Chapter 5
Link Layer and LANs

A note on the use of these ppt slides:
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:
- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2010
J.F Kurose and K.W. Ross, All Rights Reserved

Computer Networking: A Top Down Approach
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.
Link Layer

5.1 Introduction and services
5.2 Error detection and correction
5.3 Multiple access protocols
  Slotted ALOHA
  Unslotted ALOHA
  CSMA, CSMA/CD

5.4 Link-layer Addressing
5.5 Ethernet
5.6 Link-layer switches
6.3 IEEE 802.11 wireless LANs (“Wi-Fi”)
5.7 PPP
5.8 Link virtualization:
  MPLS
5.9 A day in the life of a web request
MAC Addresses and ARP

- **32-bit IP address:**
  - *network-layer address*
  - used to get datagram to destination IP subnet

- **MAC (or LAN or physical or Ethernet) address:**
  - function: *get frame from one interface to another physically-connected interface (same network)*
  - 48 bit MAC address (for most LANs)
    - burned in NIC ROM, also sometimes software settable
LAN Addresses and ARP

Each adapter on LAN has unique LAN address

Broadcast address = FF-FF-FF-FF-FF-FF

71-65-F7-2B-08-53

58-23-D7-FA-20-B0

0C-C4-11-6F-E3-98

1A-2F-BB-76-09-AD

LAN (wired or wireless)

= adapter
**LAN Address (more)**

- **MAC address allocation administered by IEEE**
- **manufacturer buys portion of MAC address space (to assure uniqueness)**
- **analogy:**
  - (a) **MAC address:** like Social Security Number
  - (b) **IP address:** like postal address
- **MAC flat address ➔ portability**
  - can move LAN card from one LAN to another
- **IP hierarchical address NOT portable**
  - address depends on IP subnet to which node is attached
ARP: Address Resolution Protocol

**Question:** how to determine MAC address of B knowing B’s IP address?

- Each IP node (host, router) on LAN has **ARP table**
- ARP table: IP/MAC address mappings for some LAN nodes
  - < IP address; MAC address; TTL>
    - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)
ARP protocol: Same LAN (network)

- A wants to send datagram to B, and B’s MAC address not in A’s ARP table.
- A broadcasts ARP query packet, containing B’s IP address
  - dest MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B’s) MAC address
  - frame sent to A’s MAC address (unicast)

- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is “plug-and-play”:
  - nodes create their ARP tables without intervention from net administrator
Addressing: routing to another LAN

walkthrough: send datagram from A to B via R.

- focus on addressing - at both IP (datagram) and MAC layer (frame)
- assume A knows B’s IP address
- assume A knows B’s MAC address (how?)
- assume A knows IP address of first hop router, R (how?)
- assume A knows MAC address of first hop router interface (how?)
Addressing: routing to another LAN

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
Addressing: routing to another LAN

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram

A  
111.111.111.111  
74-29-9C-E8-FF-55  
111.111.111.112  
CC-49-DE-D0-AB-7D

R  
222.222.222.220  
1A-23-F9-CD-06-9B

B  
222.222.222.222  
49-BD-D2-C7-56-2A  
222.222.222.221  
88-B2-2F-54-1A-0F
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram
Link Layer

5.1 Introduction and services
5.2 Error detection and correction
5.3 Multiple access protocols
Slotted ALOHA
Unslotted ALOHA
CSMA, CSMA/CD

5.4 Link-layer Addressing
5.6 Link-layer switches
5.5 Ethernet
6.3 IEEE 802.11 wireless LANs (“Wi-Fi”)
5.7 PPP
5.8 Link virtualization: MPLS
5.9 A day in the life of a web request
Hubs

... physical-layer ("dumb") repeaters:
- bits coming in one link go out *all* other links at same rate
- all nodes connected to hub can collide with one another
- no frame buffering
- no CSMA/CD at hub: host NICs detect collisions
Switch

- link-layer device: smarter than hubs, take active role
  - store, forward Ethernet frames
  - examine incoming frame’s MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- transparent
  - hosts are unaware of presence of switches
- plug-and-play, self-learning
  - switches do not need to be configured
Switch: allows *multiple simultaneous transmissions*

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- **switching:** A-to-A’ and B-to-B’ simultaneously, without collisions
  - not possible with dumb hub

*switch with six interfaces (1,2,3,4,5,6)*
Switch Table

- **Q**: how does switch know that A’ reachable via interface 4, B’ reachable via interface 5?
- **A**: each switch has a switch table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!
- **Q**: how are entries created, maintained in switch table?
  - something like a routing protocol?
Switch: self-learning

- switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch “learns” location of sender: incoming LAN segment
  - records sender/location pair in switch table

<table>
<thead>
<tr>
<th>MAC addr</th>
<th>interface</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>60</td>
</tr>
</tbody>
</table>

Switch table *(initially empty)*

Source: A
Dest: A’
Switch: frame filtering/forwarding

When frame received:

1. record link associated with sending host
2. index switch table using MAC dest address
3. if entry found for destination
   then {
     if dest on segment from which frame arrived
       then drop the frame
       else forward the frame on interface indicated
   }
else flood

forward on all but the interface on which the frame arrived
Self-learning, forwarding: example

- frame destination unknown: *flood*
- destination A location known: *selective send*

```
<table>
<thead>
<tr>
<th>MAC addr</th>
<th>interface</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>A'</td>
<td>4</td>
<td>60</td>
</tr>
</tbody>
</table>
```

Source: A
Dest: A'

Switch table (initially empty)
Interconnecting switches

- switches can be connected together

Q: sending from A to G - how does S₁ know to forward frame destined to F via S₄ and S₃?
A: self learning! (works exactly the same as in single-switch case!)
Self-learning multi-switch example

Suppose C sends frame to I, I responds to C

Q: show switch tables and packet forwarding in $S_1$, $S_2$, $S_3$, $S_4$
Institutional network

to external network

router

mail server

web server

IP subnet
Switches vs. Routers

- both store-and-forward devices
  - routers: network-layer devices (examine network-layer headers)
  - switches are link-layer devices (examine link-layer headers)
- routers maintain routing tables, implement routing algorithms
- switches maintain switch tables, implement filtering, learning algorithms
VLANs: motivation

What’s wrong with this picture?

What happens if:

- CS user moves office to EE, but wants connect to CS switch?
- single broadcast domain:
  - all layer-2 broadcast traffic (ARP, DHCP) crosses entire LAN (security/privacy, efficiency issues)
- each lowest level switch has only few ports in use
**VLANs**

**Virtual Local Area Network**

Switch(es) supporting VLAN capabilities can be configured to define multiple *virtual* LANS over single physical LAN infrastructure.

Port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch ......

... operates as *multiple* virtual switches

Electrical Engineering (VLAN ports 1-8)

Computer Science (VLAN ports 9-16)

Data Link Layer 5-27
Port-based VLAN

- **traffic isolation**: frames to/from ports 1-8 can only reach ports 1-8
  - can also define VLAN based on MAC addresses of endpoints, rather than switch port

- **dynamic membership**: ports can be dynamically assigned among VLANs

- **forwarding between VLANs**: done via routing (just as with separate switches)
  - in practice vendors sell combined switches plus routers
**VLANS spanning multiple switches**

- **trunk port**: carries frames between VLANS defined over multiple physical switches
  - frames forwarded within VLAN between switches can’t be vanilla 802.1 frames (must carry VLAN ID info)
  - 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports
802.1Q VLAN frame format

2-byte Tag Protocol Identifier (value: 81-00)

Tag Control Information (12 bit VLAN ID field, 3 bit priority field like IP TOS)

Recomputed CRC
Link Layer

5.1 Introduction and services
5.2 Error detection and correction
5.3 Multiple access protocols
  Slotted ALOHA
  Unslotted ALOHA
  CSMA, CSMA/CD

5.4 Link-layer Addressing
5.5 Ethernet
6.3 IEEE 802.11 wireless LANs (“Wi-Fi”)
5.7 PPP
5.8 Link virtualization: MPLS
5.9 A day in the life of a web request
**Ethernet**

“dominant” wired LAN technology:
- cheap $20 for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps - 10 Gbps

**Metcalfe’s Ethernet sketch**
Star topology

- bus topology popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- today: star topology prevails
  - active switch in center
  - each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)

![Diagram of star topology]
**Ethernet Frame Structure**

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in *Ethernet frame*.

