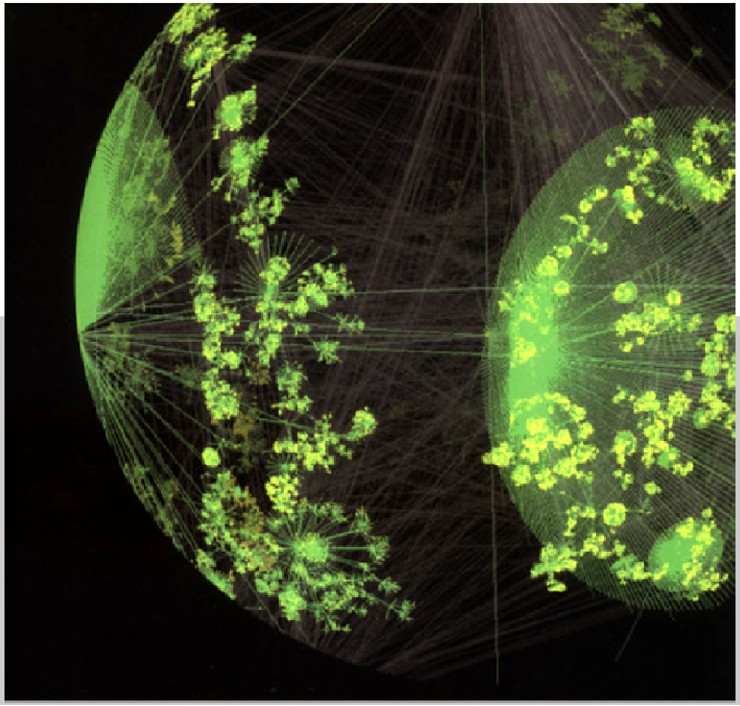
Chapter 2
A Single Segment Network – Data Link Layer



TCP/IP Essentials

A Lab-Based Approach

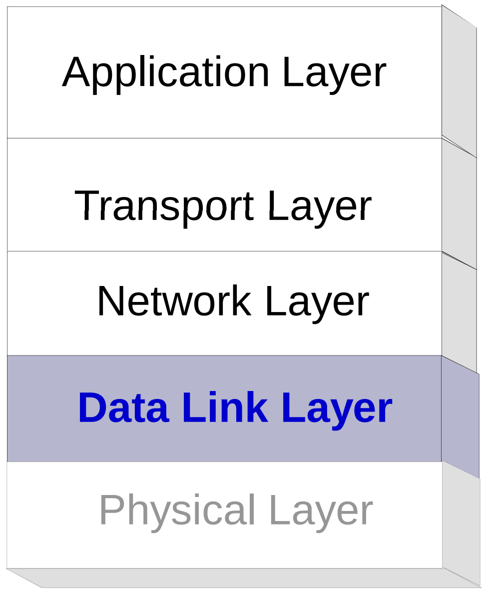
Fall 2015

Data Link Layer

Main tasks of the data link layer:

- Transfer network layer data from one machine to another machine via “a data link”.
- Convert the data between raw bit stream of the physical layer and groups of bits – frames.
- Perform flow control between sender and receiver.

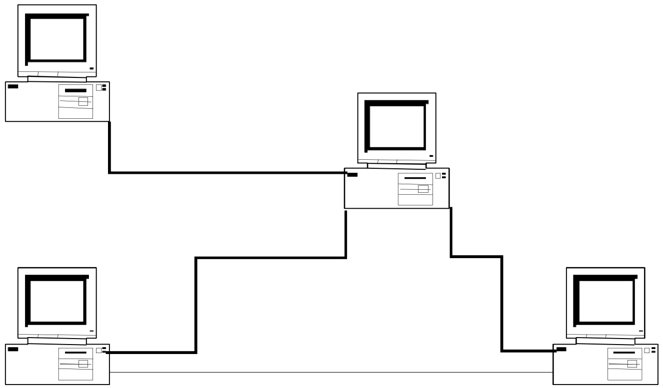
TCP/IP Suite



Types of Networks

Point-to-point network

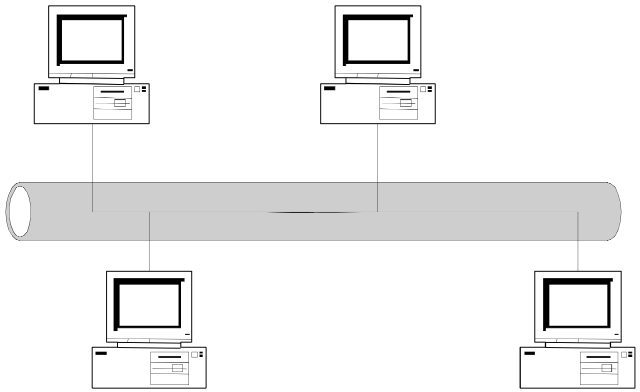
- Each link connects two end points: hosts or any network devices
- Usually for long distance connections
- Examples: DSL, POS (Packet over SONET/SDH), GbE (Gigabit Ethernet)



Point-to-Point Network

Broadcast network

- A number of stations share a common transmission medium
- Usually for local networks
- Examples: CSMA/CD Ethernet, WLAN (Wireless Local Area Network), a.k.a. Wi-Fi



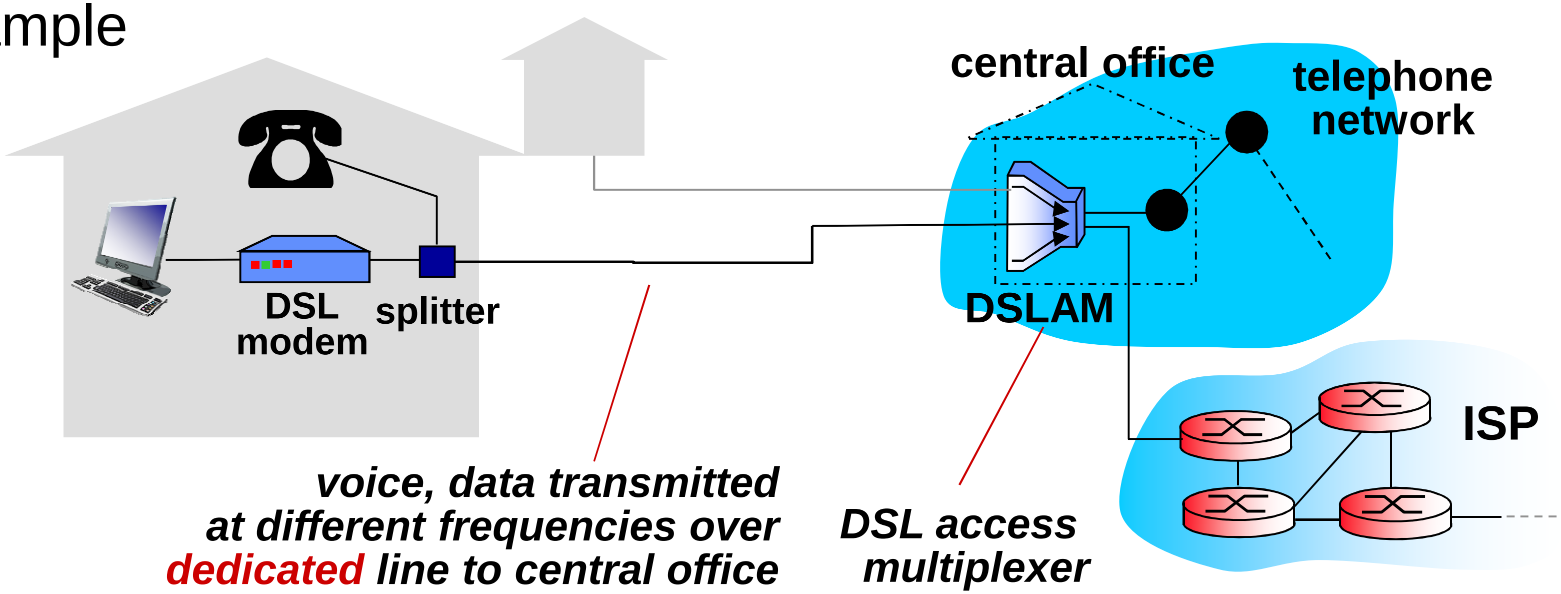
Broadcast Network

Multiple Access

- Network topology
  - Point-to-point → N(N-1)/2 links to connect N nodes
  - Broadcast → the shared medium forms a single domain
- Medium access control (MAC) protocols
  - Rules to share a medium
  - Carrier Sense Multiple Access/Collision Detection (CSMA/CD)
  - Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)

Point-to-Point Protocol
– point-to-point network example

- The Point-to-Point Protocol (PPP) is a data link protocol
- The main purpose of PPP is encapsulation and transmission of IP datagrams, or other network layer protocol data, over a serial link.
- Currently, PPP is used by most dial-up Internet access, Digital Subscriber Loop (DSL), and cable broadband services.
- DSL example



PPP Encapsulation

PPP frame format

- Flag: mark the beginning and ending of a frame
- Protocol: used to multiplex different protocol data
- No addressing, only two end hosts.

