Topics Today

- Physical layer: chips versus bits
- Link layer
- Media access control (MAC)
- Ethernet
- MPLS

Protocol Layering

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Link
1	Physical

Protocol Layering

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Link
1	Physical

Physical Layer (Layer 1)

- **Responsible for specifying the physical medium**
  - Category 5 cable (Cat5): 8 wires, twisted pair, RJ45 jack
  - WiFi wireless: 2.4GHz
- **Responsible for specifying the signal**
  - 100BASE-T: 5-level pulse amplitude modulation (PAM-5)
  - 802.11b: Binary and quadrature phase shift keying (BPSK/QPSK)
- **Responsible for specifying the bits**
  - 100BASE-T: 4-to-6 bit-to-chip encoding, 3 chip symbols
  - 802.11b: Barker code (1-2Mbps), complementary code keying (5.5-11Mbps)

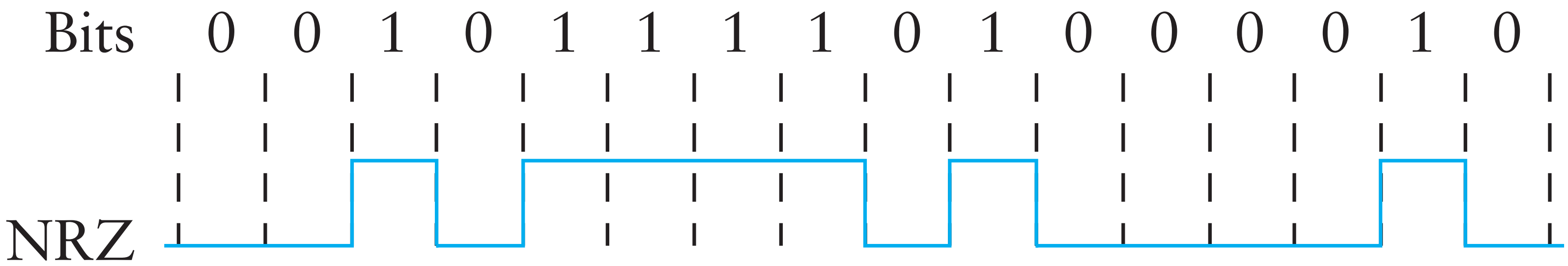
Specifying the signal

- **Chips versus bits**
  - Chips: data (in bits) at the physical layer
  - Bits: data above the physical layer
- **Physical layer specifies Analog signal $\leftrightarrow$ chip mapping**

How fast can you transmit information?

- **Depends on bandwidth and Signal/Noise ration**
- **Shannon: Channel capacity** $C = B \log_2(1 + S/N)$
  - B is bandwidth of line
  - S and N are average signal & noise power
- **For *any* transmission rate $R < C$, can have arbitrarily low error rate**
- **Example: Telephone line**
  - 3 KHz b/w, 30 db S/N $= 10^{30/10} = 1000$
  - $C \approx 30$ Kbps (so 56 Kbps modems need better S/N ratio)
- **Crude intuition for Shannon**
  - Sample rate $\sim$ B
  - V voltage levels encode $\log_2 V$ bits, so bits/sample $\sim \log_2(1 + S/N)$

