# Ethernet (Classic)
## (CSE 473S – Fall 2009)

Ken Wong
Washington University

kenw@arl.wustl.edu
www.arl.wustl.edu/~kenw

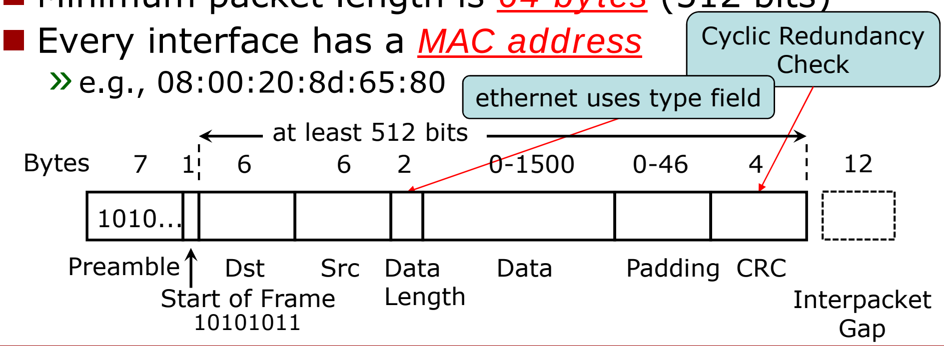
Washington University in St.Louis

## Ethernet Local Area Network

- The most prevalent wired LAN technology
  - » simple system administration (plug and play)
- Size
  - » restricted (several km, 100s of stations) ➔
    - worst-case transmission time is bounded and known in advance
- Transmission technology
  - » broadcast (cable, hub) or point-to-point (switch)
  - » high bandwidth
    - 10 Mbps, 100 Mbps, 1 Gbps, now 10 Gbps
  - » low delay, low error rate
  - » various media
    - coaxial cable, copper wire, fiber

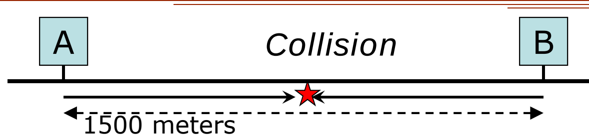
# IEEE 802.3

- IEEE 802.3
  - » specifies a family of *1-persistent CSMA/CD* with *truncated binary exponential backoff* running at speeds from 1 Mbps to 10 Gbps on various media
- Ethernet
  - » A specific implementation of IEEE 802.3
- Minimum packet length is *64 bytes* (512 bits)
- Every interface has a *MAC address*
  - » e.g., 08:00:20:8d:65:80





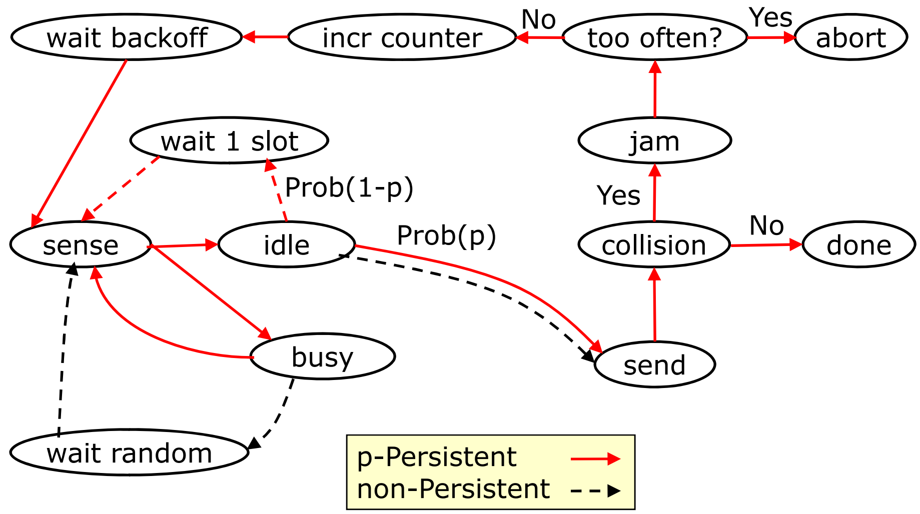

# Shared Medium (Cable, Hub)




- ? Two senders begin transmitting at the same time
  - » propagation speed = 200 m/usec ( meters per microsecond )
    - 2/3 speed of light in vacuum ( 3 x $10^8$ meters per sec )
  - » first bit of each frame collides midway (750 m) ➔ collision at time 3.75 usec ( = 750 m/(200 m/usec) )
  - » collision causes a noisy signal which is detected 7.5 usec after beginning of transmission
- ? How does A's ethernet interface detect a collision
  - » wait for the worst-case delay of noise burst
    - host at end of cable sends pkt which collides at the other end
    - for 500 meter cable: 5.0 usec

