THE UNIVERSITY OF MELBOURNE

ELEN 90061 Communication Networks

Module 3 – Link Layer

Dr. Rajitha Senanayake

Lesson Overview

- Link Layer
- Error detection and correction
- Media Access Control (MAC)
- MAC protocols
- MAC Addressing

Lesson Overview

- How are the network layer datagrams encapsulated in the link layer frames for transmission over a single link?
- Are different link layer protocols used in different links?
- How is the transmission control in broadcast links resolved?
- Is there addressing in link layer?
- What is the difference between a switch and a router?
- …

Suggested Reading

Note that there is overlap between these reading materials. It is a comprehensive list and you can use slides as a guideline for what to focus on.

- Chapters 3 and 4 from Tanenbaum
- Chapter 6 from Kurose-Ross

Link Layer

application
transport
network
link
physical

The Data Link Layer

Concerns how to transfer messages over link(s)

- Messages (here) are limited-size **frames**
- Link layer builds upon the physical layer

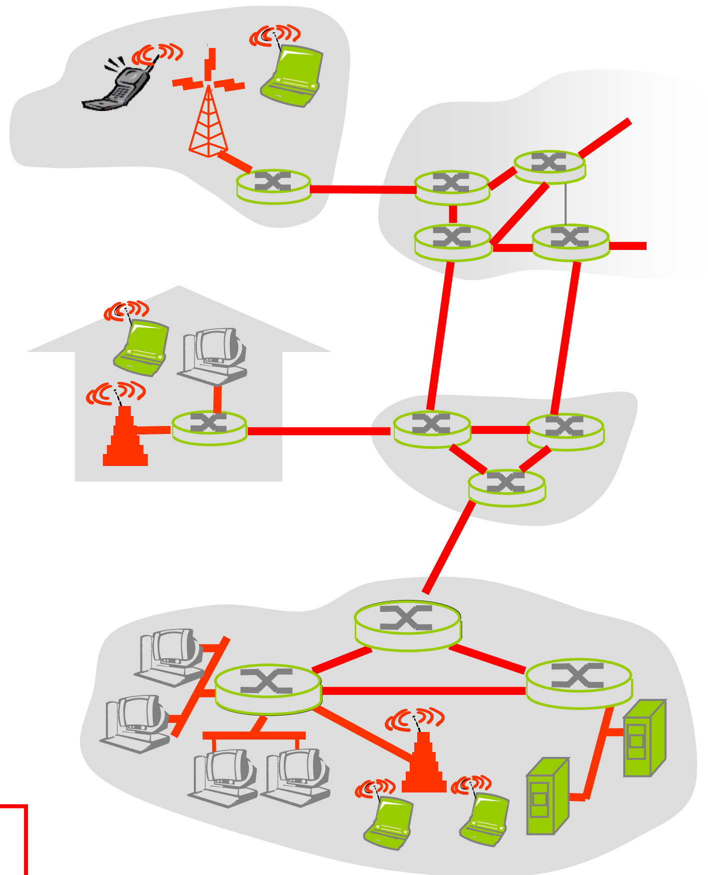
Overview:

- understand principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - *[reliable data transfer, flow control - we will discuss these later in the network layer context]*
- instantiation and implementation of various link layer technologies

Link Layer: Introduction

Some terminology:

- Any device that runs a link layer protocol is a **node**
- communication channels that connect adjacent nodes a along communication path are **links**
  - wired links
  - wireless links
- Layer 2 data unit is a **frame**, encapsulates packet from layer 3



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

Link layer: context

- data is transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- each link protocol provides different services
  - e.g., may or may not provide *reliable data transfer* over link

  (difficulty of providing QoS on the Internet)

<u>transportation analogy</u>

- trip from Melbourne to Sydney
  - Uber: Melbourne to MEL
  - plane: MEL to SYD
  - train: SYD to Sydney CBD
- tourist = data
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

Link Layer

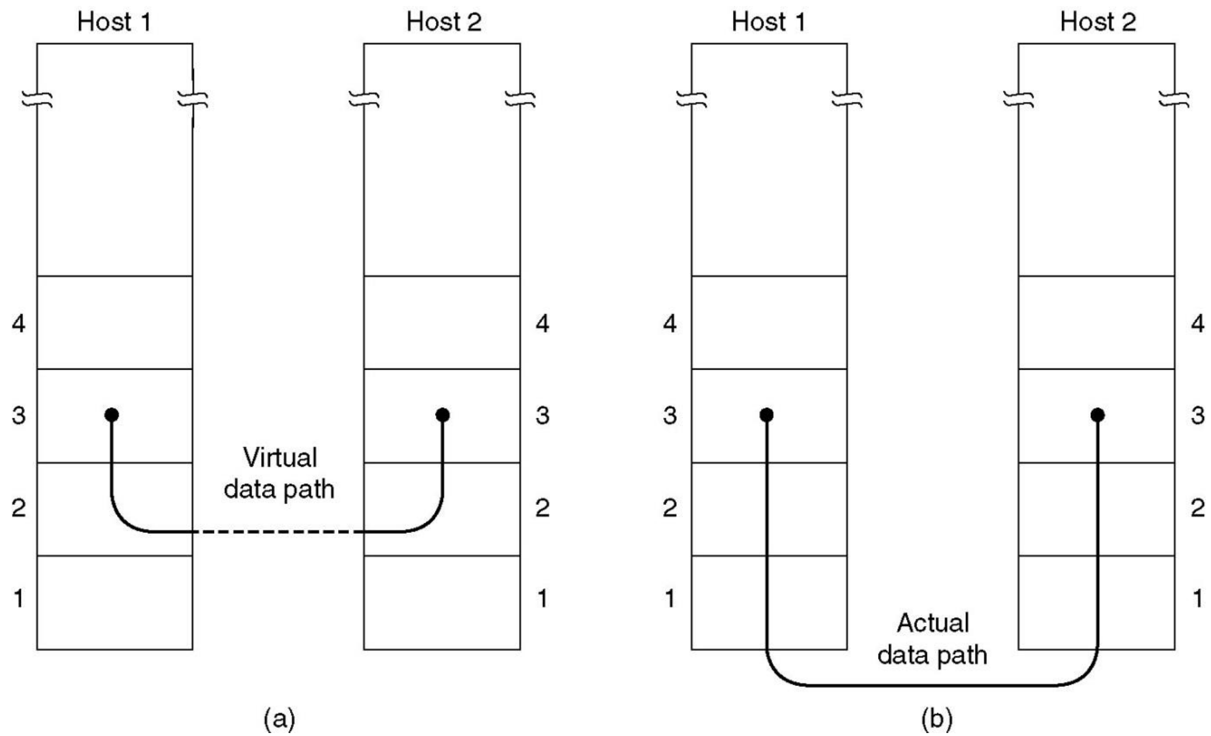
application

transport

network

link

physical



(a) virtual (b) actual communication.

Link Layer Services

- *framing, link access:*
  - encapsulate data packet into frame, adding header, trailer
  - channel access if shared medium
  - **"MAC" addresses used in frame headers** to identify source, destination
    - *different from IP address!*
- *reliable delivery between adjacent nodes*
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates

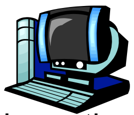
Question: why do we need both link-level and end-end reliability?