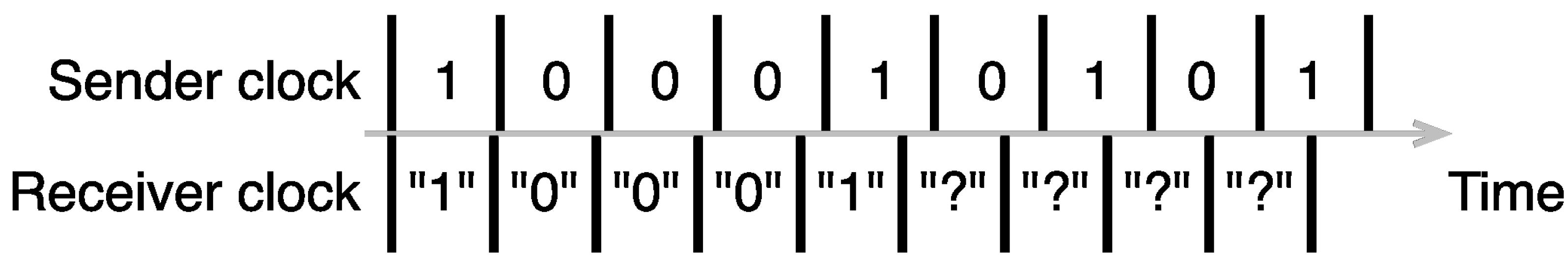
Straw man: On-off keying



- **To transmit 0 bit, sent $0\ V$, to transmit 1, sent $+5\ V$**
  - A bit is a chip in this scheme
- **OOK a form of Amplitude Shift Keying (ASK)**
  - Bits are encoded in amplitude of the signal
  - Can also have frequency shift keying (FSK)
  - And phase shift keying (PSK)
- **Also an example of non-return to zero (NRZ)**
  - E.g., four 1 bits transmitted by asserting $+5\ V$ for 4 clock ticks

NRZ drawbacks

- **Consecutive 1s or 0s are problematic**
- **Non signal could be interpreted as 0s (or vice versa)**
- **"Baseline wander" problem**
  - Where is threshold between low and high?
  - Could compare signal to average value, but avg. will drift
- **Sender and receiver need synchronized clocks**
  - Otherwise, can experience "bit slip"



Non-return to Zero Inverted (NRZI)

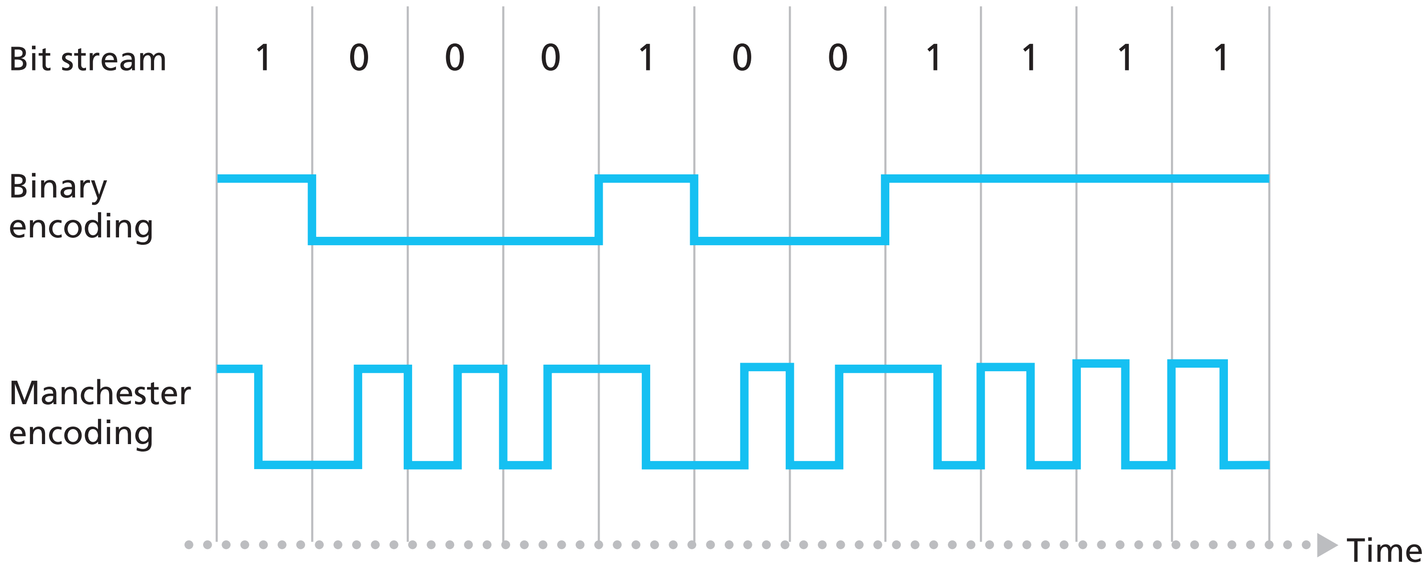
- Encode 1 with transition from current signal
- Encode 0 by staying at same level
- At least solves problem of consecutive 1s

