The Link Layer I

Smith College, CSC 249
April 1, 2008

slides mostly from J.F Kurose and K.W. Ross,
copyright 1996-2007

Chapter 5: The Data Link Layer

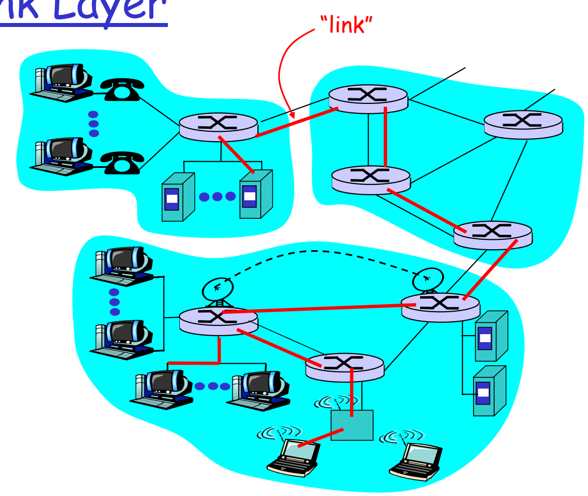
Our goals:

- Understand principles behind data link layer services:
  - error detection, correction
  - reliable data transfer, flow control: *done!*
  - sharing a broadcast channel: multiple access
  - link layer addressing
- Example of link layer technology
  - Ethernet

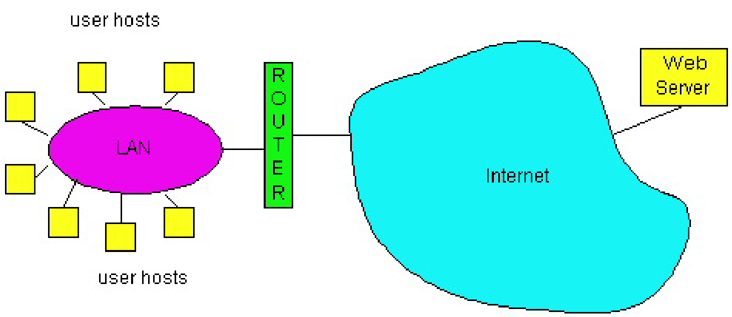
Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing: MAC
- 5.5 Ethernet Implementation
- 5.6 Hubs, switches and routers
- 5.7 PPP

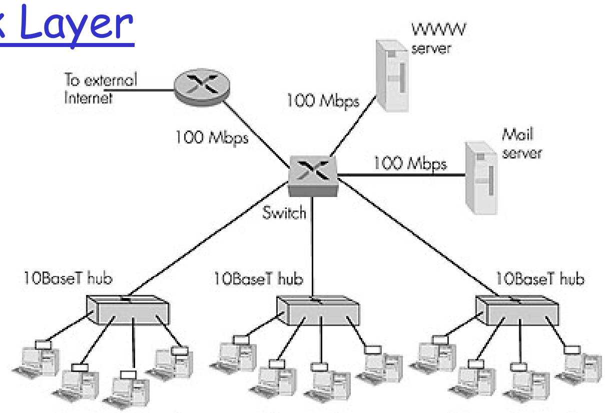
Link Layer



Link Layer



Link Layer



data-link layer has responsibility of transferring a frame from one node to an adjacent node over a link

Link Layer

Some terminology:

- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- layer-2 packet is a **frame**, encapsulates datagram

Link Layer Services

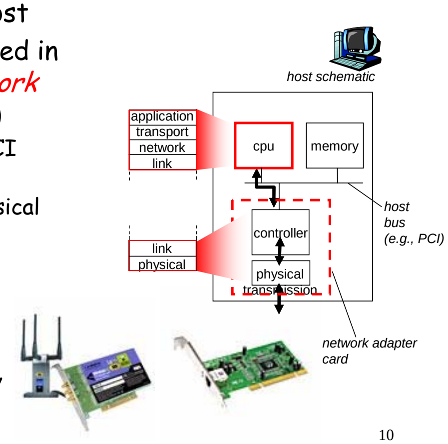
1. Framing, link access:
   - encapsulate datagram into frame, adding header, trailer
   - channel access if shared medium
   - "MAC" addresses used in frame headers to identify source and destination
     - Media Access Control address
     - different from IP address!
2. Reliable delivery between adjacent nodes
   - we learned about this in 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?