Flag	Addr.	Ctrl.	Protocol	Data	CRC	Flag
7E	FF	03				7E
1 byte	1 byte	1 byte	2 bytes	≤ 1500 bytes	2 bytes	1 byte

0021	IP Datagram
C021	Link Control Data
8021	Network Control Data

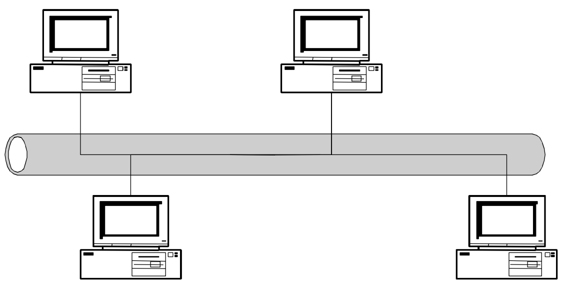
Point-to-Point Protocol (RFC 1661)

PPP consists of two types of control protocols:

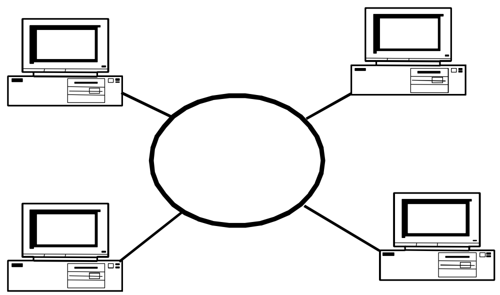
- Link Control Protocol (LCP)
  - Responsible for agreeing on PPP encapsulation options, packet size limits, and detecting common mis-configuration errors over the data link
  - Optional features to provide peer authentication, detect link status
- Network Control Protocol (NCP)
  - PPP supports a family of NCPs and treat each network protocol like an interface
  - IP Control Protocol (IPCP, RFC 1332), used for configure the link to transmit IP datagrams

Local Area Networks

- Local Area Networks (LANs) typically connect computers within a building or a campus.
- Many LANs are broadcast networks.
- Bus and Ring are two typical LAN topologies used in early days
- The protocol that determines who can transmit on a broadcast channel is called Medium Access Control (MAC) protocol.



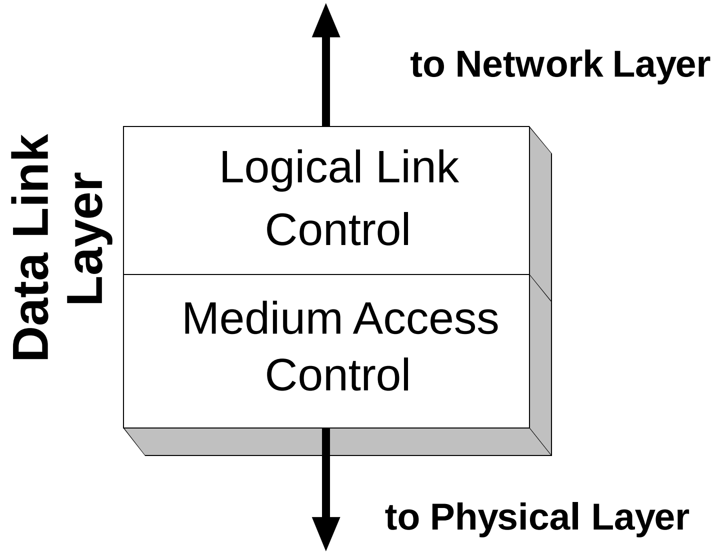
Bus LAN



Ring LAN

MAC and LLC

- In any broadcast network, the stations must ensure that only one station transmits at a time on the shared communication channel.
- The protocol that determines who can transmit on a broadcast channel is called Medium Access Control (MAC) protocol.
- The MAC protocol is implemented in the MAC sublayer which is the lower sublayer of the data link layer.
- The higher portion of the data link layer is often called Logical Link Control (LLC).



Logical Link Control

- LLC can provide different services to the network layer:
  - unacknowledged connectionless service
  - acknowledged connectionless service
  - connection-oriented service
- Framing
- Error control
- Addressing

Media Access Control

MAC algorithms are used to resolve collisions and share the medium in a broadcast network.

Examples of MAC:

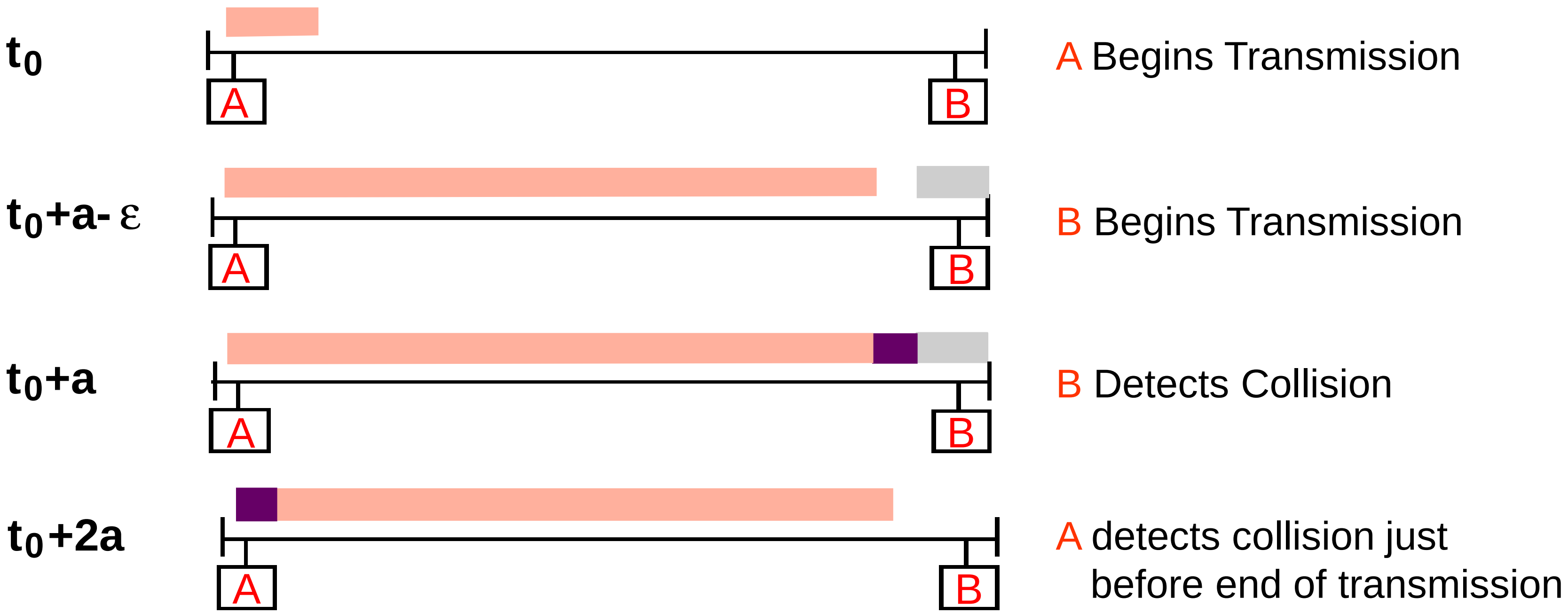
- ALOHA
- Carrier Sense Multiple Access/Collision Detection (CSMA/CD)
- Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)