Encoding Goals

- **DC balancing (same number of 0 and 1 chips)**
- **Clock synchronization**
- **Can recover from some chip errors**
- **Can constrain analog signal patterns to make signal more robust**
- **Want near-chanel capacity with negligible errors**
  - But Shannon only says it's possible, doesn't tell us how
  - Codes could also get computationally expensive
- **In practice:**
  - Higher encoding → fewer bps, more robust
  - Lower encoding → more bps, less robust

Manchester Encoding

- **Map bit 0 → chips 01, bit 1 → chips 10**
  - Transmission rate now 1 bit per *two* clock cycles
  - Like XORing an NRZ encoding with the clock



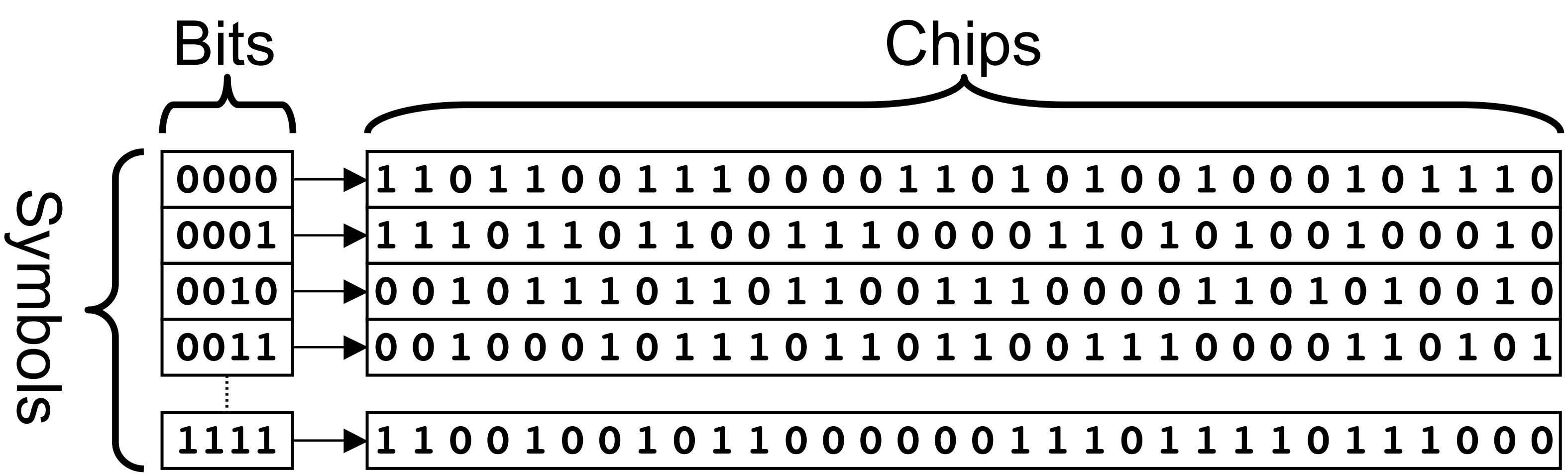
- **Solves clock synchronization & baseline wander**
- **But cuts transmission rate in half**

4B/5B

- **Every 4 bits of data encoded in 5 chips**
- **5-bit codes selected to have no more than one leading 0 and no more than two trailing 0s**
  - thus, never get more than three consecutive 0s
- **16 codes used for all 4-bit sequences**
- **Resulting 5bit codes are transmitted using NRZI**
- **Remaining codes used for other purposes**
  - E.g., 11111 – line idle, 00000 – line dead, . . .
- **Achieves 80% bit/chip efficiency**