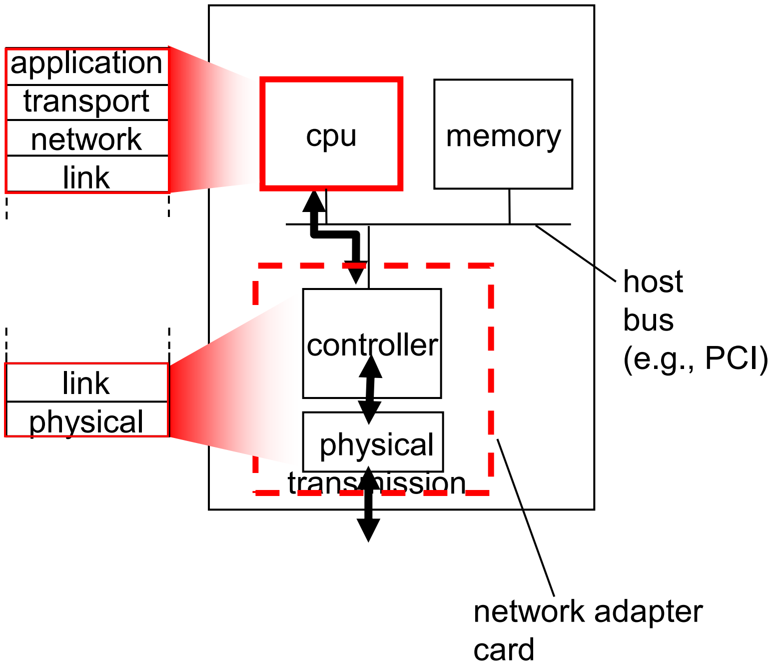
Link Layer Services (more)

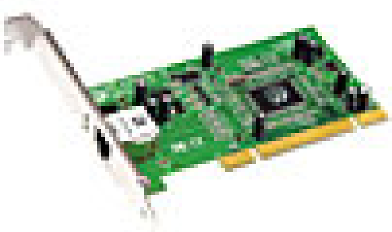
- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection*:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - *signals sender for retransmission or drops frame*
- error correction:
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *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?

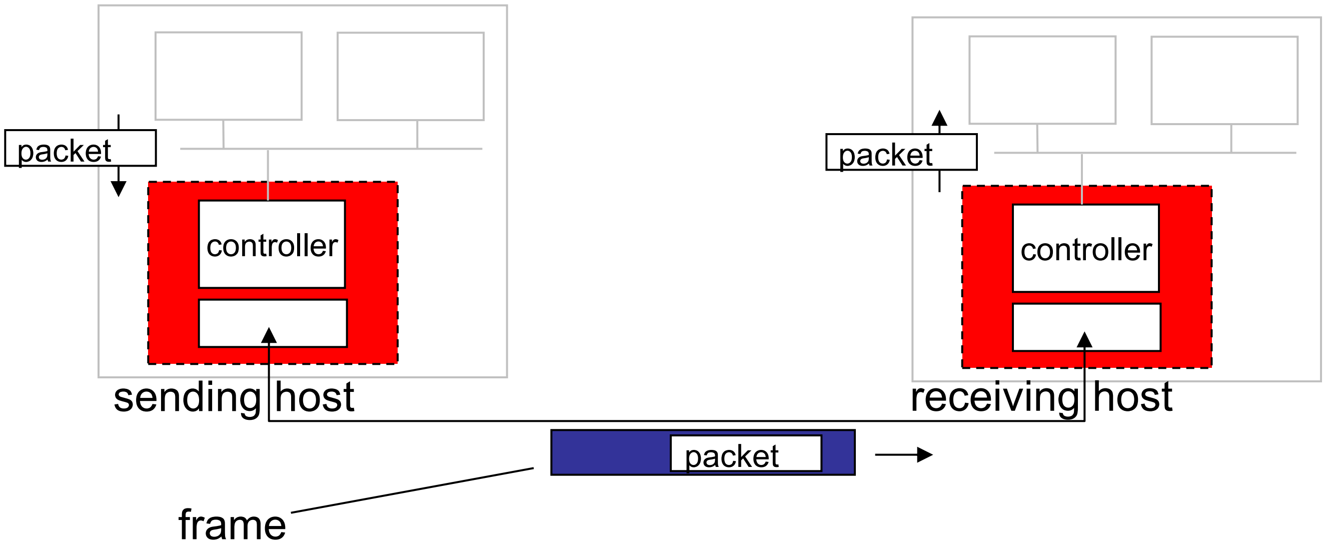
- in each and every host
- **link layer implemented as an embedded part of motherboards or OS.**
- Historically, implemented in "adaptor" (*network interface card* NIC)
  - Ethernet card, PCMCI card, 802.11 card
  - implements link, physical layer
- combination of hardware, software, firmware





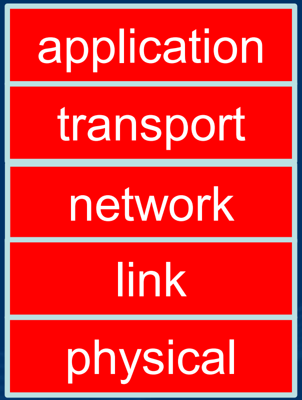


Adaptors Communicating



- sending side:
  - encapsulates packet *(from network layer)* in frame
  - adds error checking bits, reliable data transfer, flow control
- receiving side
  - looks for errors, reliable data transfer, flow control
  - extracts data, passes to upper layer at receiving side

Error Detection and Correction



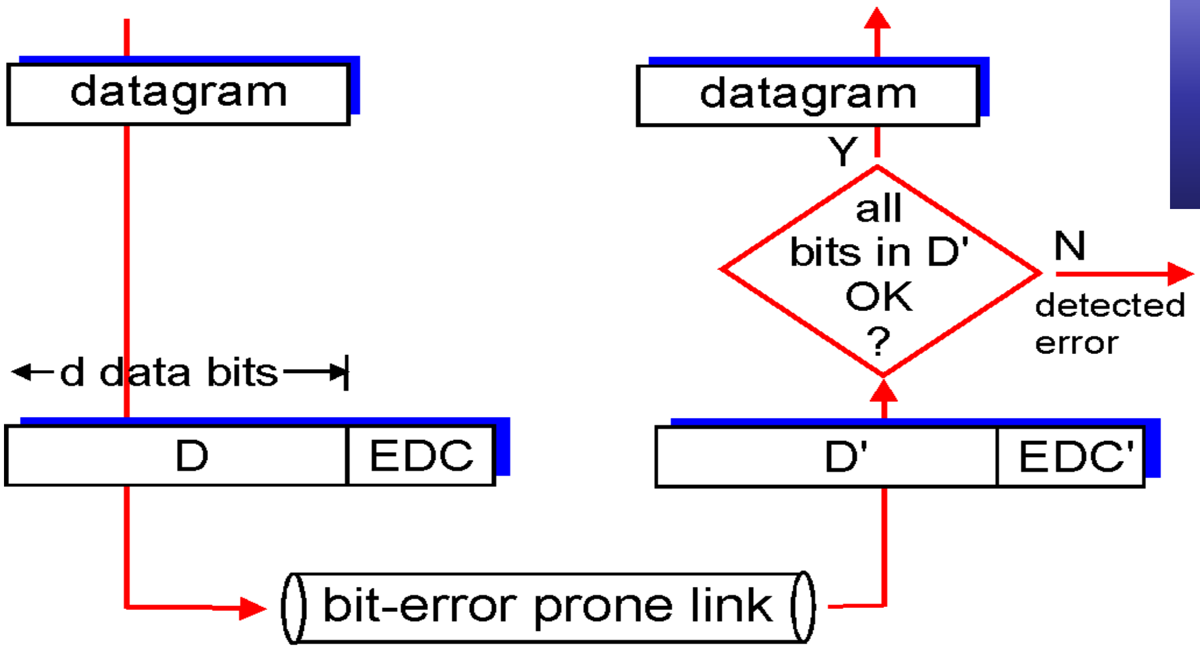
Error Detection vs Correction

- **Detection** vs **detection and correction**
- Detection-only has less overhead but requires retransmission. Acceptable when error rate is low.
- Detection and correction requires more overhead but avoids retransmission. Used often in wireless communication links which are naturally more error-prone than wired ones.

Error Detection and Correction

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

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



Redundancy does not always have to be at the end of the frame!