Ethernet

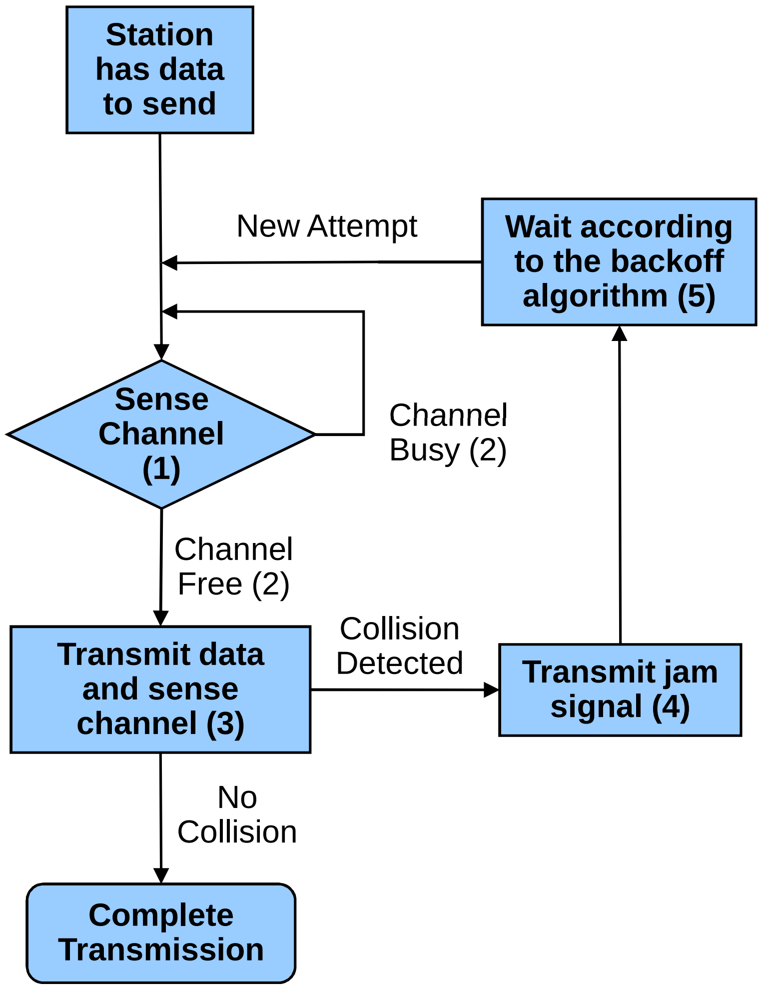
- Originally developed at Xerox PARC at 2.94 Mbps data rate over coaxial cable based on CSMA/CD, 1972 – 1977
- An industry standard since 1982 (DIX Ethernet), or
- Based on CSMA/CD adopted by IEEE 802 committee in 1985
  - Interoperates with 802.2 (LLC) as higher layer
  - Uses different encapsulation header to carry data payload

Collisions in Ethernet

- The collision resolution process of Ethernet requires that a collision is detected while a station is still transmitting.
- Assume the maximum propagation delay on the bus is *a*.
- Restrictions: Each frame should be at least twice as long as the time to detect a collision (*2a*).



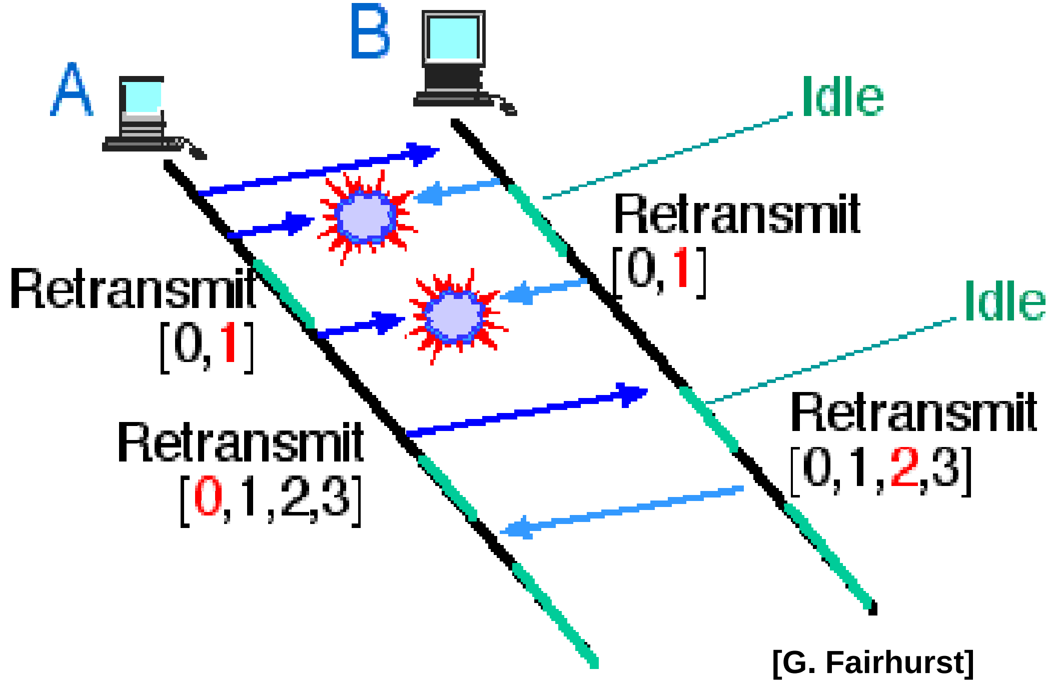
CSMA/CD Flow Diagram



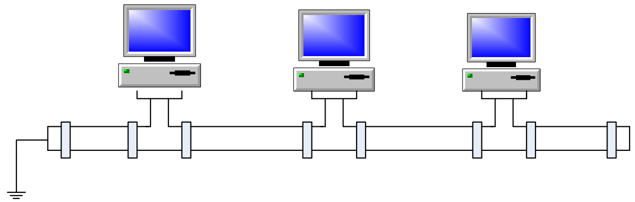
1. Each station listens before it transmits.
2. If the channel is busy, it waits until the channel goes idle, and then it transmits.
3. If the channel is idle it transmits immediately and continue sensing for *2a* seconds.
4. If collision is detected, transmit a brief jamming signal then cease transmission.
5. Wait for a random time, and retransmit. The random time is determined by exponential backoff algorithm.

Exponential Backoff Algorithm

- If a station is involved in a collision, it waits a random amount of time before attempting a retransmission.
- The random time is determined by the following algorithm:
  - Set “slot time” to *2a*.
  - After first collision wait *0* or *1* time unit.
  - After the i^{th} collision, wait a random number between *0* and 2^i*-1* time slots.
  - Do not increase random number range if *i>9*.
  - Give up after *16* collisions.

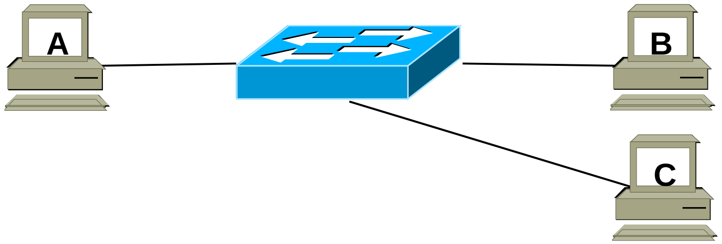


Ethernet Switches



In an Ethernet LAN, hosts can be

- Attached to a common cable, or
- Connected by *Ethernet switches*.



Ethernet switches are MAC layer devices that switch frames between ports connected to different LAN segments.

- Offer guaranteed bandwidth for segments.
- Separate a LAN into collision domains.

Ethernet Encapsulation (RFC 894)

Dest. Addr	Src. Addr.	Type	Data	CRC
6	6	2	46-1500	4

Type 0800	IP datagram
2	46-1500

Type 0806	ARP req./reply	PAD
2	28	18

Type 8035	RARP req./reply	PAD
2	28	18

- **Dest. Addr., Src. Addr.:** MAC addresses are 48 bit
- **Type:** Identifies the content of the data field (must ≥ Ox0600)
- **CRC:** cyclic redundancy check

IEEE 802.2/802.3 Encapsulation

←802.3 MAC →←802.2 LLC →←802.2 SNAP →

Destination address	Source address	Length	DSAP AA	SSAP AA	cntl 03	Org code 0	type	Data	CRC
6	6	2	1	1	1	3	2	38-1492	4

- Destination address, Source address:
  MAC addresses are 48 bit (displayed as 12 hexadecimal characters)
- Length: frame length in number of bytes (<Ox0600, 1,500 bytes → Ox05dc)
- DSAP, SSAP: always set to 0xaa
- Ctrl: set to 3
- Org code: set to 0
- Type field: identifies the content of the data field
- CRC: cyclic redundancy check

0800	IP datagram
2	38-1492

0806	ARP request/reply	PAD
2	28	10

8035	RARP request/reply	PAD
2	28	10

Total frame size: 64 bytes to 1518 bytes

Overhead: 38 bytes including 12 bytes Inter Frame Gap (IFG)

IEEE 802.11 Wireless LANs

Alternative to the wired Ethernet:

- Wireless channel
- Frequency band: unlicensed radio spectrums

Protocols:

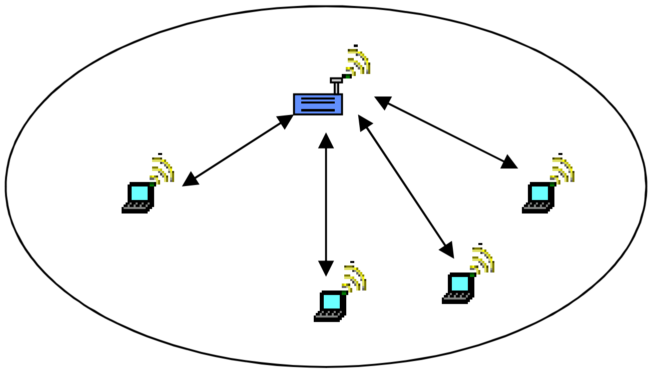
- IEEE 802.11b: 5, 11Mbps channel speed, 2.4GHz frequency band
- IEEE 802.11a: 6, 9, 12, 18, 24, 36, 48, 54Mbps, 5GHz frequency band
- IEEE 802.11g: 54 Mbps, 2.4GHz band
- **IEEE 802.11i**: security
- IEEE 802.11f: Inter Access Point Protocol
- **IEEE 802.11e**: Quality of Service enhancement, …, video optimized
- **IEEE 802.11n**: data rate great than 100 Mbps using MIMO, 2.4G and/or 5GHz bands
- **IEEE 802.11ac**: single stream up to 433 Mbps, more spatial streams, 5GHz band

Range: Transmission power up to 100mW

- Indoor: 20 - 25 meters
- Outdoor: 50 - 100 meters

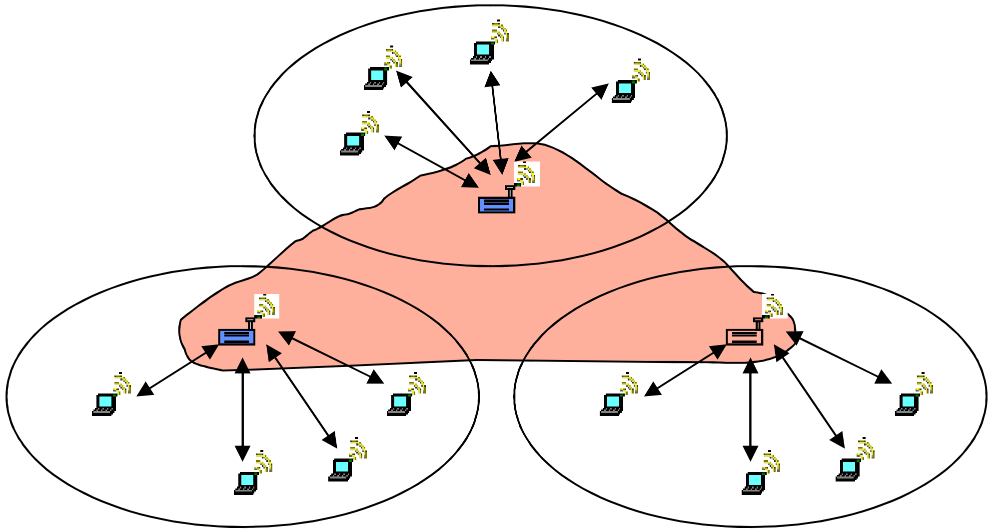
IEEE 802.11 Architecture

Basic Service Set (BSS)



Extended Service Set (ESS)

a.k.a. Infrastructure Mode



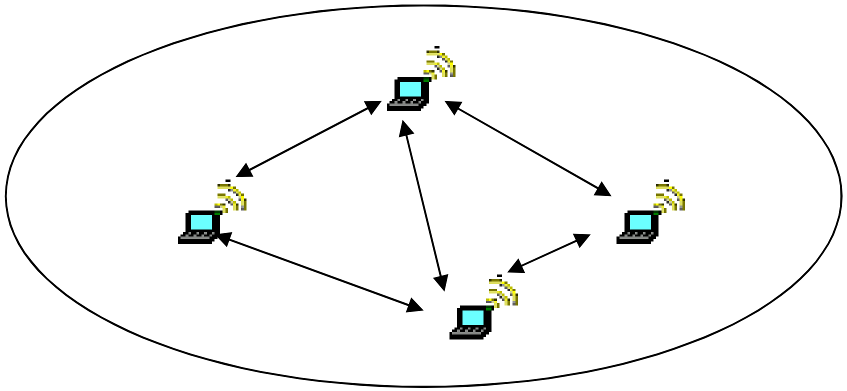
Infrastructure mode

- Fixed Access Point (AP) provides:
  - – Connection to wireline network
  - – Relay function
- Handoff, an active host moves from one access point to another.

IEEE 802.11 Architecture (cont'd)

The ad hoc mode, a.k.a. Independent BSS

- No access point.
- Hosts communicate with each other directly.



IEEE 802.11 Frame Format

Frame Control	Duration	MAC Address 1	MAC Address 2	BSS ID	Sequence Control	MAC Address 4	Data	CRC
2	2	6	6	6	2	6	0 ~ 2312	4

- More fields than other data-link protocols
- High overhead:
  - 30 byte header
  - Four Address fields: BSSID, Source Address, Destination Address, Receiving Station Address, Transmitting station Address depend on Frame Control setting
- Different frame types for different tasks:
  - Some fields are not presented in all types of frames

802.11 MAC Addresses

MAC header contains up to 4 MAC addresses:

- MAC addresses are globally unique IDs assigned by manufacturer to any network interface card (NIC).
- Several IEEE formats: MAC-48, EUI-48, EUI-64

Addresses:

- RA, TA = receiving and transmitting MAC addresses
- BSSID = ID of basic service set of the transmission
- DA, SA = end-to-end destination and source MAC address
  - May be different from RA, TA in multi-hop transmission across a Distribution System
  - Rarely used.

Scenario		To DS	From DS	Addr 1	Addr 2	Addr 3	Addr 4
IBSS (Ad Hoc network)		0	0	RA/DA	TA/SA	BSSID	N/A
Infrastructure network	AP to STA	0	1	RA/DA	BSSID	TA/SA	N/A
	STA to AP	1	0	BSSID	TA/SA	RA/DA	N/A
	AP to AP	1	1	RA	TA	DA	SA

CSMA/CA

CSMA/CA:

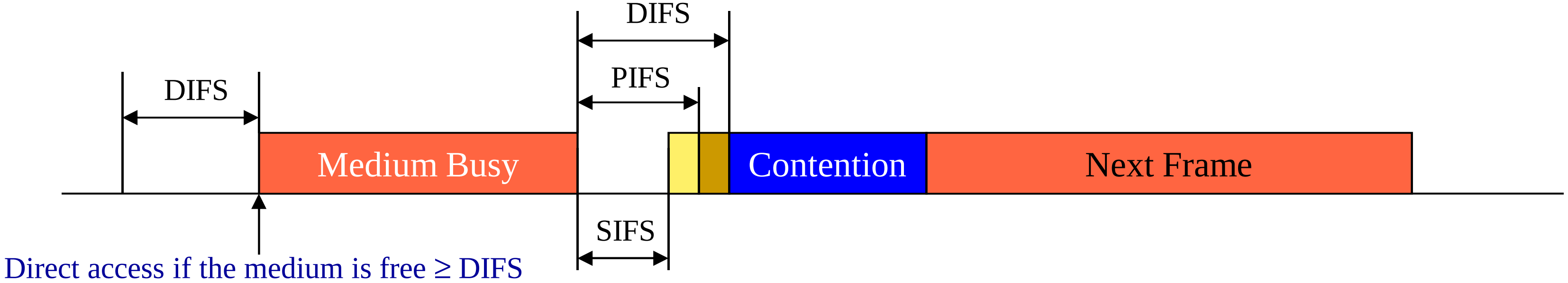
- CSMA: carrier sensing
  - Carrier: don't send
  - No carrier: send
  - Needs to be enhanced in wireless networks
- CA: collision avoidance
  - random backoff
  - priority ack protocol
- MAC coordination function:
  - Distributed Coordination Function (DCF) for multiple access
  - Point Coordination Function (PCF) for polling-based priority
  - Hybrid Coordination Function (HCF) per 802.11e