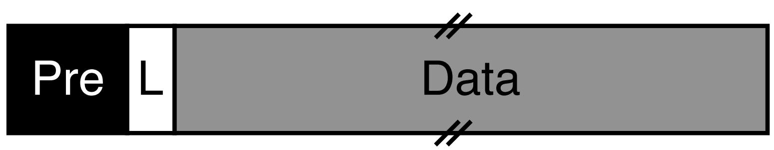
802.15.4

- **Standard for low-rate wireless personal networks**
  - Must tolerate high chip error rates
- **Uses a 32-to-4 chip-to-bit encoding**



Physical Layer Frames

- **Usually minimalist: "here's N bytes"**
  - Start symbol/preamble
  - Length field
  - Payload (link layer frame)

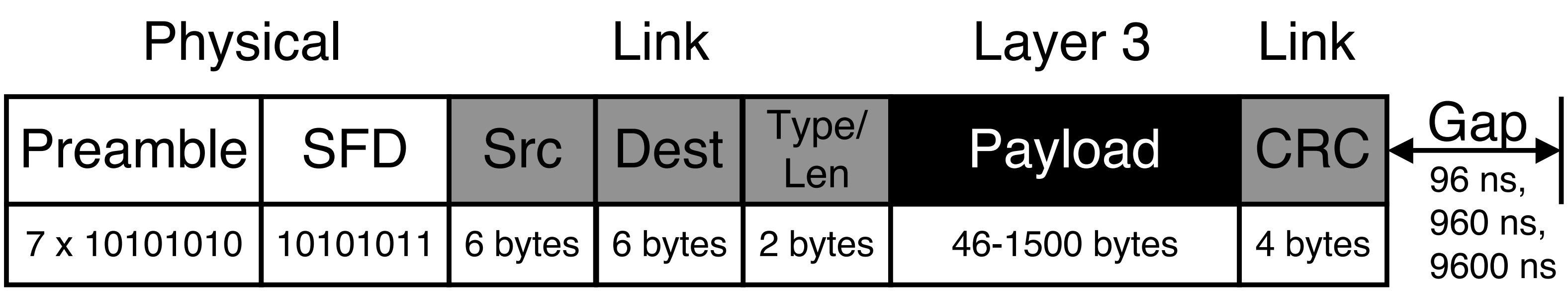


Link Layer Responsibilities

- **Single-hop addressing (e.g., Ethernet addresses)**
- **Media access control**
  - Link-layer congestion control
  - Collision detection/collision avoidance
- **Single-hop acknowledgements**

Ethernet: 802.3

- **Dominant wired LAN technology**
  - 10BASE5 (vampire taps)
  - 10BASE-T, 100BASE-TX, 1000BASE-T
- **Frame format:**



Ethernet Addressing

- **Each Ethernet card has a unique 48-bit ID**
  - Example: `www.scs.stanford.edu` has 00:07:e9:0f:1f:3e
  - Example: `myth15` has 00:1e:c9:2f:a2:9c
- **24-bit organizationally unique identifier, 24-bit ID**
  - 0x000000–0x000009: Xerox
  - 0x0007e9: Intel (www.scs)
  - 0x001ec9: Dell (myth15)
  - `http://standards.ieee.org/regauth/oui/oui.txt`

Media Access Control (MAC)

- **Control access to shared physical medium**
  - E.g., who can use coax/radio when?
  - If everyone talks at once, no-one hears anything
  - This job falls to the link layer
- **Prevent collisions by controlling when nodes send**
- **Variety of approaches**
  - Time Division Multiple Access (TDMA)
  - Carrier Sense Multiple Access, Collision Detection (CSMA/CD)
  - Carrier Sense Multiple Access, Collision Avoidance (CSMA/CA)
  - Request-to-send, clear-to-send (RTS/CTS)

MAC Approaches

- **Channel Partitioning**
  - Divide channel into smaller "pieces," allocate pieces to nodes
- **Random Access**
  - Don't divide channel, allow conflicts
  - Recover from errors caused by conflicts
- **"Taking turns"**
  - Nodes take turns, but nodes with more to send can take longer turns
- **MAC goals: Maximize use of the link capacity**
  - One node should get 100% in absence of competition
  - Multiple nodes can each get a share, not collide