Different Error Detection and Correcting Codes

Error Detection Codes:

- Parity
- Checksums
- Cyclic Redundancy Checks (CRCs)

Error Correcting Codes:

- Hamming codes.
- Reed-Solomon codes.

Coding Basics

- Consider a frame consisting of *m* data bits (i.e.,message) and *r* check or parity bits.

- Let the total length of a block be $n = m+r$. This will be an *(n,m) code*

- In an *(n,m)* code, the n-bit unit containing data and check bits is referred to as an n-bit **codeword**.

- The **code rate**, or simply rate, is the fraction m/n.

Coding Basics

- In a **block code**, the in coming data stream is divided into fixed-size blocks. The *r* check bits are computed solely as a function of the *m* data bits.

- In a **systematic code**, the *m* data bits are sent directly, along with the check bits, rather than being encoded themselves before they are sent.

- In a **linear code**, the *r* check bits are computed as a linear function of the *m* data bits. Ex: **Exclusive OR (XOR)** or modulo2 addition is a popular choice for linear block codes.

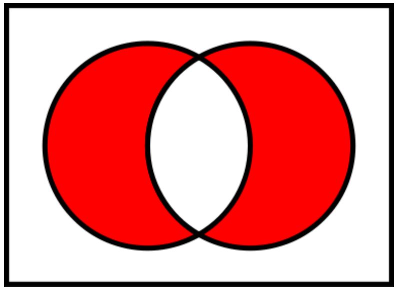
Review: XOR Operation

- XOR or "Exclusive OR" is a logic operation with the truth table shown.
- Consider it as "one or the other but not both," i.e. A or B excluding A and B.
- **It is equivalent to modulo 2 addition.**
- The XOR operation does not have a standard symbol but $\oplus$ is sometimes used to represent it.

XOR Truth Table

Input		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

***Note**: In modulo-2 arithmetic, there are no carries or borrows, and "+", "-", and "XOR" are equivalent operations!*



Hamming Distance

- The number of bit positions in which two codewords differ is called the **Hamming distance**. If two codewords are a Hamming distance ***d*** apart, then it requires ***d*** single-bit errors to convert one into the other.
- The error-detecting and error-correcting properties of a block code depend on its Hamming distance

$$\begin{array}{r} 10001001 \\ \underline{10110001} \\ 00111000 \end{array}$$

Question:

1. What is the Hamming distance between these two codewords?
2. How many bit errors can these two codewords tolerate (without any danger of confusing one for the other)?

Effectiveness of a code

- The effectiveness of a code is usually measured by three parameters
  - the minimum Hamming distance of the code
  - the burst-detecting capability
  - the probability that a completely random string will be accepted as error-free

- See e.g. Tannenbaum Chapter 3 for more coding information.

Error Detection Codes

- Parity
- Checksums
- Cyclic Redundancy Checks (CRCs)

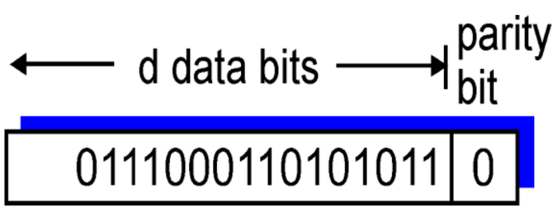
Parity Checks

Single Bit Parity:

Detect single bit errors

Even parity scheme: The sender includes one additional bit and choose its value such that the total number of 1s in the d+1 bits is even

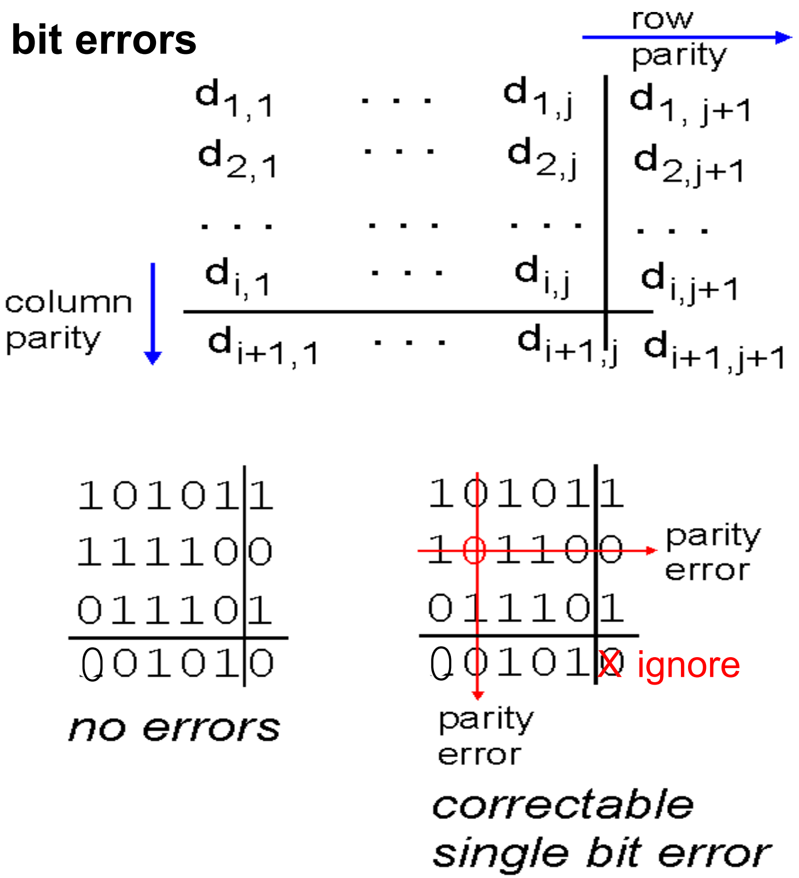
In case of an odd parity the situation is reversed



(example uses odd-parity)

Two Dimensional Bit Parity:

Detect and correct single bit errors



(example uses even-parity)

Question 1

Suppose the information content of a packet is the bit pattern 1010101010101011, and an even parity scheme is being used.

Suggest and analyse a minimum-length two-dimensional parity scheme.

Interleaving for burst error detection

- **Goal**: detect burst errors (multiple bit errors in one burst).
- **Idea**: **interleaving**, i.e. compute the parity bits over the data in a different order than the order in which the data bits are transmitted.

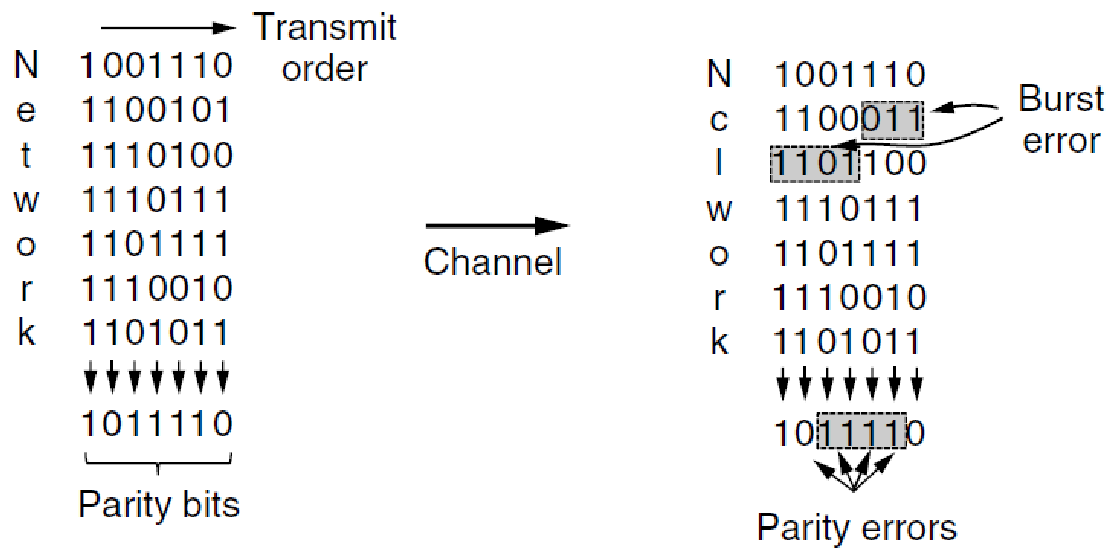


Figure 3-8. Interleaving of parity bits to detect a burst error.