Practically, CSMA/CA is CSMA with explicit ACK frame

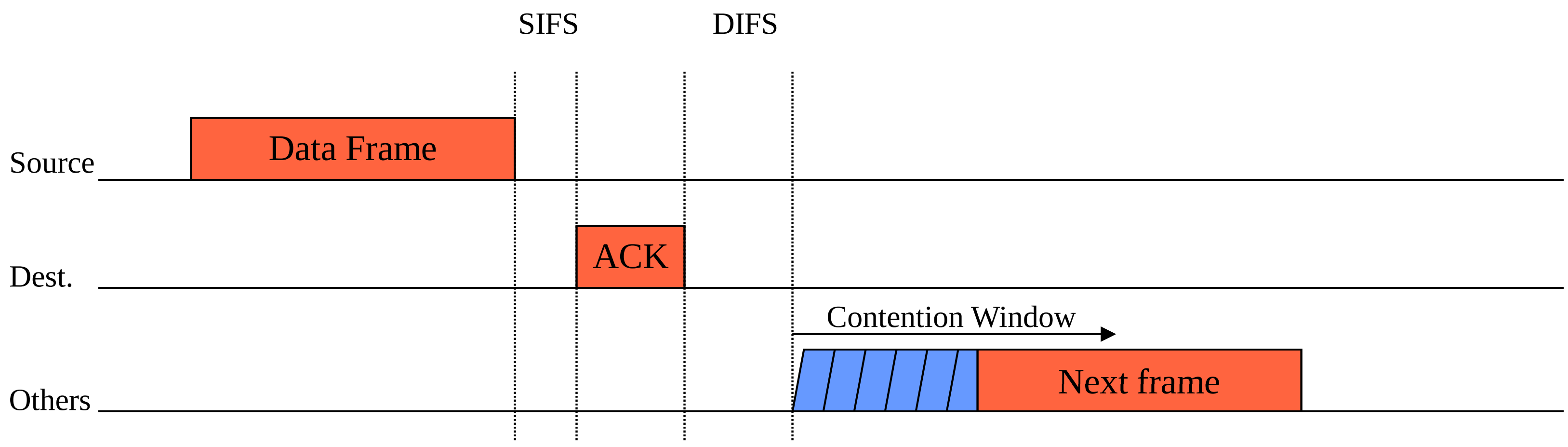
IEEE 802.11 MAC Layer Priority

MAC layer priority is defined through different Inter Frame Spaces

- DIFS(DCF IFS)
  - Lowest priority, for asynchronous data service
- PIFS (PCF IFS)
  - Medium priority, for time-bounded service using PCF
- SIFS (Short Inter Frame Spacing)
  - Highest priority, for ACK, Clear To Send (CTS), Polling response



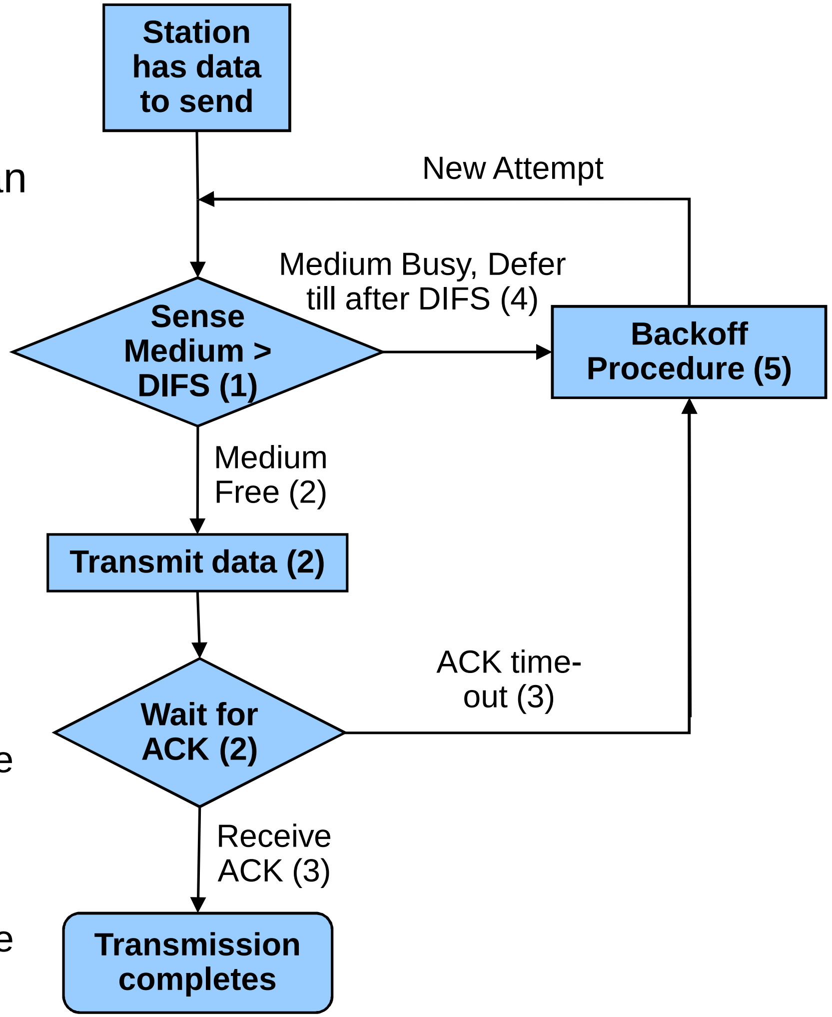
CSMA/CA : ACK Protocol



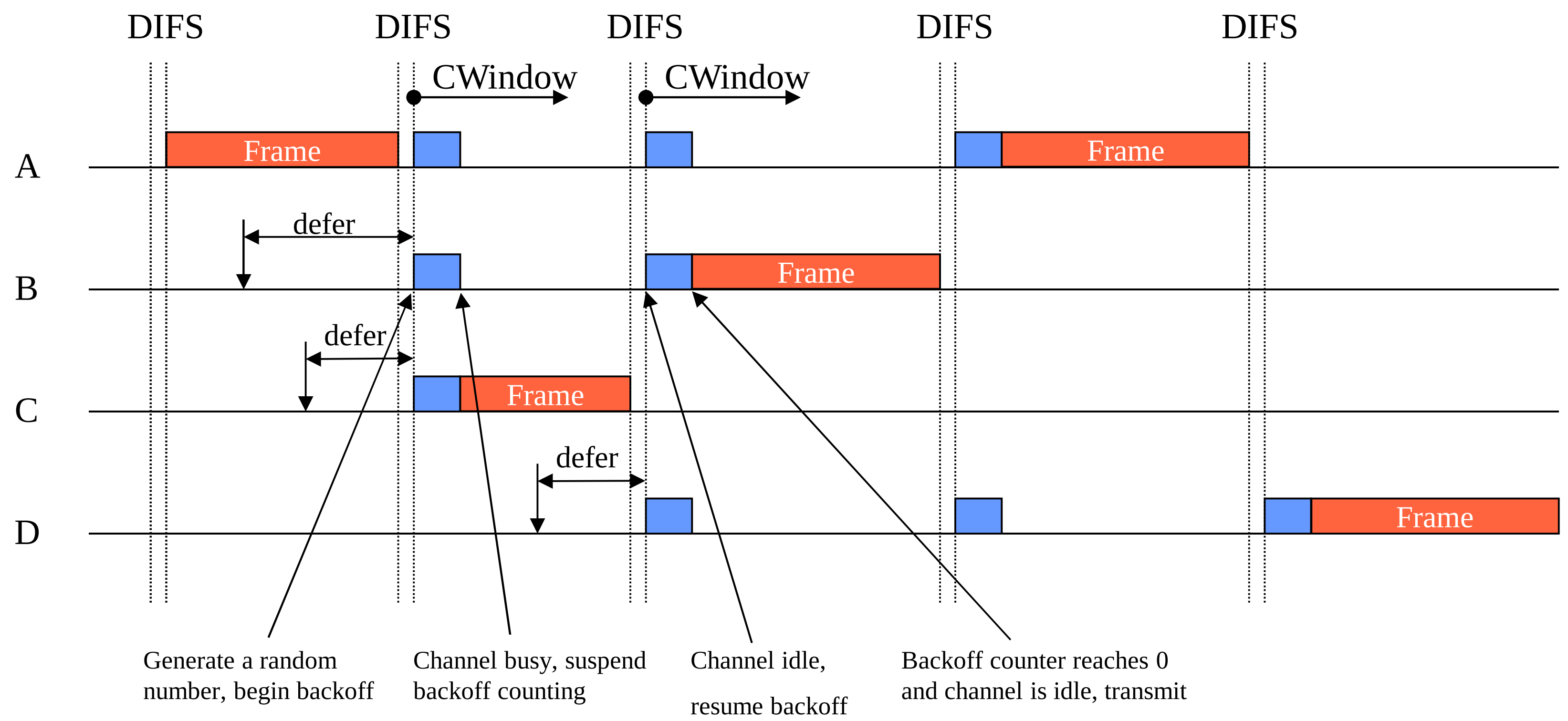
- Receiver of directed frames returns an 14 Byte ACK immediately when CRC is correct.
- If no ACK received, the sender will retransmit after a random backoff

CSMA/CA in DIF Mode

1. Sense medium for a free slot ≥ DCF Inter Frame Space (DIFS)
2. Immediate access when medium is free and start an ACK timer
3. If timeout, goto Backoff procedure. Otherwise transmission completes
4. When medium is not free, defer until the end of current frame transmission + DIFS, then begin backoff procedure
5. To begin Backoff procedure:
   - Choose a random number in (0, Cwindow)
   - Listen to determine if the medium is busy for each time slot
   - Decrement backoff time by one slot if medium is idle
   - Suspend backoff procedure if channel is busy in a time slot
   - Resume backoff when the channel becomes idle again.