Link Layer Services (more)

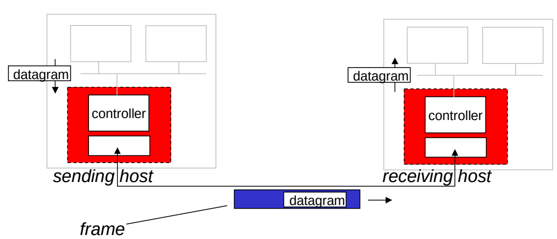
3. *Flow Control:*
   - pacing between adjacent sending and receiving nodes
4. *Error Detection:*
   - errors caused by signal attenuation, noise.
   - receiver detects presence of errors:
     - signals sender for retransmission or drops frame
5. *Error Correction:*
   - receiver identifies *and corrects* bit error(s) without resorting to retransmission
6. *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)
  - Ethernet card, PCMCI card, 802.11 card
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Adaptors Communicating



- sending side:
  - 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

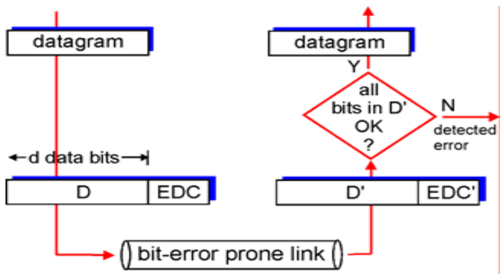
Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM

Error Detection

EDC= Error Detection and Correction bits (redundancy)
D = Data protected by error checking, may include header fields

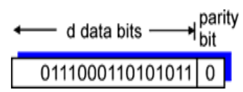
- Error detection not 100% reliable
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity Checking

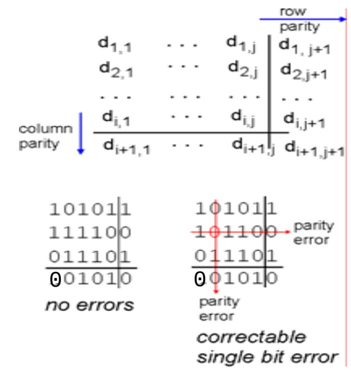
Single Bit Parity:

Detect single bit errors



Two Dimensional Bit Parity:

Detect *and correct* single bit errors



Parity Text Problem, P1

- ❑ Suppose a packet contains 1010101010101011
- ❑ An even parity scheme is used
- ❑ What would the value of the field containing the parity bits be, for the case of a 2D parity scheme?

```
1 0 1 0
1 0 1 0
1 0 1 0
1 0 1 1
```


Parity Text Problem, P2

- ❑ For the previous question, show an example of
  - o 1-bit error detected and corrected
  - o 2-bit error detected but not corrected
    - ▪ Note row 2, columns 2 and 3

```
1 0 1 0
1 1 0 0
1 0 1 0
1 0 1 1
```

Internet checksum

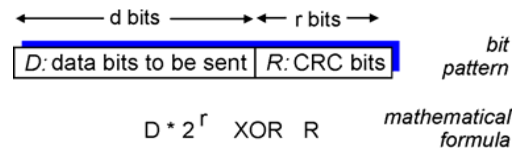
Goal: detect "errors" (e.g., flipped bits) in transmitted segment (note: used at transport layer *only*)

Sender:
- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into checksum field

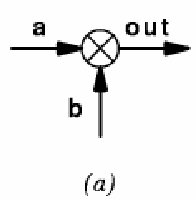
Receiver:
- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?*

Cyclic Redundancy Check

- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), *G*
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by *G* (modulo 2)
  - receiver knows *G*, divides <D,R> 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)