Checksumming

Three steps:

- Break the original message into *k* blocks with *n* bits each
- Sum the *k* data blocks segments with any overflow encountered during the sum being wrapped around
- Perform 1's complement to the sum

Example:

We have a message that is 24 bits long

01100000 01010101 10001100

Let k = 3, n = 8.

<u>Sender:</u>

Step1:

0 1 1 0 0 0 0 0

0 1 0 1 0 1 0 1

1 0 0 0 1 1 0 0

Example:

Receiver:

All 8-bit words are added including the checksum

If no errors: 11111111

0 1 1 0 0 0 0 0

0 1 0 1 0 1 0 1

1 0 0 0 1 1 0 0

If one of the bits is 0: Errors have occurred

Example

Internet checksum (RFC 1071)

Goal: detect "errors" (e.g., flipped bits) in transmitted packet (widely used in networks and storage)

Sender:

- treat segment contents as sequence of 16-bit integers
- Performs the 1s complement of the sum of all the 16-bit words in the segment with any overflow encountered during the sum being wrapped around
- sender puts checksum value into checksum field

Receiver:

- 1's complement sum is calculated over the received data, including the checksum.
- If the result is all 1 bits:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?*

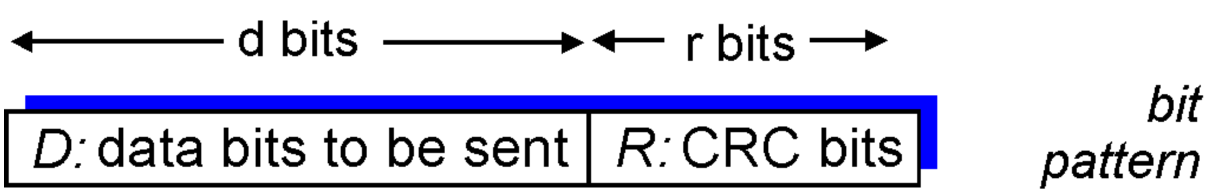
Cyclic Redundancy Checks

Let's say the data stream is *d* bits long. The divisor is *(r+1)* bits long

Steps in generating the CRC at the sender:

- Append *r* zeros to the original data stream
- Perform modulo 2 division
- The remainder is the CRC

- widely used in practice (Ethernet, 802.11 WiFi)
- Can detect burst errors up to the length *r*.



Example

Find the CRC for 101110 with the divisor 1001

Cyclic Redundancy Checks

Receiver:

Steps in the error detection using CRC:

- Using the received data and the divisor, perform modulo 2 division
- If the remainder is 0 – no transmission error

  1 – detects a transmission error

Question 3

Consider a CRC scheme as discussed above with a divisor 1001. What is the CRC if the message D is 10101010 ?

[optional] Repeat the question for D=10010001

Check your results using the methods discussed in the lecture.

***Note**: In modulo-2 arithmetic, there are no carries or borrows, and "+", "-", and "XOR" are equivalent operations!*

This specific CRC can detect 3 bit errors…

Question 4

A file is partitioned into many packets each with the length of 64 bits and sent over a communication link. The link bit error probability is 0.005. Assume no error correction is used at the receiver, thus retransmission is required for each packet when an error is detected.

How many times does each packet needs to be transmitted on average for the file to be successfully received?

Error Correcting Codes

- Hamming codes.

- Reed-Solomon codes.

Hamming Code

- In Hamming codes the bits of the codeword are numbered consecutively, starting with bit 1 at the left end, bit 2 to its immediate right, and so on.

- The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits. The rest (3, 5, 6, 7, 9, etc.) are filled up with the data bits.

Hamming Code

- Each check bit forces the modulo 2 sum, or parity, of some collection of bits, including itself, to be **even if even parity**

- Hamming code specifies how to calculate the check bits

- Example in a 7-bit Hamming code:

$$P1 = m3 + m5 + m7 \text{ (modulo 2 addition)}$$

$$P2 = m3 + m6 + m7$$

$$P3 = m5 + m6 + m7$$

Example

We want to send the 4-bit message 1001. Assuming even parity find the 7-bit Hamming codeword to be sent

Example

How can we correct this error?

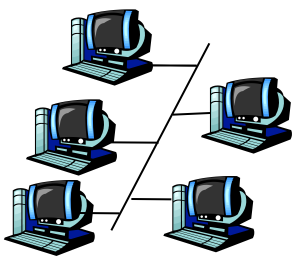
Media Access Control (MAC)

application
transport
network
link
physical

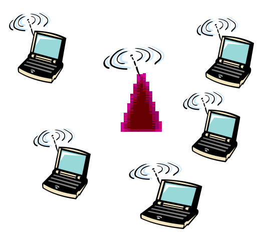
Multiple Access Links

Two types of "links":

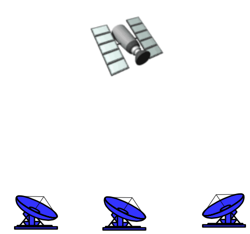
- point-to-point
  - DSL, plain old telephony
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - 802.11 wireless LAN
  - satellite links



shared wire (e.g., Cable access network)



shared RF (e.g., 802.11 WiFi)



satellite

humans at a cocktail party (shared air, acoustical)

Multiple Access Protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference leads to **collision** if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when a node can transmit
- *communication about channel sharing must use channel itself*!
  - no out-of-band `signaling' channel for coordination

Ideal Multiple Access Protocol

Given: broadcast channel of rate R bps

Wishlist:

1. when one node wants to transmit, it can send at rate R, (*efficiency*).
2. when M nodes want to transmit, each can send at average rate R/M (*fairness*)
3. *fully decentralized*:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. *simple*

MAC Protocols: a taxonomy

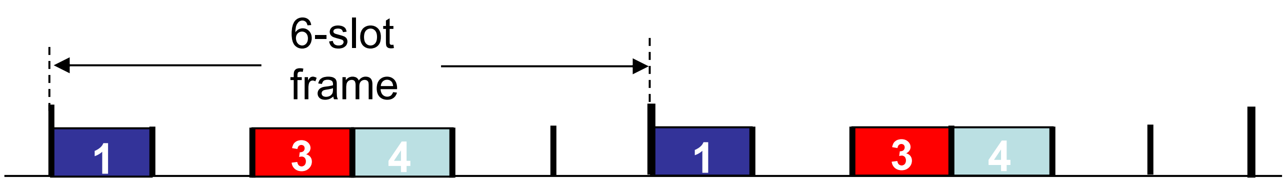
Three broad classes:

- Channel Partitioning
  - divide channel into smaller “pieces” (time slots, frequency, code)
  - allocate a piece to a 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

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

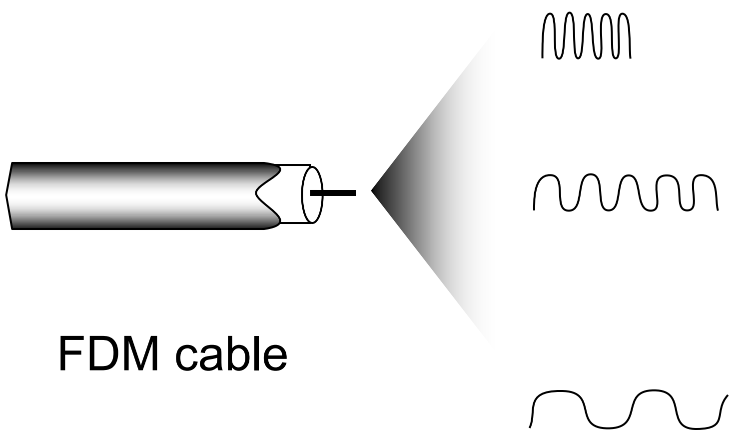
Example: 6-station LAN, 1,3,4 have packets, slots 2,5,6 idle