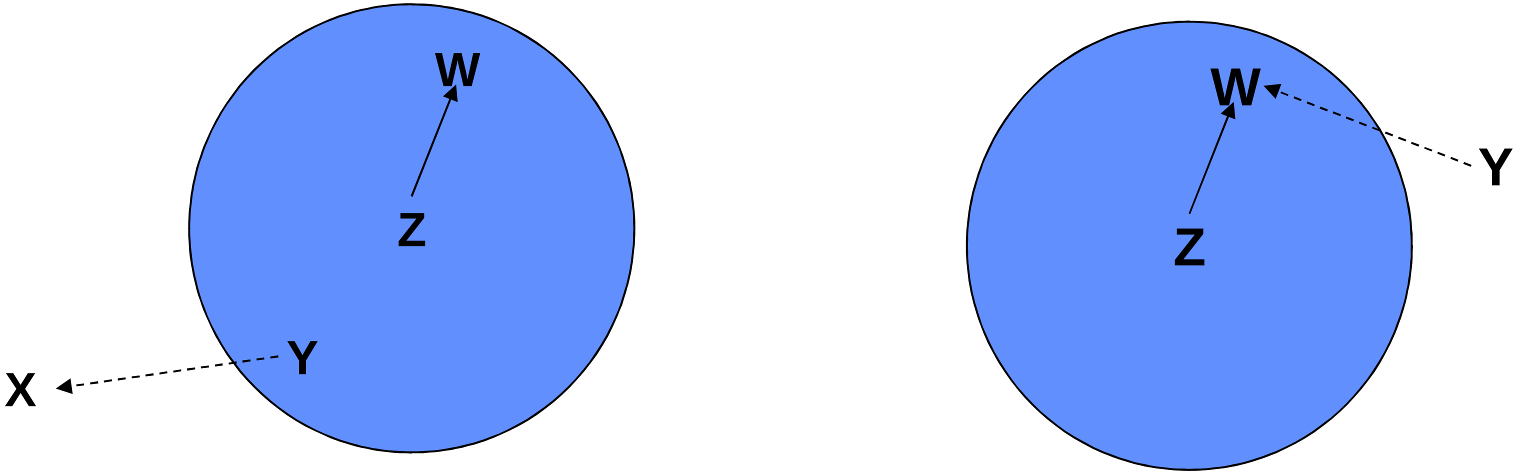
CSMA/CA: Backoff with Cwindow



C(ontention)window in unit of slot time:

- Increase after each failure: 31, 63, 127, 255, 511, 1023, then give up
- Reset to 31 after each successful transmission

Exposed & Hidden Terminal Problems



The Exposed Terminal problem

- Y will not transmit to X even though it can do so

The Hidden Terminal problem

- Y finds that medium is free and transmits a packet to W

RTS/CTS

- The sender send Request-to-Send (RTS): 20bytes
- Receiver returns Clear-to-Send (CTS): 14 bytes
- Then transmission begins
- Solves Hidden Terminal problem

The Address Resolution Protocol

- IP addresses are not recognizable in the interface layer where physical addresses (or MAC addresses) are used.
- Different kinds of physical networks use different addressing schemes.
- Address Resolution Protocol (ARP): maps an IP address to a MAC address per RFC 826.
- Reverse Address Resolution Protocol (RARP): maps a MAC address to an IP address per RFC 903.

ARP Process

- When a source host wants to send an IP packet to a destination, it first *broadcasts* an *ARP request* asking for the MAC address corresponding to a target IP address.
- A target device will return an *ARP reply* with its MAC address.

ARP Packet Format

Hardware Type	Protocol Type	Hardware Size	Protocol Size	Operation Field	Sender Eth. Addr.	Sender IP Addr.	Target Eth. Addr.	Target IP Addr.
2	2	1	1	2	6	4	6	4 bytes

- 28 bytes long.
- An *ARP request* or *ARP reply* is encapsulated in an Ethernet frame.
  - The protocol type in Ethernet frame is set to 0x0806 for ARP messages.
- Hardware Type - Specifies a hardware interface type for which the sender requires a response, i.e. Ethernet (1) in our case
- Protocol type - Specifies the type of high-level protocol address the sender has supplied, ie. IP (Ox0800) in our case
- Hlen - Hardware address length.
- Plen - Protocol address length.
- Operation field specifies ARP request (1), ARP reply (2), RARP request (3), or RARP reply (4).

ARP Request

Hardware Type	Protocol Type	Hardware Size	Protocol Size	Operation Field	Sender Eth. Addr.	Sender IP Addr.	Target Eth. Addr.	Target IP Addr.
2	2	1	1	2	6	4	6	4 bytes

- Ethernet destination: ff:ff:ff:ff:ff:ff (broadcast address)
- Target Ethernet Address: not set.

ARP Reply

Hardware Type	Protocol Type	Hardware Size	Protocol Size	Operation Field	Sender Eth. Addr.	Sender IP Addr.	Target Eth. Addr.	Target IP Addr.
2	2	1	1	2	6	4	6	4 bytes

The ARP reply is sent by the node whose IP address matches the *target IP address* in the ARP request.

- It fills its MAC address into the *target Ethernet address* field of the ARP request.
- It then swaps the two sender addresses (Ethernet and IP addresses) with the two target addresses, sets the op field to 2.
- The ARP reply is sent back to the source host only.

All other nodes receiving the broadcast ARP ignore the request, since their IP addresses do not match the target IP address.

ARP Cache

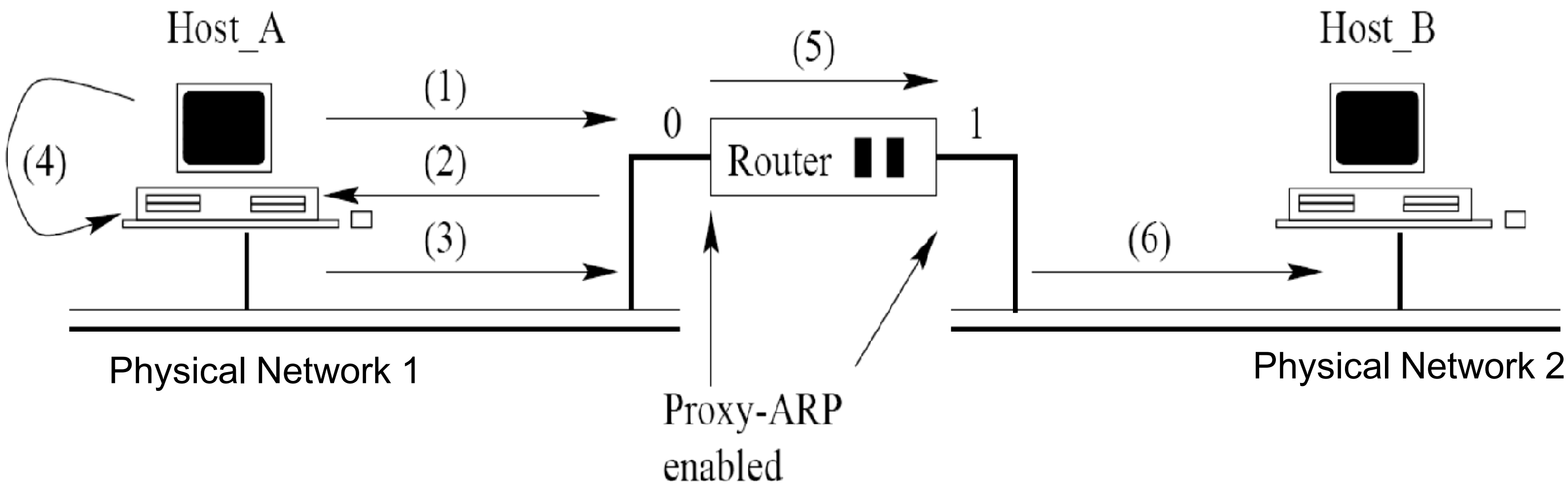
- Sending an ARP request/reply for each IP datagram is inefficient.
- Each host maintains an ARP cache containing the recent resolved IP addresses.
- A source host first checks its ARP cache for the destination MAC address,
  - If an entry is found, sends out the IP packet within an Ethernet frame.
  - Otherwise, sends out an ARP request.

Manipulating the ARP Table

- Elements of an entry in the ARP table:
  - An IP address
  - A MAC address
  - Flags
- A normal entry expires after 20 minutes after it is created or the last time it is referred.
- Manipulate ARP table by the arp command:
  - arp –a: Displays all entries in the ARP table.
  - arp –d: Deletes an entry in the ARP table.
  - arp –s: Inserts an entry into the ARP table.

Proxy ARP (RFC 1027)

- Hide the two physical networks from each other.
- A router answers ARP requests targeted for a host.