# CSMA Protocols

ethernet: 1-Persistent, CSMA/CD with
truncated binary exponential backoff






# CSMA Protocols

- IEEE 802.3
  - a family of 1-persistent, CSMA/CD with truncated binary exponential backoff
- *Carrier Sense Protocol*
  - stations listen for a carrier (transmission) before transmitting
  - implies minimum frame transmission time requirement
    - frame transmission time > max round-trip propagation delay
- *Persistent CSMA*
  - station waits for the medium to become idle
- *Nonpersistent CSMA*
  - busy medium
    - station waits a random time interval and retries
  - effect
    - randomly probes the medium until it is idle

# Persistent CSMA

- p-persistence (slotted channels)
  - » when channel becomes idle, either:
    - send a frame with probability p; or
    - wait one time slot with probability 1-p before repeating process
- Pr [2 waiting stations will cause a collision] = $p^2$
  - » when p = 1, 2 waiting stations are guaranteed to collide on their first retry
- Choice of p
  - » a tradeoff between
    - performance under heavy load, and
    - mean message delay
  - » if there are n waiting stations, then
    - mean number of simultaneous sends is equal to *np*
  - » want np < 1



# Exponential Backoff

- Sender
  - » if collision occurs during transmission
  - » send a 32-bit *jamming signal*
  - » wait W time slots
    - choose W equiprobably from 0 to $2^B$-1
  - » increment the backoff count B with B = min(n, 10)
    - where n = number of successive collisions
- Backoff at most 15 times
  - » i.e., B = 1, 2, ... , 9, 10, 10, 10, 10, 10, 10
- One time slot = *512 bit-times*
  - » = max round-trip propagation delay when there are 5 segments (and 4 repeaters)
- Backoff time
  - » W x 512 bit-times (for 10 Mbps, = W x 51.2 usec)
    - *W is an equiprobable random variable between 0 and $2^B$-1*

# 10 Mbps Media Options

| Name | Cable | Max Segment | Nodes/ Segment | Advantages |
|---|---|---|---|---|
| 10Base5 | Thick Coax | 500 m | 100 | Good for backbones |
| 10Base2 | Thin Coax | 200 m | 30 | Cheap |
| 10Base-T | Twisted Pair | 100 m | 1024 | Easy maintenance |
| 10Base-F | Fiber Optics | 2000 m | 1024 | Between buildings |

- Nomenclature: xBASEy (e.g., 10BASE5)
  - x indicates network data rate in Mbps (e.g., 10 Mbps)
  - y indicates maximum segment length in 100 meters
    - e.g., 500 meters
  - Base indicates *baseband* signaling (only carries Ethernet)
- *Attenuation*
  - Signal loses strength as it travels through a lossy medium



# Ethernet Adaptor

- Receiver looks at all frames and accepts all frames:
  - addressed to its own address
  - addressed to the broadcast address ff:ff:ff:ff:ff:ff (hexadecimal)
  - addressed to a multicast address
    - 01:00:5e:00:00:00 – 01:00:5e:7f:ff:ff
  - if it is placed in *promiscuous mode*
- Sender does most of the work
  - listens for idle media
  - transmits frame
  - listens for collision
  - transmits jamming signal when it detects a collision
  - implements exponential backoff algorithm

# Higher Bandwdith Ethernet

- Summary of 10Base5 (10 Mbps, 500 meter)
  - Max distance between any 2 hosts = 2500 meters
  - Minimum Frame Size = 64 bytes = 512 bits
  - Slot Time = 512 bits
  - Interbit Time = 100 ns
- *Naive scaling* to 100 Mbps and 1 Gbps:

| Property | 10 Mbps | 100 Mbps | 1 Gbps |
|---|---|---|---|
| Inter-Packet Gap (bits) | 96 | 96 | 96 |
| Interbit Time (nsec) | 100 | 10 | 1 |
| Min Frame Size (bits) | 512 | 512 | 512 |
| Max Distance Between 2 Hosts (m) | 2500 | 250 | 25 |



# Repeaters, Bridges, Routers

- *Repeater*: Physical layer (OSI layer 1)
  - Signal restoration, Preamble restoration, Collision detection, Lockup protection, Delay and inter-packet gap (IPG) shrinkage
  - Restore frame and broadcast on all output ports
- *Bridge*: MAC (Media Access Control) or Link layer (OSI layer 2)
  - Filtering, Learning, Forwarding
  - Store and forward frame to port closest to destination
- *Router*: Network layer (OSI layer 3)
  - Store and forward packet to port closest to destination
- *Hub*: A modern multiport repeater
- *Ethernet Switch*: A modern hardware bridge
  - Concurrent, full duplex, auto-sensing, multiport operation