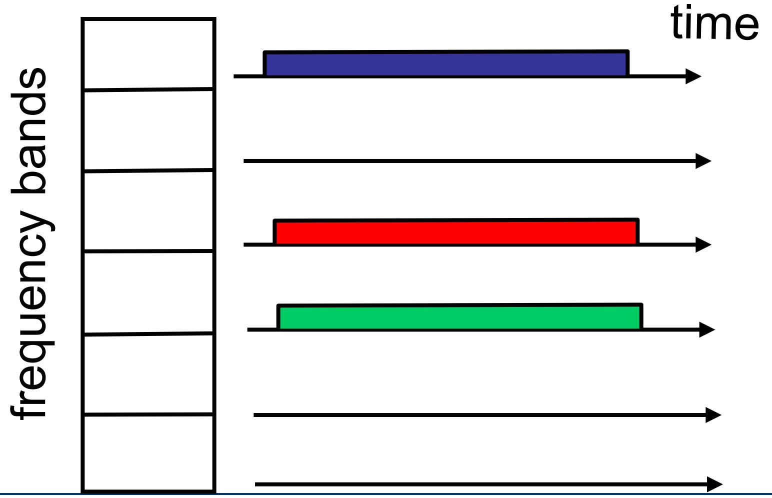


FDMA: frequency division multiple access

- 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 packets, frequency bands 2,5,6 idle





Random Access Protocols

- When a 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:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

Assumptions:

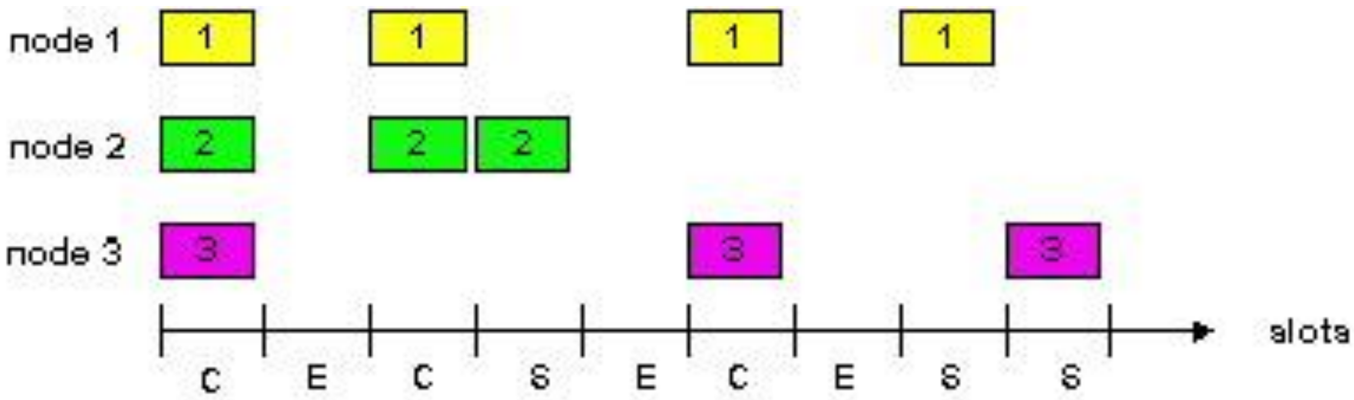
- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only at beginning of slot
- nodes are synchronized, i.e. there is a synced clock
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation:

- when node obtains a fresh frame, tries to transmit it in next slot
  - *if no collision:* it is sent successfully
  - *if collision:* collision detected, **retransmits frame in each subsequent slot with probability p until success**

THE UNIVERSITY OF MELBOURNE

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 waste slots
- idle slots
- Increased delay for users
- clock synchronization
- Inefficiency at high loads

Slotted Aloha efficiency

Efficiency : long-run fraction of successful slots (many nodes, all with many frames to send)

- *suppose:* N nodes with many frames to send, each transmits in slot with probability p
- prob that given node has success in a slot = $p(1-p)^{N-1}$
- prob that *any* node succeeds: $\boldsymbol{P_{success} = Np(1-p)^{N-1}}$

- max efficiency: find p* that maximizes $Np(1-p)^{N-1}$
- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

Max efficiency = **1/e = 0.37**

At best: channel used for useful transmissions 37% of time!

!

Question

Complete the derivation of slotted Aloha efficiency.

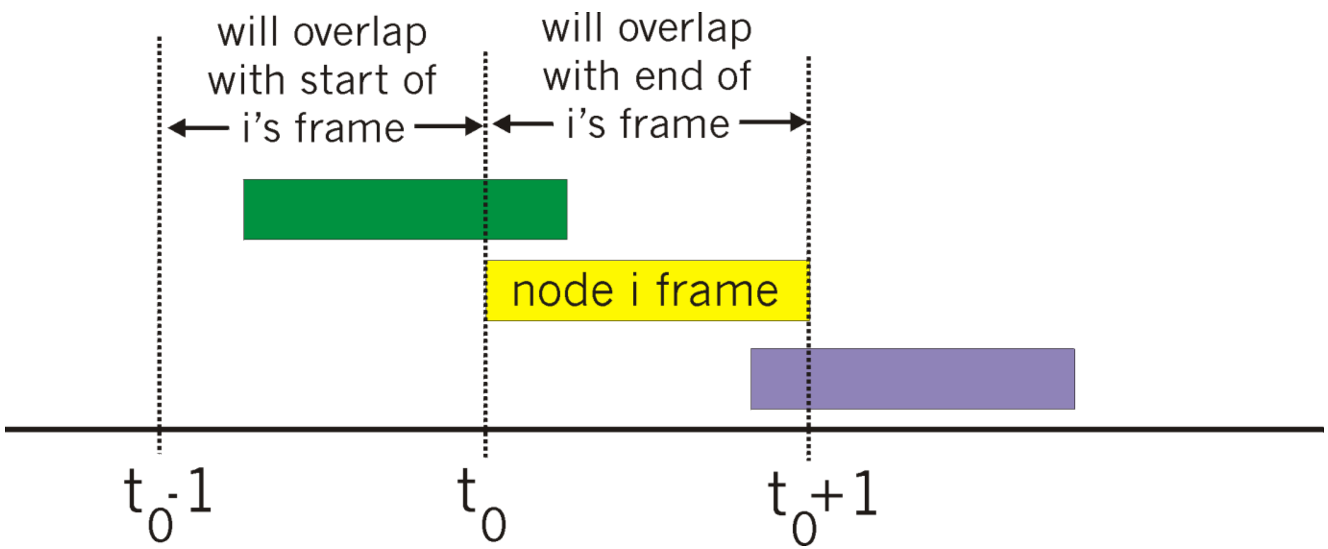
a) Find the probability p* that maximizes the probability that *any* node has a success

$$p^* = \arg\max Np(1-p)^{N-1}$$

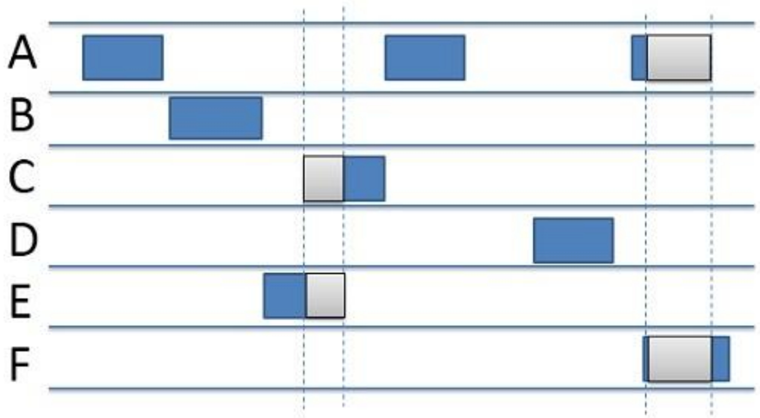
b) Using the result in part (a), find an upper-bound on efficiency. *Hint: it is obtained as a limit as N goes to infinity.*