---

**Preamble:**

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- used to synchronize receiver, sender clock rates
Ethernet Frame Structure (more)

- **Addresses**: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame

- **Type**: indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)

- **CRC**: checked at receiver, if error is detected, frame is dropped
Ethernet: Unreliable, connectionless

- **connectionless**: No handshaking between sending and receiving NICs
- **unreliable**: receiving NIC doesn’t send acks or nacks to sending NIC
  - stream of datagrams passed to network layer can have gaps (missing datagrams)
  - gaps will be filled if app is using TCP
  - otherwise, app will see gaps
- Ethernet’s MAC protocol: unslotted CSMA/CD
Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission
   If NIC senses channel busy, waits until channel idle, then transmits
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame!
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters exponential backoff: after $m$th collision, NIC chooses $K$ at random from $\{0, 1, 2, ..., 2^m-1\}$. NIC waits $K\cdot512$ bit times, returns to Step 2
Ethernet’s CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits

Bit time: .1 microsec for 10 Mbps Ethernet; for K=1023, wait time is about 50 msec

Exponential Backoff:
- **Goal**: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from \{0,1\}; delay is K \cdot 512 bit transmission times
- after second collision: choose K from \{0,1,2,3\}...
- after ten collisions, choose K from \{0,1,2,3,4,\ldots,1023\}

See/interact with Java applet on AWL Web site: highly recommended!
CSMA/CD efficiency

- $T_{prop} =$ max prop delay between 2 nodes in LAN
- $t_{trans} =$ time to transmit max-size frame

$$efficiency = \frac{1}{1 + 5 \frac{t_{prop}}{t_{trans}}}$$

- efficiency goes to 1
  - as $t_{prop}$ goes to 0
  - as $t_{trans}$ goes to infinity

- better performance than ALOHA: and simple, cheap, decentralized!
many different Ethernet standards
- common MAC protocol and frame format
- different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
- different physical layer media: fiber, cable
Manchester encoding

- used in 10BaseT
- each bit has a transition
- allows clocks in sending and receiving nodes to synchronize to each other
  - no need for a centralized, global clock among nodes!
- Hey, this is physical-layer stuff!
Link Layer

5.1 Introduction and services
5.2 Error detection and correction
5.3 Multiple access protocols
  Slotted ALOHA
  Unslotted ALOHA
  CSMA, CSMA/CD
5.4 Link-layer Addressing
5.5 Ethernet
  6.3 IEEE 802.11 wireless LANs (“Wi-Fi”)
5.6 Link-layer switches
5.7 PPP
5.8 Link virtualization: MPLS
5.9 A day in the life of a web request
Wireless network characteristics

Multiple wireless senders and receivers create additional problems (beyond multiple access):

Hidden terminal problem
- B, A hear each other
- B, C hear each other
- A, C can not hear each other

This means A, C unaware of their interference at B.

Signal attenuation:
- B, A hear each other
- B, C hear each other
- A, C can not hear each other

interfering at B.
IEEE 802.11 Wireless LAN

- **802.11b**
  - 2.4-5 GHz unlicensed spectrum
  - up to 11 Mbps
  - direct sequence spread spectrum (DSSS) in physical layer
    - all hosts use same chipping code

- **802.11a**
  - 5-6 GHz range
  - up to 54 Mbps

- **802.11g**
  - 2.4-5 GHz range
  - up to 54 Mbps

- **802.11n**: multiple antennae
  - 2.4-5 GHz range
  - up to 200 Mbps

- all use CSMA/CA for multiple access
- all have base-station and ad-hoc network versions
802.11 LAN architecture

- Wireless host communicates with base station
  - Base station = access point (AP)
- Basic Service Set (BSS) (aka “cell”) in infrastructure mode contains:
  - Wireless hosts
  - Access point (AP): base station
  - Ad hoc mode: hosts only
802.11: Channels, association

- 802.11b: 2.4GHz-2.485GHz spectrum divided into 11 channels at different frequencies
  - AP admin chooses frequency for AP
  - interference possible: channel can be same as that chosen by neighboring AP!

- host: must **associate** with an AP
  - scans channels, listening for **beacon frames** containing AP’s name (SSID) and MAC address
  - selects AP to associate with
  - may perform authentication [Chapter 8]
  - will typically run DHCP to get IP address in AP’s subnet
802.11: passive/active scanning

Passive Scanning:
1. beacon frames sent from APs
2. association Request frame sent: H1 to selected AP
3. association Response frame sent: selected AP to H1

Active Scanning:
1. Probe Request frame broadcast from H1
2. Probes response frame sent from APs
3. Association Request frame sent: H1 to selected AP
4. Association Response frame sent: selected AP to H1
IEEE 802.11: multiple access

- Avoid collisions: 2+ nodes transmitting at the same time
- 802.11: CSMA - sense before transmitting
  - Don’t collide with ongoing transmission by other node
- 802.11: No collision detection!
  - Difficult to receive (sense collisions) when transmitting due to weak received signals (fading)
  - Can’t sense all collisions in any case: hidden terminal, fading
  - Goal: Avoid collisions: CSMA/C(ollision)A(voidance)
IEEE 802.11 MAC Protocol: CSMA/CA