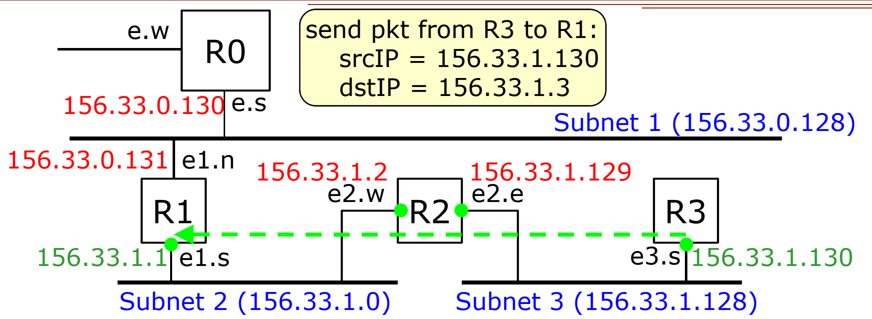
## Ethernet and the Address Resolution Protocol



# Abstract Routing Algorithm

```
Route( Datagram dgram, RouteTbl rt )
        dstIP = Extract destination IP address from dgram;
        Find best matching entry in rt;
        if( no match )          { Routing Error; }
        if( matches a directly connected network address )
                { nxtHopIP = dstIP; }
        if( matches router interface entry )
                { nxtHopIP = From matching rt entry; }
        Physical Address = Resolve( nxtHopIP );   // ARP
        I = Outgoing interface from matching rt entry;
        Encapsulate and Send dgram over interface I;
}
```

# ARP Example




- R3 needs *mac(156.33.1.129)*, the next hop interface
  - R3 *broadcasts* ARP request to find out mac(156.33.1.129)
  - R2/e2.e sends ARP reply to R3/e3.s (*unicast*, not a broadcast)
  - Now, R3 knows the binding of 156.33.1.130 to e2.e!
- Alternative: *Gratuitous ARP*
  - During boot process, every host sends an ARP request for its own IP address
    - ➔ Effectively announces its own IPaddr-to-MACaddr binding



# Example Routing Tables

9C 21 00 80

FF FF FF 80

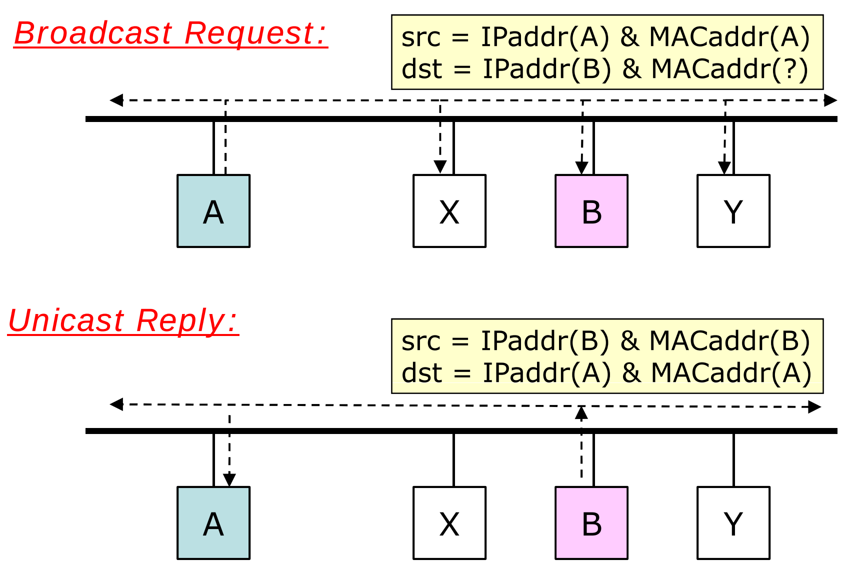
| Entry | Network | Net Mask | Next Hop | Interface | Note |
|---|---|---|---|---|---|
| R0[0] | 156.33.0.128 | 255.255.255.128 | DIRECT | e.s | Subnet 1 |
| R0[1] | 156.33.1.0 | 255.255.255.128 | 156.33.0.131 | e.s | Subnet 2 |
| R0[2] | 156.33.1.128 | 255.255.255.128 | 156.33.0.131 | e.s | Subnet 3 |
| R0[3] | 0.0.0.0 | 0.0.0.0 | Internet | e.w | Internet |
| R1[0] | 156.33.0.128 | 255.255.255.128 | DIRECT | e1.n | Subnet 1 |
| R1[1] | 156.33.1.0 | 255.255.255.128 | DIRECT | e1.s | Subnet 2 |
| R1[2] | 156.33.1.128 | 255.255.255.128 | 156.33.1.2 | e1.s | Subnet 3 |
| R1[3] | 0.0.0.0 | 0.0.0.0 | 156.33.0.130 | e1.n | Default |
| R2[0] | 156.33.1.0 | 255.255.255.128 | DIRECT | e2.w | Subnet 2 |
| R2[1] | 156.33.1.128 | 255.255.255.128 | DIRECT | e2.e | Subnet 3 |
| R2[2] | 0.0.0.0 | 0.0.0.0 | 156.33.1.1 | e2.w | Default |
| R3[0] | 156.33.1.128 | 255.255.255.128 | DIRECT | e3.s | Subnet 3 |
| R3[1] | 0.0.0.0 | 0.0.0.0 | 156.33.1.129 | e3.s | Default |