(1): Host_A sends ARP request for Host_B's MAC

(2): Router Port 0 replies for Host_B

(3): Host_A sends the frame to Router Port 0

(4): Host_A inserts a new entry in its ARP cache:
{(Host_B's IP) at (Router Port 0's MAC)}

(5): Router forwards the frame to port 1

(6): Router port 1 sends the frame to Host_B

Note

- Two networks are logically in the same subnet (at least from Host A's point of view …)
- Often used by a router connecting to a host with a serial link, ex. PPP to Host B

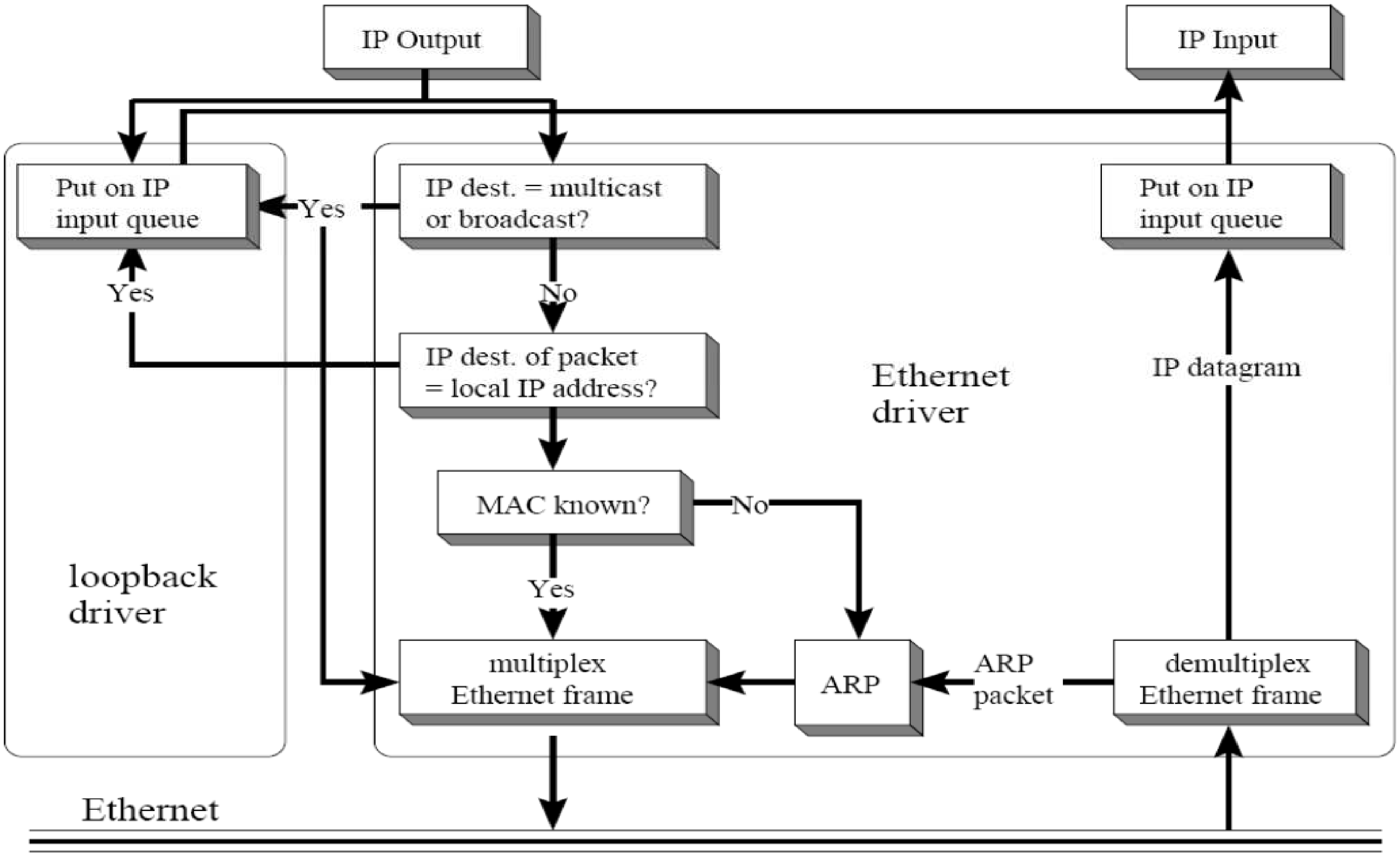
Gratuitous ARP

- Occurs when a host sends an ARP request resolving its own IP address.
- Usually happens when the interface is configured at bootstrap time.
- The interface uses gratuitous ARP to determine if there are other hosts using the same IP address.
- The sender's IP and MAC address are broadcast, and other hosts will insert this mapping into their ARP tables.

Loopback Interface

- Most TCP implementations have a loopback interface with IP address 127.0.0.1 and named as localhost.
- The localhost behaves as a separate data link interface.
- A packet that is sent to the loopback interface moves down the protocol stack and is returned back by the driver software for the localhost "device".
- Used for debugging.
- Packets sent to loopback interface will not appear on network.

Network Interface Operations



Functional Diagram of an Ethernet Interface Card

Maximum Transmission Unit

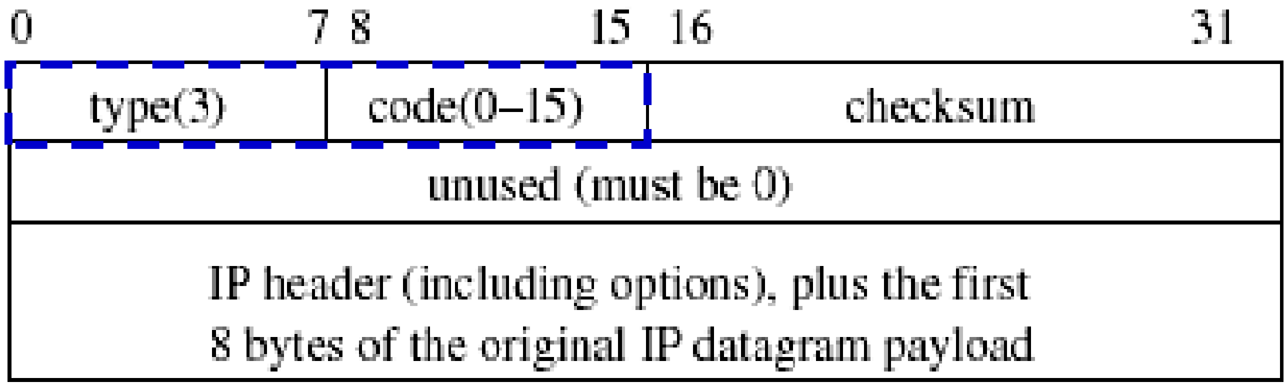
- There is a limit on the data packet size of each data link layer protocol.
- This limit is called Maximum Transmission Unit (MTU).
- MTUs for various data link layers:
  - Ethernet, PPP: 1500 bytes
  - FDDI: 4352 bytes
  - PPP (low delay): 296 bytes
- MTU does not count its own header and trailer bytes of the data link protocol. e.g. Ethernet's MTU is 1500 bytes.

Dest. Addr	Src. Addr.	Type	Data	CRC
6	6	2	46-1500	4

Internet Control Message Protocol

- The Internet Control Message Protocol (ICMP) is the protocol used for error and control messages in Internet.
- ICMP provides an error reporting mechanism of routers to the sources.
- All ICMP packets are encapsulated as IP datagrams (IP protocol type 1)
- The packet format is simple.

Example below shows destination unreachable error message below:



Types of ICMP Packets

Many ICMP packet types exist, each with its own format.

Type	Code	Description	–
0	0	echo reply	query
3	0–15	destination unreachable	error
5	0–3	redirect	error
8	0	echo request	query
9	0	router advertisement	query
10	0	router solicitation	query
11	0–1	time exceeded	error

ICMP Message Types

- ICMP messages are either query messages or error messages.
- ICMP query messages:
  - Echo request / Echo reply
  - Router advertisement / Router solicitation
  - Timestamp request / Timestamp reply
  - Address mask request / Address mask reply
- ICMP error messages:
  - Host unreachable
  - Source quench
  - Time exceeded
  - Parameter problem

ICMP Error Messages

- Each ICMP error message contains the header and at least the first 8 bytes of the IP datagram **payload** that triggered the error message.
- To prevent that too many ICMP messages, ICMP error messages are NOT sent
  - for multiple fragments of the same IP datagrams
  - in response to an error message
  - in response to a broadcast packet