TDMA

- **Divide time into slots**
  - Each device is allowed to transmit in some number of slots
- **No collisions**
- **Link is fully utilized when everyone transmits:**

1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4

- **Single node cannot use all of the capacity ($\frac{1}{n}$):**

1				1				1				1			

- **Can't get full link utilization unless everyone transmits:**

1			4	1			4	1	2		4	1	2		

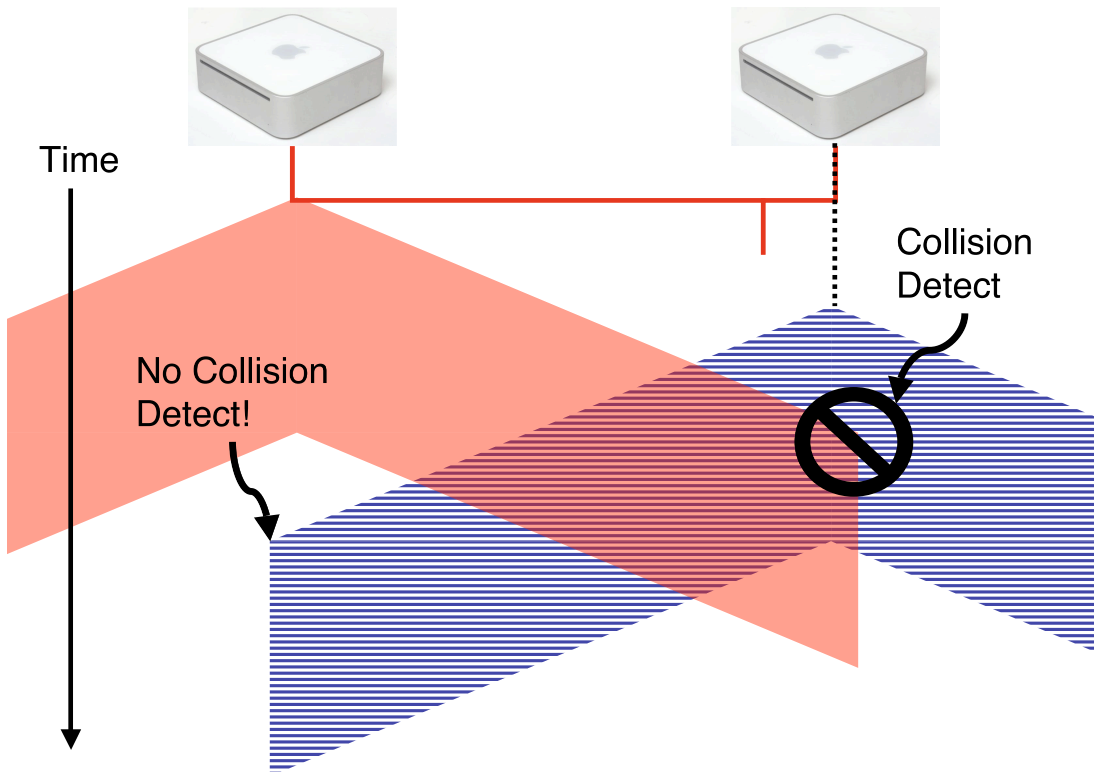
CSMA

- **Node senses the channel for activity**
- **Transmits if it thinks the channel is idle**
- **CSMA/CD: can detect if there is a collision, and back off**
  - Randomized backoff time, grows exponentially
  - After C consecutive collisions, wait $rand(0, 2^C) \cdot 512$
  - Drop when C grows large (in practice)

Collision Detect (10base2 Ethernet)

- **Detect collision when average voltage spikes**
  - 10base2 uses Manchester encoding
  - Has constant average voltage unless multiple transmitters
- **When a node detects a collision**
  - Broadcasts jam signal to ensure other nodes drop packet
- **Collision detection constrains protocol**
  - Imposes min. packet size (64 bytes)
  - Imposes maximum network diameter (2800 m)
  - Ensure transmission time $\geq$ twice propagation time—why?