Pure (unslotted) ALOHA

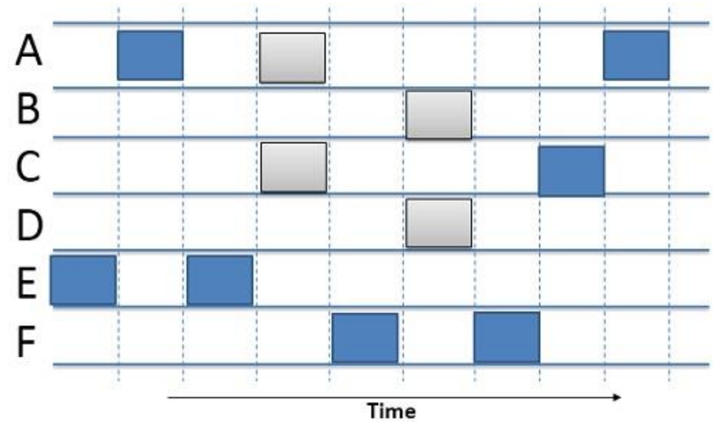
- Unslotted Aloha: simpler, no synchronization
- when frame first arrives
  - transmit immediately
- **collision probability increases**:
  - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



Pure vs Slotted ALOHA



Pure ALOHA



Slotted ALOHA

Main Differences

- Slotted Aloha performs better (throughput, success prob.)
- However, slotted Aloha requires time synchronisation, necessary for frames.

Images https://techdifferences.com/difference-between-pure-aloha-and-slotted-aloha.html

Pure Aloha efficiency

P(success by given node) = P(node transmits) ·

P(no other node transmits in $[t_0\text{-}1, t_0]$) ·

P(no other node transmits in $[t_0, t_0+1]$)

$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$

$= p \cdot (1-p)^{2(N-1)}$

… choosing optimum p and then letting N → infinity...

= 1/(2e) = .18

even worse than slotted Aloha!

Question

3 active nodes A, B, C are competing in a slotted ALOHA system. Assume each node has always a packet to send. The slots are numbered as 1, 2, 3, ….

Each node attempt to transmit in each slot with probability *p*.

1. What is the probability that Node A succeeds for the first time in Slot 4?
2. What is the probability that one of the nodes (A or B or C) succeeds in Slot 2?

ALOHA Throughput Analysis

- Let stations generate frames according to a Poisson distribution with a mean of **G** frames per frame time.
- The throughput, $\mathbf{S=GP_0}$, where $\mathbf{P_0}$ is the probability that a frame does not suffer a collision.
- The *probability that k frames are generated during a given frame* time is

$$\Pr[k] = \frac{G^k e^{-G}}{k!}$$

ALOHA Throughput Analysis

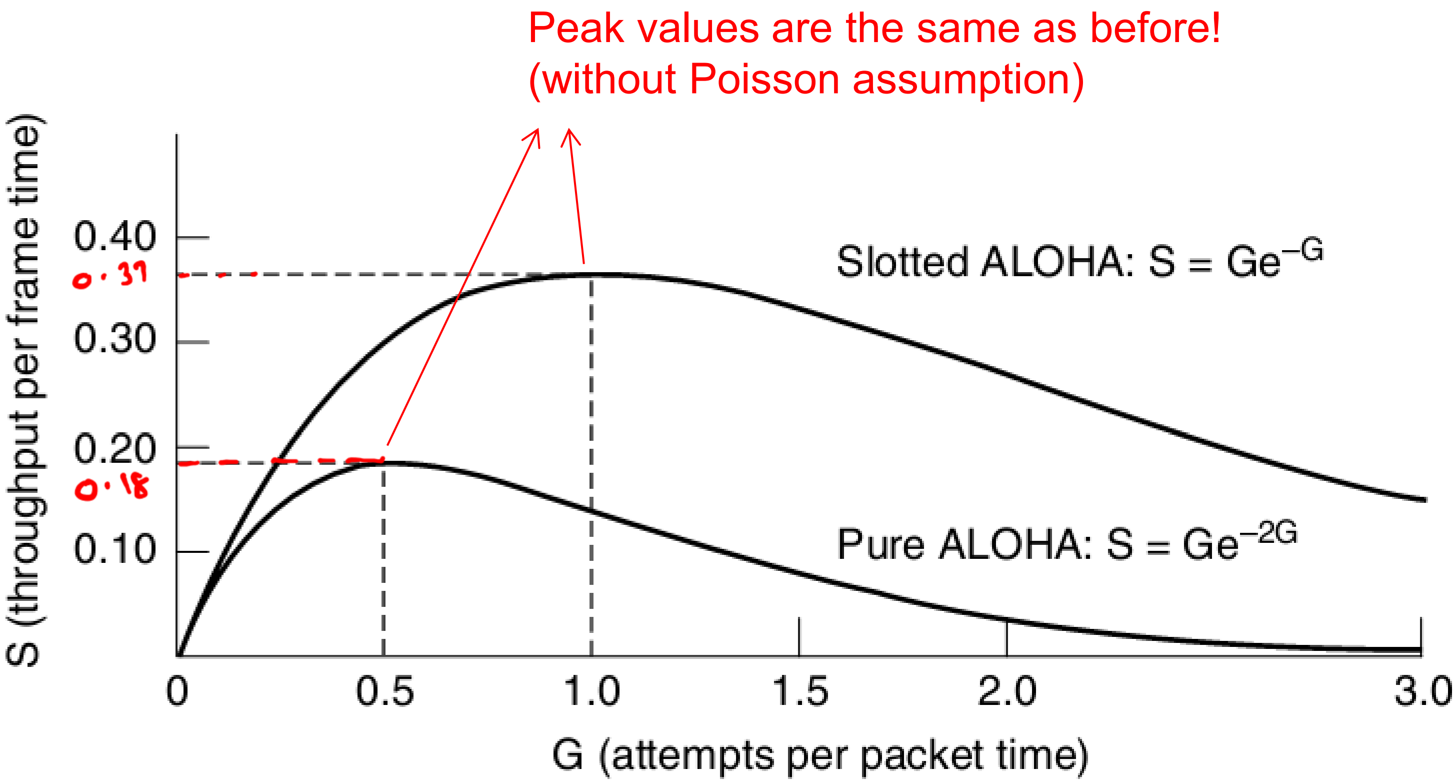
- In an interval two frame times long, the mean number of frames generated is **2G**.
- The probability of no frames being initiated during the entire vulnerable period is thus given by $\mathbf{P_0=e^{-2G}}$.
- Thus, the ***unslotted ALOHA throughput*** is:

$$\boxed{S = Ge^{-2G}}$$

- In ***slotted ALOHA***, The probability of no other traffic during the same slot is $\mathbf{P_0=e^{-G}}$.
- Thus, the ***slotted ALOHA throughput*** is:

$$\boxed{S = Ge^{-G}}$$

ALOHA Throughput Analysis



Throughput versus offered traffic for ALOHA systems <u>under Poission traffic assumption</u>.

