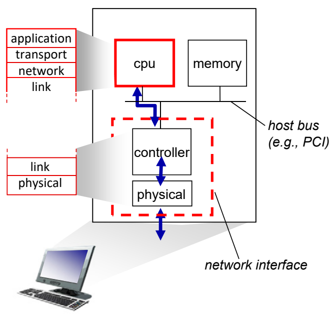


Link layer: services (more)

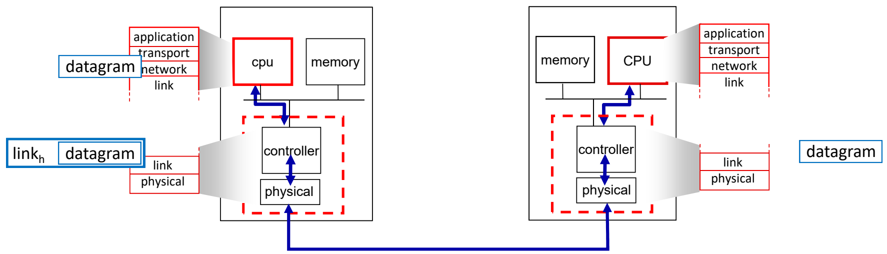
- flow control:
  - pacing between adjacent sending and receiving nodes
- error detection:
  - errors caused by signal attenuation, noise.
  - retransmits or drops frame, if receiver detects errors
- error correction:
  - receiver identifies *and corrects* bit error(s) without retransmission
- half-duplex 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 host
- link layer implemented in *network interface card* (NIC)
  - (Ethernet, WiFi) card or chip
  - implements link+physical layers
- attaches into host's system buses
- combination of hardware, software, firmware



Interfaces communicating



sending side:
- encapsulates datagram in frame
- adds error checking bits, reliable data transfer, flow control, etc.

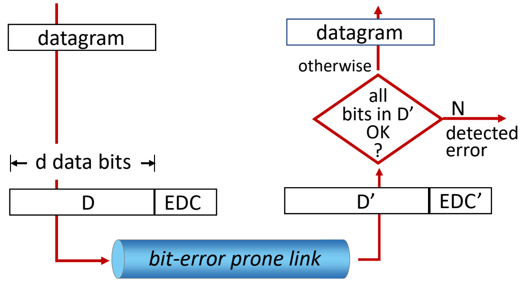
receiving side:
- looks for errors, reliable data transfer, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking
- a day in the life of a web request

Error detection

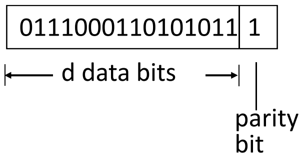
EDC: error detection and correction bits (e.g., redundancy)
D: data protected by error checking, may include header fields



Error detection has its capability
- protocol may miss some errors, but rarely
- larger EDC field yields better detection
  - same as correction

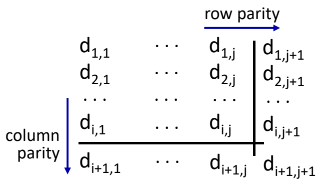
Parity checking

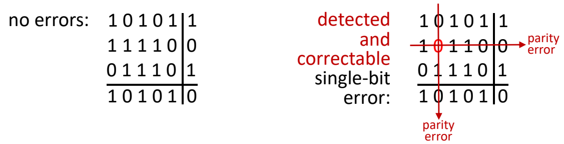
single bit parity:
- detect single bit errors



Even parity: set parity bit so there is an even number of 1's

two-dimensional bit parity:
- detect *and correct* single bit errors





UDP/TCP/IP checksum (review)

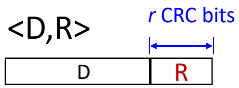
Goal: detect errors (*i.e.,* flipped bits) in transmitted segment

sender:
- treat contents of UDP segment (including UDP header fields and IP addresses) as sequence of 16-bit integers
- checksum: addition (one's complement sum) of segment content
- checksum value put into UDP checksum field

receiver:
- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - not equal - error detected
  - equal - no error detected. *But maybe errors nonetheless?* More later ….

Cyclic Redundancy Check (CRC)

- error-detection code more powerful than TCP/UDP/IP's checksum
  - used by Ethernet and WiFi
  - can detect all burst errors less than *r*+1 bits
- both sender and receiver know G in advance:
  - G: generator of length *r*+1 bits
- sender computes the CRC bits R and transmits <D,R>
  - D: data bits
  - R = $D \cdot 2^r$ % G = $\langle D, \overbrace{00...0}^{r \text{ CRC bits}} \rangle$ % G is the remainder
    - the CRC bits R is of length *r* bits
    - use bitwise-XOR for addition/subtraction
    - <D,R> is divisible by G
- receiver checks out whether <D,R> % G = 0
  - non-zero remainder → error detected
  - zero remainder → either no error or error not detected



Cyclic Redundancy Check (CRC): example

- G = 1001 and D = 101110
  - in binary representations
  - *r* = 3 in this example
- sender computes the remainder R
  - R = $D \cdot 2^r$ % G
    = <D,00...0> % G