Violating Timing Constraints



- **Without min packet size, might miss collision**

Ethernet Capture Effect

- Exponential backoff leads to self-adaptive use of channel
- When a node succeeds, it transmits the next packet immediately
- Result: bursts of packets from single nodes

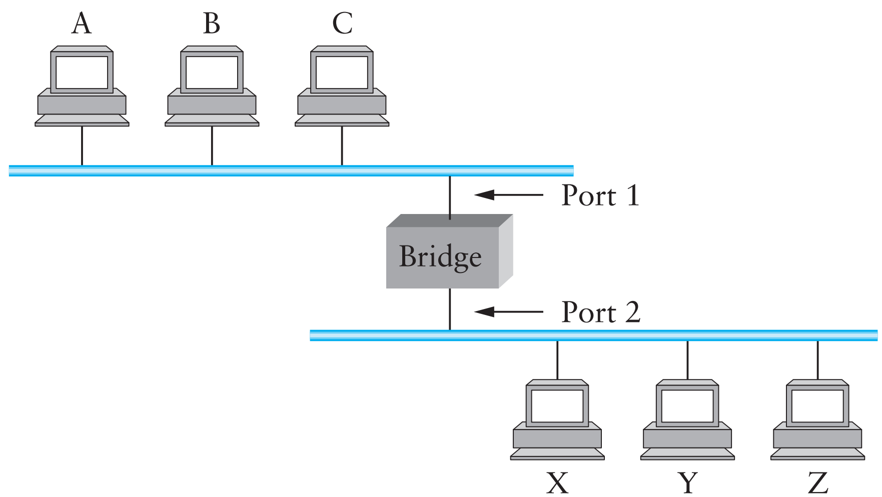
Ethernet Speeds

- **Network diameter limits:**
  - 10Mbps: 2800m
  - 100Mbps: 205m
  - Gigabit: 205m!
- **Gigabit Ethernet**
  - Uses more of the CAT5 wires (125 MHz $\cdot$ 8 signals)
  - Pad with dummy data (signal extension) for CD
    (so min packet size is now 512 *bytes*, not bits)

Hubs vs. Switches

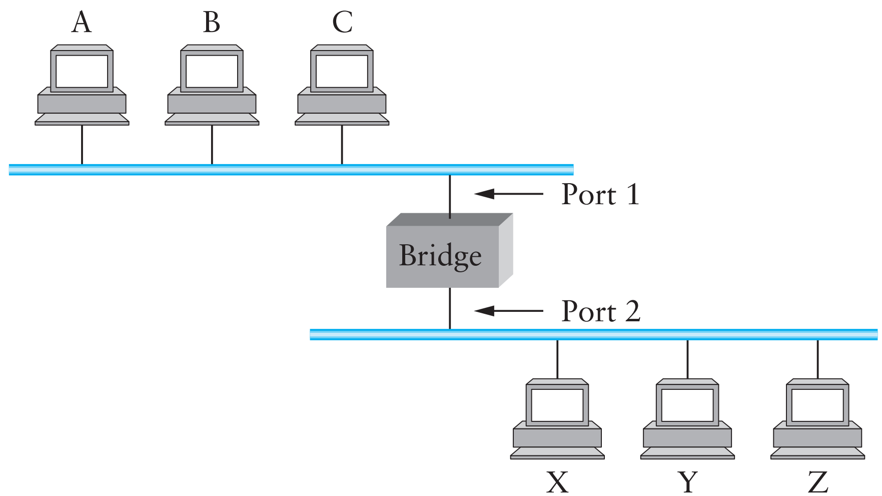
- **Hub: connects multiple Ethernet segments to act like a single segment (shared collision domain, physical layer connectivity)**
- **Switch: store and forward between segments (single collision domains, link layer connectivity)**
- **Very little Ethernet today is shared**
  - Means collision detection never triggered (duplex, separate RX and TX wires)
  - 10Gbps Ethernet standard does not allow shared medium