802.11 sender
1 if sense channel idle for DIFS then
   transmit entire frame (no CD)
2 if sense channel busy then
   start random backoff time
   timer counts down while channel idle
   transmit when timer expires
   if no ACK, increase random backoff
   interval, repeat 2

802.11 receiver
- if frame received OK
  return ACK after SIFS (ACK needed due to hidden terminal problem)
Avoiding collisions (more)

**idea:** allow sender to “reserve” channel rather than random access of data frames: avoid collisions of long data frames

- sender first transmits *small* request-to-send (RTS) packets to BS using CSMA
  - RTSs may still collide with each other (but they’re short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions

avoid data frame collisions completely using small reservation packets!
Collision Avoidance: RTS-CTS exchange

Reservation collision

defer
### 802.11 Frame: Addressing

A 802.11 frame consists of several components, each serving a specific function. Here is a breakdown of the frame components:

- **Frame Control**: 2 bits, indicating the type of frame.
- **Duration**: 2 bits, indicating the duration of the frame.
- **Address 1**: 6 bits, the MAC address of the wireless host or AP transmitting the frame.
- **Address 2**: 6 bits, the MAC address of the wireless host or AP to receive the frame.
- **Address 3**: 6 bits, the MAC address of the router interface to which the AP is attached.
- **Sequence Control**: 2 bits, indicating the sequence control.
- **Address 4**: 6 bits, used only in ad hoc mode.
- **Payload**: 0-2312 bits, the data payload of the frame.
- **CRC**: 4 bits, the cyclic redundancy check for error detection.

**Addressing Details**:

- **Address 1**: MAC address of the wireless host or AP transmitting the frame.
- **Address 2**: MAC address of the wireless host or AP to receive this frame.
- **Address 3**: MAC address of the router interface to which the AP is attached.
- **Address 4**: Used only in ad hoc mode.

---

Wireless, Mobile Networks 6-52
802.11 frame: addressing

802.11 frame: addressing

802.3 frame

802.11 frame

AP MAC addr | H1 MAC addr | R1 MAC addr
--- | --- | ---
address 1 | address 2 | address 3

Internet

R1 router

H1

AP

802.11 frame: addressing

Wireless, Mobile Networks 6-53
802.11 frame: more

frame seq #
(for RDT)

duration of reserved
transmission time (RTS/CTS)

frame control
duration
address 1
address 2
address 3
seq control
address 4
payload
CRC

Protocol version
Type
Subtype
To AP
From AP
More frag
Retry
Power mgt
More data
WEP
Rsld

frame type
(RTS, CTS, ACK, data)
H1 remains in same IP subnet: IP address can remain same

switch: which AP is associated with H1?

- self-learning (Ch. 5): switch will see frame from H1 and “remember” which switch port can be used to reach H1
802.11: advanced capabilities

Rate Adaptation

- base station, mobile dynamically change transmission rate (physical layer modulation technique) as mobile moves, SNR varies

1. SNR decreases, BER increase as node moves away from base station
2. When BER becomes too high, switch to lower transmission rate but with lower BER
802.11: advanced capabilities

Power Management

- node-to-AP: “I am going to sleep until next beacon frame”
  - AP knows not to transmit frames to this node
  - node wakes up before next beacon frame
- beacon frame: contains list of mobiles with AP-to-mobile frames waiting to be sent
  - node will stay awake if AP-to-mobile frames to be sent; otherwise sleep again until next beacon frame
802.15: personal area network

- less than 10 m diameter
- replacement for cables (mouse, keyboard, headphones)
- ad hoc: no infrastructure
- master/slaves:
  - slaves request permission to send (to master)
  - master grants requests
- 802.15: evolved from Bluetooth specification
  - 2.4-2.5 GHz radio band
  - up to 721 kbps
802.16: WiMAX

- like 802.11 & cellular: base station model
  - transmissions to/from base station by hosts with omnidirectional antenna
  - base station-to-base station backhaul with point-to-point antenna

- unlike 802.11:
  - range ~ 6 miles ("city rather than coffee shop")
  - ~14 Mbps
802.16: WiMAX: downlink, uplink scheduling

- transmission frame
  - down-link subframe: base station to node
  - uplink subframe: node to base station

base station tells nodes who will get to receive (DL map) and who will get to send (UL map), and when

- WiMAX standard provide mechanism for scheduling, but not scheduling algorithm