# Routing Algorithm Details

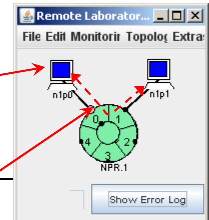
- Match ?
  - ((dst IP address) && netmask) == network address in route table
- Idea
  - allow arbitrary netmasks ➔ handle special cases in general way
  - special cases: default route, host-specific route
- Route to a specific host
  - netmask = 255.255.255.255
- Default route
  - netmask = 0.0.0.0



# Address Resolution Protocol (ARP)



# ONL ARP Cache Example




```
onl022> arp -n

Address       HWtype  HWaddress          Flags Mask   Iface
10.0.1.2      ether   00:0B:DB:70:97:E4    C          eth0
10.0.1.3      ether   00:0B:DB:70:9A:76    C          eth0

onl022> ping -c 3 n1p1
. . . output deleted . . .

onl022> arp -n

Address       HWtype  HWaddress          Flags Mask Iface
10.0.1.2      ether   00:0B:DB:70:97:E4    C        eth0
192.168.1.31  ether   00:00:50:33:13:05    C        eth1
10.0.1.3      ether   00:0B:DB:70:9A:76    C        eth0
```



# Ethernet ARP Implementation

- Request for binding (IPaddr → MACaddr):
  - search ARP cache
  - broadcast ARP request and wait for reply
    - broadcast has MACaddr and IPaddr of sender and IPaddr of destination
    - reply can be delayed (busy host) or never received (down host)
    - buffer outgoing packet that triggered ARP request
    - release buffer when reply is returned or a timeout occurs
    - handle ALL outstanding ARP requests for the same destination
    - stale ARP cache value (age cached values; i.e., soft state)
  - update ARP cache
  - process packets waiting for IPaddr → MACaddr binding
- Entire subnet reads IPaddr → MACaddr request
  - cache broadcaster's IPaddr → MACaddr mapping
  - send ARP reply message to broadcaster if receiver is the ARP target

# ARP Implementation Issues

- Target host may be down or too busy to accept request
- Request can be lost because Ethernet provides a best-effort service
- Stale ARP cache entry
  - » e.g., host ethernet interface is replaced
  - » cache entry has soft state
    - i.e., entry is removed if timer expires
- Optimizations
  - » Address Resolution Cache (Cache IPaddr → MACaddr mappings)
  - » piggyback broadcaster's IPaddr-MACaddr binding onto the broadcast message
  - » all hosts on the broadcast network can cache the broadcaster's Ipaddr → MACaddr binding



# ARP Message Format

| Hardware Type | | Protocol Type |
|---|---|---|
| Hlen | Plen | Operation |
| Sender HA | | |
| Sender HA | | Sender IP |
| Sender IP | | Target HA |
| Target HA | | |
| Tartget IA | | |

- Encapsulated in Ethernet frame

# ARP Protocol Format

- No fixed format for ARP messages; depends on network technology
- Header indicates field lengths
- Ethernet ARP/RARP Message Format
  - » Hardware Type (1 ➔ Ethernet)
  - » Protocol Type (x0800 ➔ High-level addresses are in IP format)
  - » Hlen: Hardware address length
  - » Plen: Protocol address length
  - » Operation: (1) Request or (2) Reply
  - » Sender HA, IP: Sender's hardware and IP addresses
  - » Target HA, IP: Target's hardware and IP addresses
- ARP requestor supplies Sender HA, IP, and Target IP
- Replier fills in Target HA; swaps Sender and Target