Bridges and extended LANs



- **LANs have physical limitations (e.g., 208 m)**
- **Connect two or more LANs with a *bridge***
  - Operates on Ethernet addresses
  - No encapsulation required
- **Ethernet switch like a multi-way bridge**

Learning bridges



- **Idea: Don't forward packet if not useful**
  - If you know recipient is not on that port
- **Learn host's location based on source address**
  - Switch builds a table when it receives packets

A	B	C	X	Y	Z
1	1	1	2	2	2

- **Table says when *not* to forward packet**
  - Does not need to be complete for correct behavior
  - Spanning tree algorithm avoids loops

Congestion Interaction

- Congestion can occur at layer 2 (collisions, high utilization)
- Congestion control can occur at layer 2 (backoff)
- Congestion can occur at layer 3 (packet drops)
- Congestion control can occur at layer 4 (rate adaptation)
- Interactions are non-trivial

ARP and DHCP, revisited

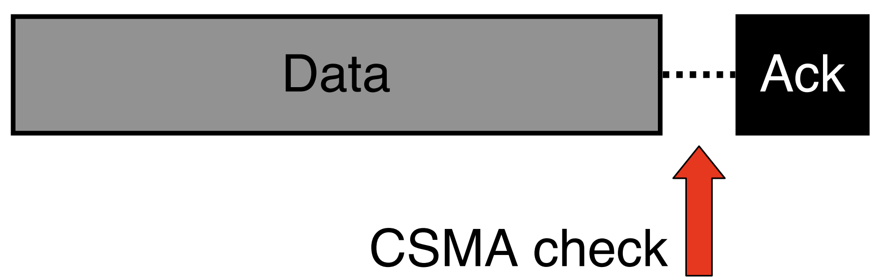
- **DHCP allows a node to dynamically obtain an IP address, netmask, and gateway**
- **Address Resolution Protocol maps IP addresses to link address**
- **Common exchange:**
  - Broadcast DHCP discover
  - Receive gateway IP address IP_G, local address IP_A
  - ARP to get gateway address IP_G (announcing self), receive Ether_G
  - Send packet to IP_B using $Ether_G$ as next hop
- **What if node is on the subnet?**

Layer 2 Acknowledgements

- **Common in wireless (more on this in wireless lecture)**
- **If layer 2 successfully receives a frame, it immediately sends an ACK**
- **Assumes** $t_{\text{prop}} \ll t_{\text{trans}}$
- **Hypothetical situation:**
  - Let's say a router won't send an ACK if it drops the packet
  - Let's say a router will keep on retrying a packet until it is ACKed
  - Do we still need end-to-end ACKs?

Ack Effect on CSMA

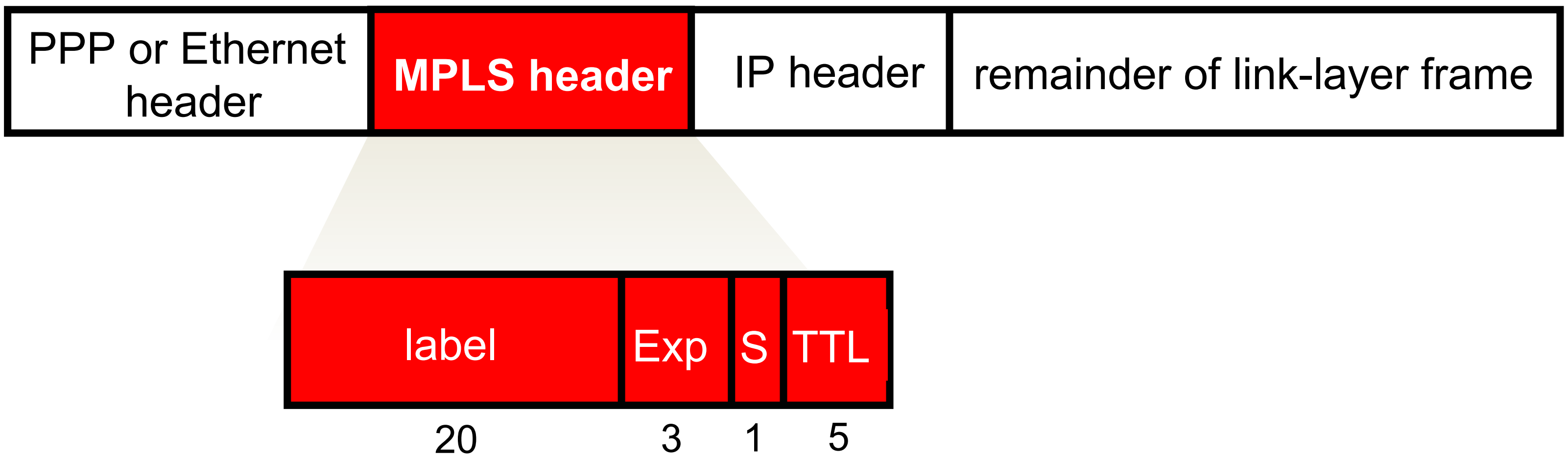
- Layer 2 acks require two channel checks
- Want to make sure we don't check between packet and ACK