CRC Hardware Circuit



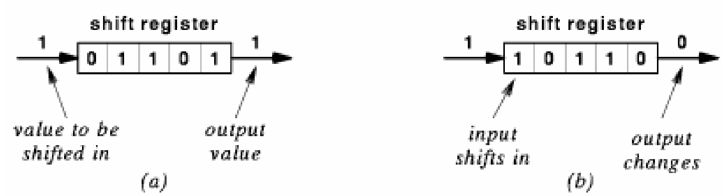
(a)

a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

(b)

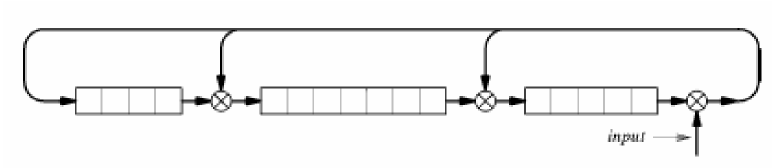
(a) a diagram of hardware that computes an *exclusive or*, and
(b) the output value for each of the four combinations of input values. Such hardware units are used to calculate a CRC.

CRC Hardware Circuit



A shift register
(a) before and
(b) after a shift operation
During a shift, each bit moves right one position, and the output becomes equal to the rightmost bit

CRC Hardware Circuit



A diagram of the hardware used to compute a CRC. After bits of a message have been shifted into the unit, the shift registers contain the *16*-bit CRC for the message.

Typically used to detect changes to a small set of bits near a single location – a 'burst error' (electrical interference, lightening)

Error Detection

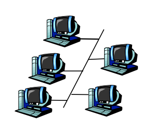
- Parity – typically applied to individual bytes
- Checksum
  - Applied to a packet, a packet header…
  - Is moderately robust
- CRC can detect more errors
  - A single bit of the packet affects the CRC in a more complex manner than for checksum
    - Each bit feeds into the CRC in three places
    - Each bit then cycles through and interacts with remaining bits

Link Layer

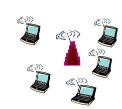
Multiple Access Links and Protocols

Two types of "links":

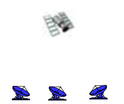
- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - traditional Ethernet
  - 802.11 wireless LAN



shared wire (e.g., cabled Ethernet)



shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

Multiple Access protocols

Single shared broadcast channel

- All nodes receive all frames
- There is 'collision' if more than one node transmits at the same time

Multiple access protocol

- Coordinate access to a shared broadcast channel
- Establish rules for dealing with collisions

Ideal Multiple Access Protocol

Principles for a broadcast channel of rate R

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

MAC Protocols: Three Types

- Channel Partitioning
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- Random Access
  - channel not divided, allow collisions
  - "recover" from collisions
- "Taking turns"
  - Nodes take turns, but nodes with more to send can take longer turns

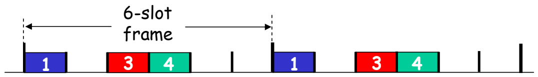
Activity

- Text to be communicated for class, in honor of April 1
- Using
  - TDM, FDM
  - CSMA
  - Token-ring

Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

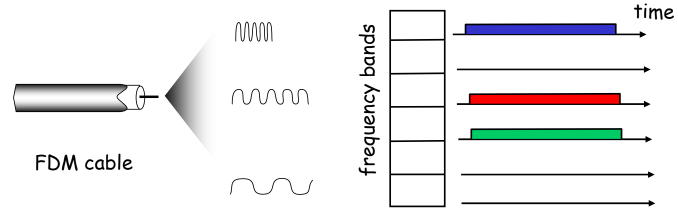
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



Channel Partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes
- two or more transmitting nodes ➜ “collision”,
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

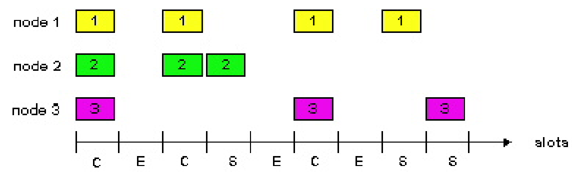
Assumptions

- time is divided into equal size slots, time to transmit 1 frame
- all frames same size
- nodes start to transmit frames only at beginning of slots
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation

- when node obtains fresh frame, it transmits in next slot
- no collision, node can send new frame in next slot
- if collision, node retransmits frame in each subsequent slot with prob. p until success

Slotted ALOHA



Pros
- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons
- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

CSMA (Carrier Sense Multiple Access)

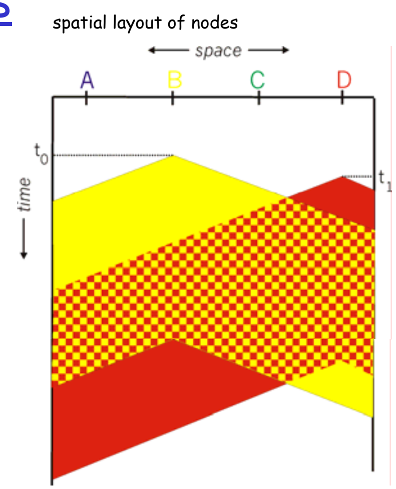
CSMA: listen before transmit:

If channel sensed idle: transmit entire frame

- If channel sensed busy, defer transmission

- Human analogy: don't interrupt others!

CSMA collisions



collisions *can* still occur:
propagation delay means two nodes may not hear each other's transmission

collision:
entire packet transmission time wasted

note:
role of distance & propagation delay in determining collision probability

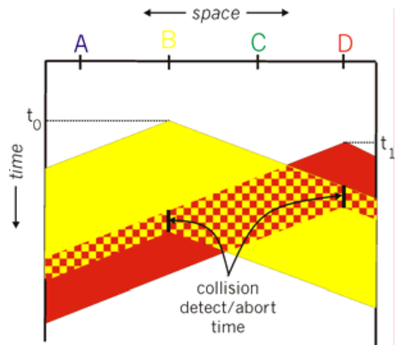
CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting
  - **csma/cd applet:**

http://wps.aw.com/aw_kurose_network_3/0,9212,1406346-,00.html

CSMA/CD collision detection



"Taking Turns" MAC protocols

channel partitioning MAC protocols:

- share channel efficiently and fairly at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

Random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

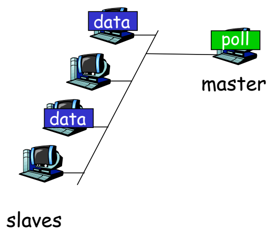
"taking turns" protocols

look for best of both worlds!

"Taking Turns" MAC protocols

Polling:

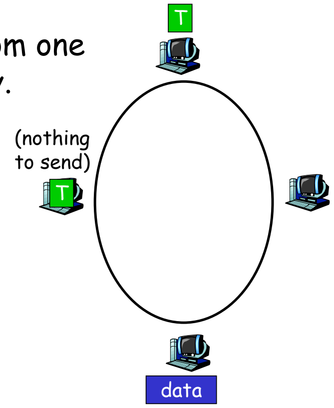
- master node "invites" slave nodes to transmit in turn
- typically used with "dumb" slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



"Taking Turns" MAC protocols

Token passing:

- control **token** passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



Text from Activities

While the origin of April Fools' celebrations is much disputed it is believed that the April fools were people who continued to celebrate the new year on April 1, the ancient start to the new year and the beginning of spring. This tradition continued well into the 1500's when the calendar was changed to reflect January 1, as the official start to the new year. Those individuals who continued to welcome in the new year on April 1, were considered simple minded. As a way to have fun with these people, they were sent invitations to fake parties and given gag gifts. Over time, April 1 evolved into a day of playing tricks and practical jokes on friends and family members.

http://hometown.aol.com/funology1/aprilfools.htm

Summary of MAC protocols

- What do you do with a shared media?
  - Channel Partitioning
    - Partition in time, by frequency, with "codes" …
  - Random access (dynamic),
    - ALOHA
    - carrier sensing: easy in some technologies (wire), hard in others (wireless)
    - CSMA/CD used in Ethernet
    - CSMA/CA used in 802.11, LocalTalk (Apple)
  - Taking Turns
    - Polling
    - Token passing; token ring
    - Bluetooth

Next Class: Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs & switches
- 5.7 PPP