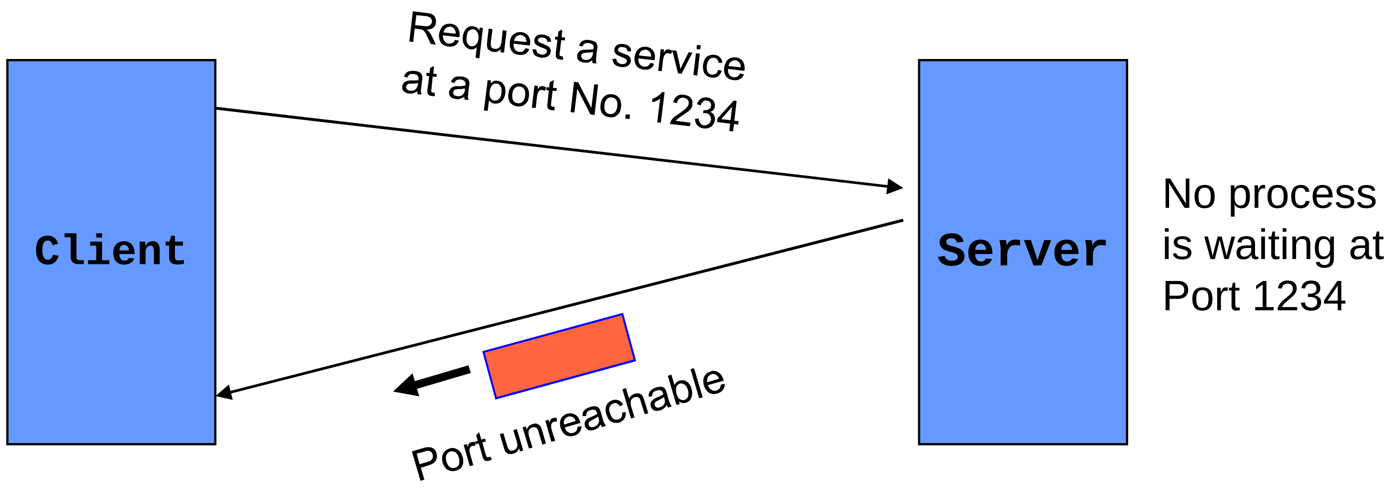
ICMP Type 3 Error Message Code

There are 16 different ICMP error messages (‘codes’) of type “Destination Unreachable” (Type = 3)

Code	Message Type	Code	Message Type
0	Network unreachable	8	Source host isolated
1	Host unreachable	9	Destination network administratively prohibited
2	Protocol unreachable	10	Destination host administratively prohibited
3	Port unreachable	11	Network unreachable for TOS
4	Fragmentation needed but bit not set	12	Host unreachable for TOS
5	Source route failed	13	Communication administratively prohibited by filtering
6	Destination network unknown	14	Host precedence violation
7	Destination node unknown	15	Precedence cutoff in effect

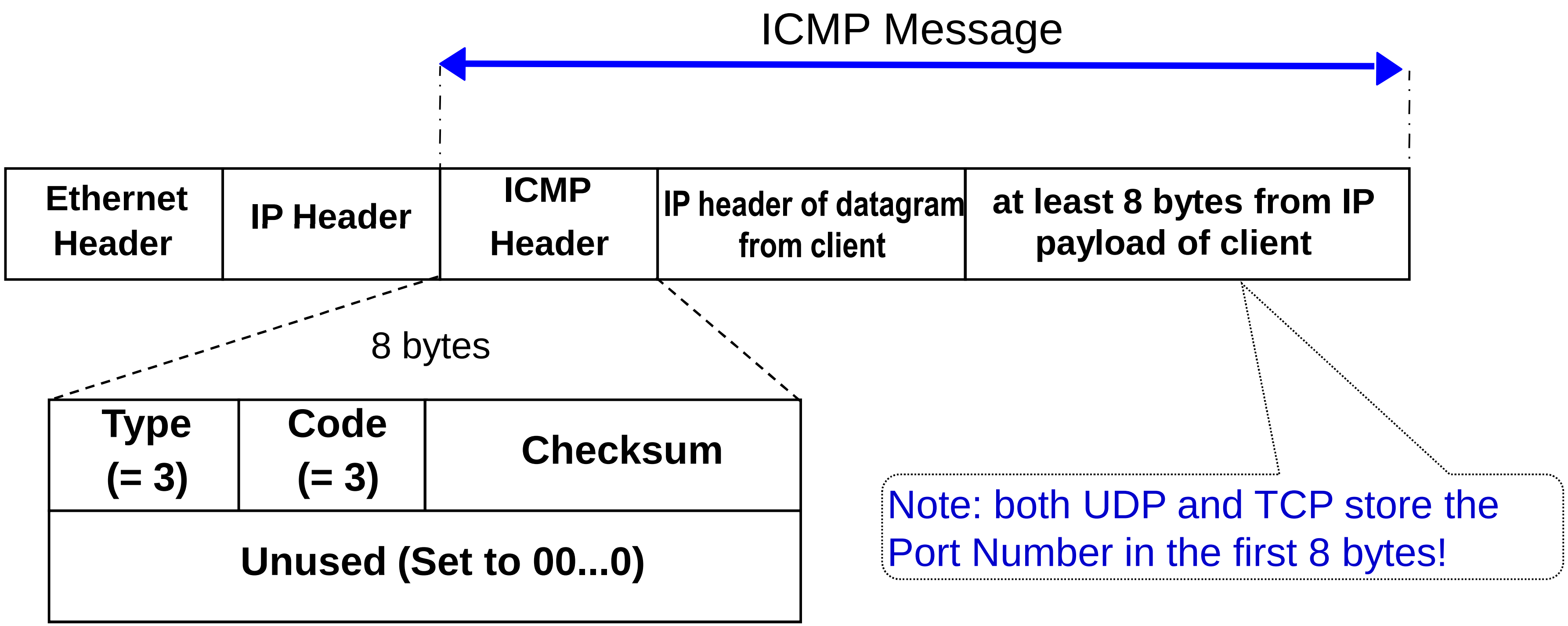
ICMP Port Unreachable

If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a port unreachable message to the source host.



ICMP Port Unreachable

Format of the Port Unreachable Message

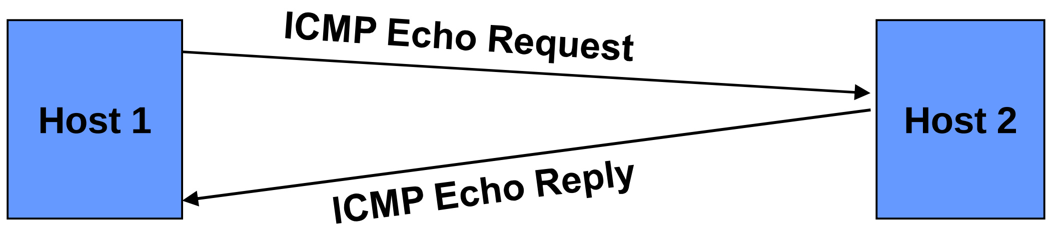


Packet InterNet Gopher (PING)

- PING is a program that utilizes the ICMP echo request and echo reply messages.
- PING is used to verify if a certain host is up and running. It is used extensively for fault isolation in IP networks.
- PING can be used with a wide variety of options, e.g.
  - *-R* (Record route): includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets.
  - *-s packetsize*: specifies the number of data bytes to be sent (default is 56) (in newer implementations, -s is used to continuously generate queries)

Echo Request and Reply

- Ping's are handled directly by the kernel.
- Each Ping is translated into an ICMP Echo Request.
- The Ping'ed host responds with an ICMP Echo Reply.

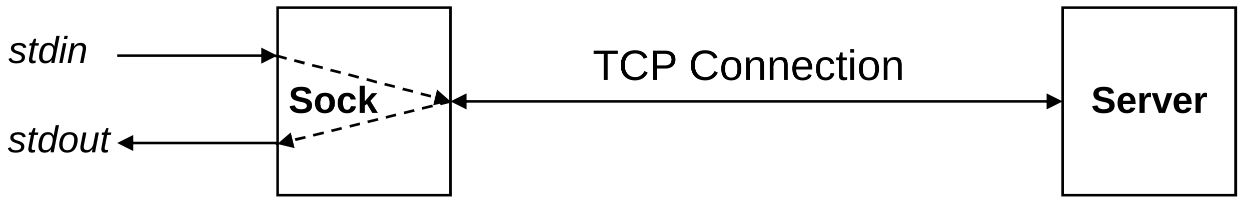


- Message format
  - Identifier is set to process ID of querying process.
  - Sequence number is incremented for each new echo request.

0 – 7	8 – 15	16 – 31
type(0 or 8)	code(0)	checksum
identifier		sequence number
optional data		

Sock Traffic Generator

- Sock is a test program.
  - Can be run as a client or as a server
  - Use UDP or TCP.



- Sock operates in one of the following four modes:
  - Interactive client
    - > Copy data from stdin, send to server, receive data back from server, copy on screen (stdout)
  - Interactive server
    - > Wait for a request from client
  - Source client
    - > Send packets to a specific server
  - Sink server
    - > Receive packets from a client and discard the data