MPLS

- **Multiprotocol Label Switching**
- **Sits between layer 2 and 3 (“layer 2.5”)**
- **Prepend a “label” to frame**
- **Switch in terms of label, rather than destination address**
  - Two packets to the same destination can take different paths
  - Separating addressing from forwarding enables traffic engineering
  - Label changes from input to output

MLPS packet format



- **20-bit label**
- **3 experimental bits**
- **1 “bottom of stack bit”**
  - Allows multiple MPLS headers to be stacked in a packet
- **5-bit TTL (since network-level TTL not used)**

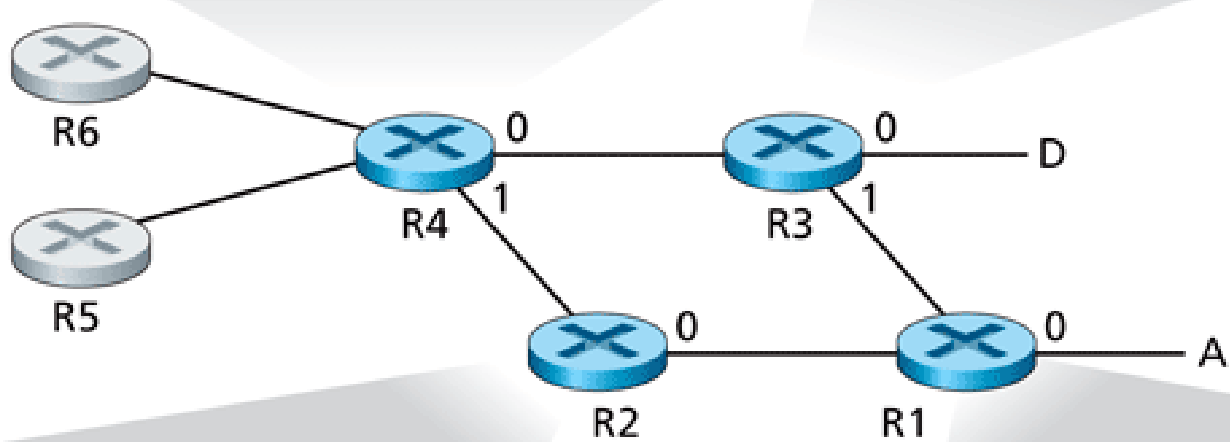
MLPS Architecture

- **Label Edge Routers (LERs)**
  - Talks to regular IP routers and MPLS-enabled ones
- **Label Switch Routers (LSRs)**
  - E.g., The core routers in a large backbone provider
- **Label Distribution Protocol (LDP)**
- **Label Forwarding Information Base (LFIB)**

Example MPLS (from textbook)

in label	out label	dest	out interface
	10	A	0
	12	D	0
	8	A	1

in label	out label	dest	out interface
10	6	A	1
12	9	D	0



in label	out label	dest	out interface
8	6	A	0

in label	out label	dest	out interface
6	–	A	0

Figure 5.37 ♦ MPLS-enhanced forwarding