CSMA (Carrier Sense Multiple Access)

CSMA: listen before transmit:

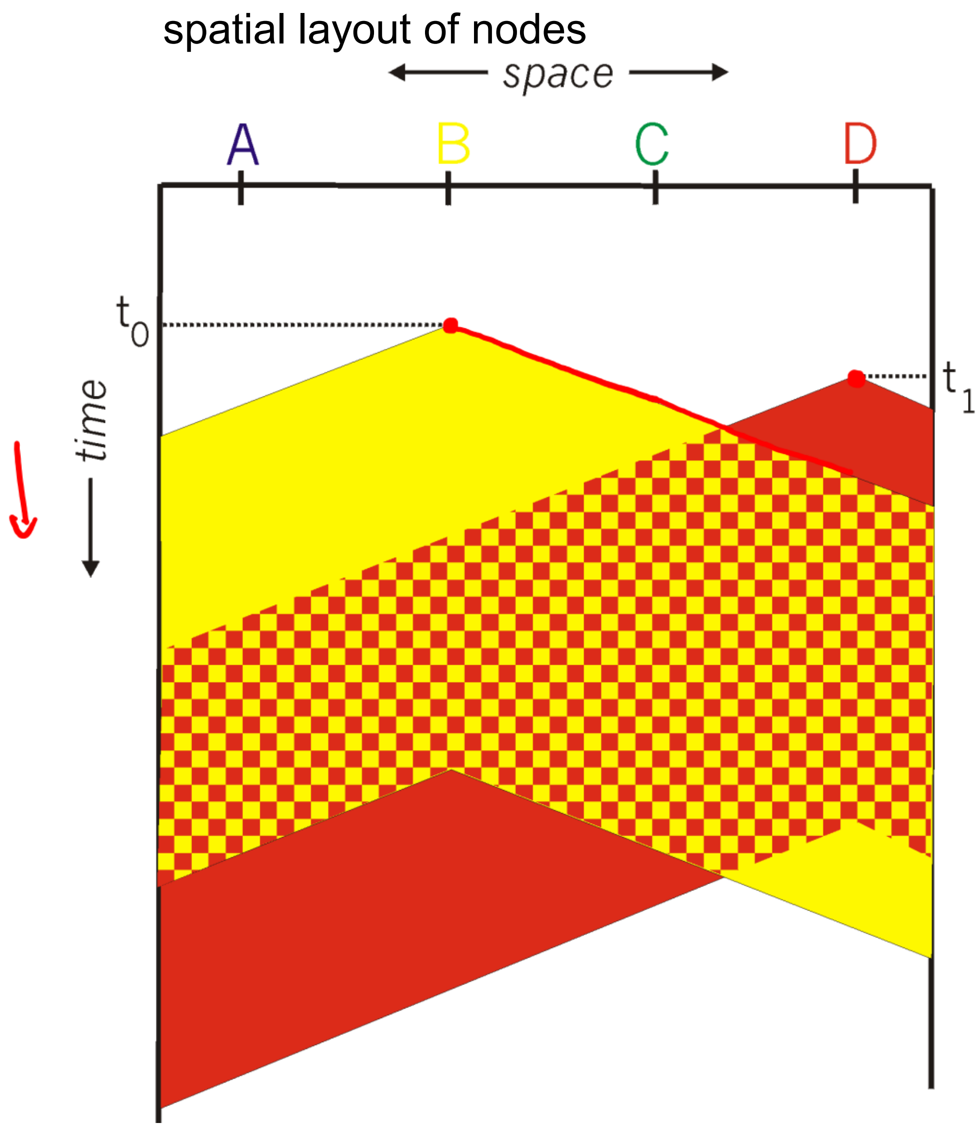
- If channel sensed idle: transmit entire frame
- If channel sensed busy, defer transmission

- human analogy: don't interrupt others!

CSMA Variants

- **1-persistent**: if the channel is idle, then send the data.
- **Non-persistent**: if the channel is already in use, do not continually sense it; wait a random period of time and then repeat the algorithm.
- **p-persistent**: (slotted system) if the channel is idle, transmit with a probability p; wait for next slot with probability 1-p.

CSMA collisions



collisions can still occur:

propagation delay means two nodes may not hear each other's transmission

collision:

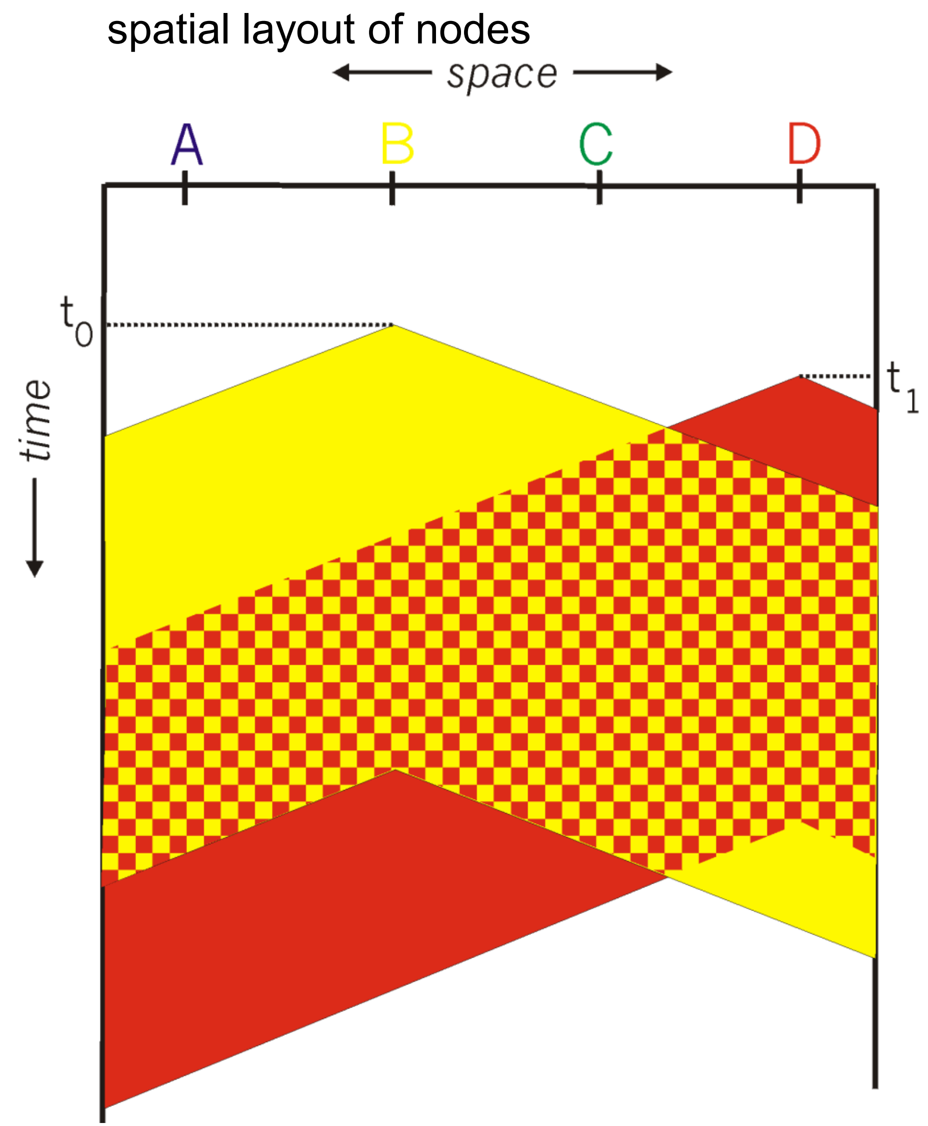
entire packet transmission time wasted

note:

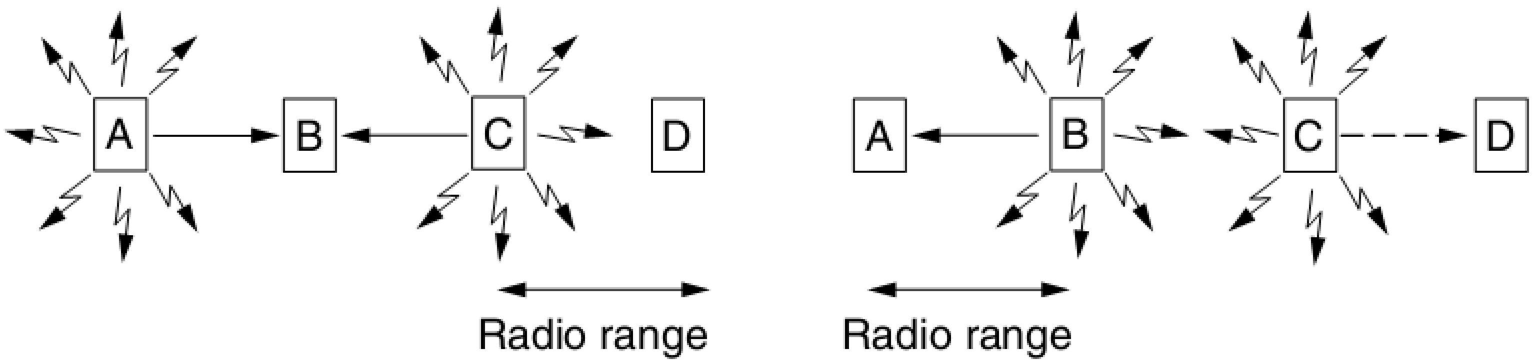
role of distance & propagation delay in determining collision probability

Question

What is the worst-case scenario for collision when the farthest nodes have a signal propagation distance of τ seconds?



Hidden Terminal Problem



(a) (b)

A wireless LAN and two different scenarios as follows:
(a) A and C are **hidden terminals** when transmitting to B.
(b) B and C are exposed terminals when transmitting to A and D .

The problem of a station not being able to detect a potential competitor for the medium because the competitor is too far away is called the **hidden terminal problem**.

CSMA/CD (Collision Detection)

CSMA/CD:

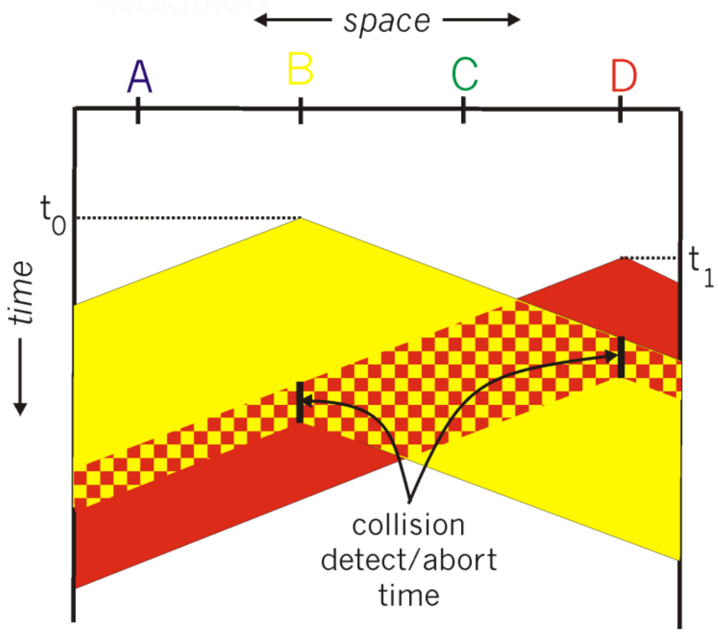
- Transmit and monitor
- If a collision is detected immediately stops transmission : reduce channel wastage
- Sends a jam signal: inform other stations
- Random back-off and retransmission

Collision detection:

- easy in wired LANs: measure signal strengths, compare transmitted, received signals
- difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

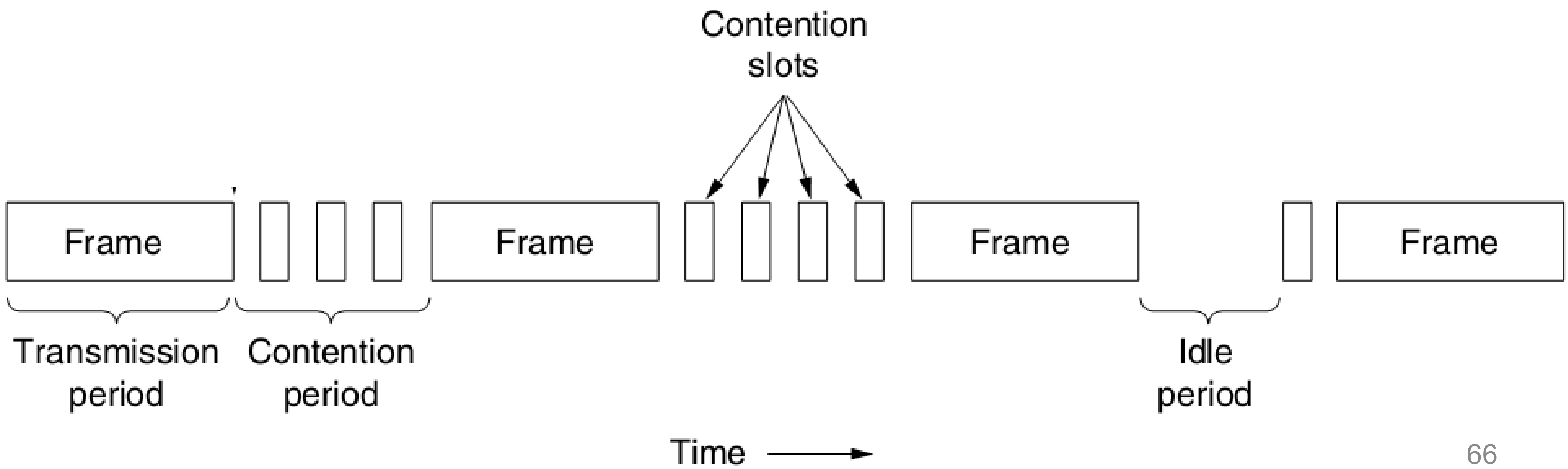
human analogy: the polite conversationalist

CSMA/CD collision detection



The minimum time to detect the collision is just the time it takes the signal to propagate from one station to the other.

CSMA/CD can be in contention, transmission, or idle state.



CSMA/CA (Collision Avoidance)

CSMA/CA:

- Carrier sense
- If the channel is idle wait a predefined period of time (Interframe space (IFS). Different types of traffic have different IFS to allow prioritization.
- After IFS, station enters a random back-off time.
- When the back-off counter is zero transmit
- Wait for Ack

"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

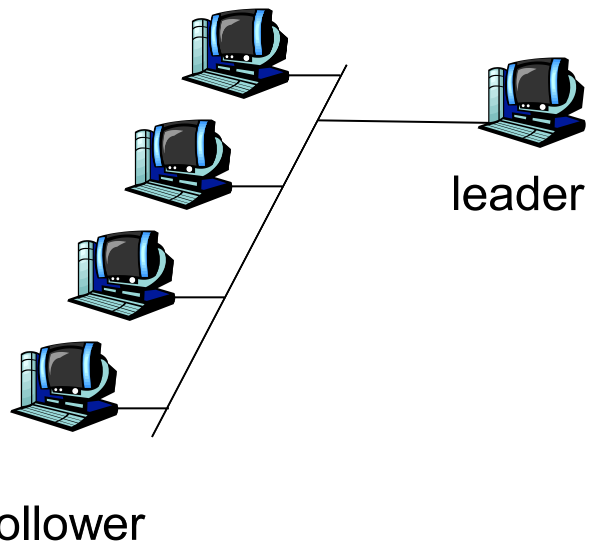
"Taking turns" protocols

look for best of both worlds!

"Taking Turns" MAC protocols

Polling:

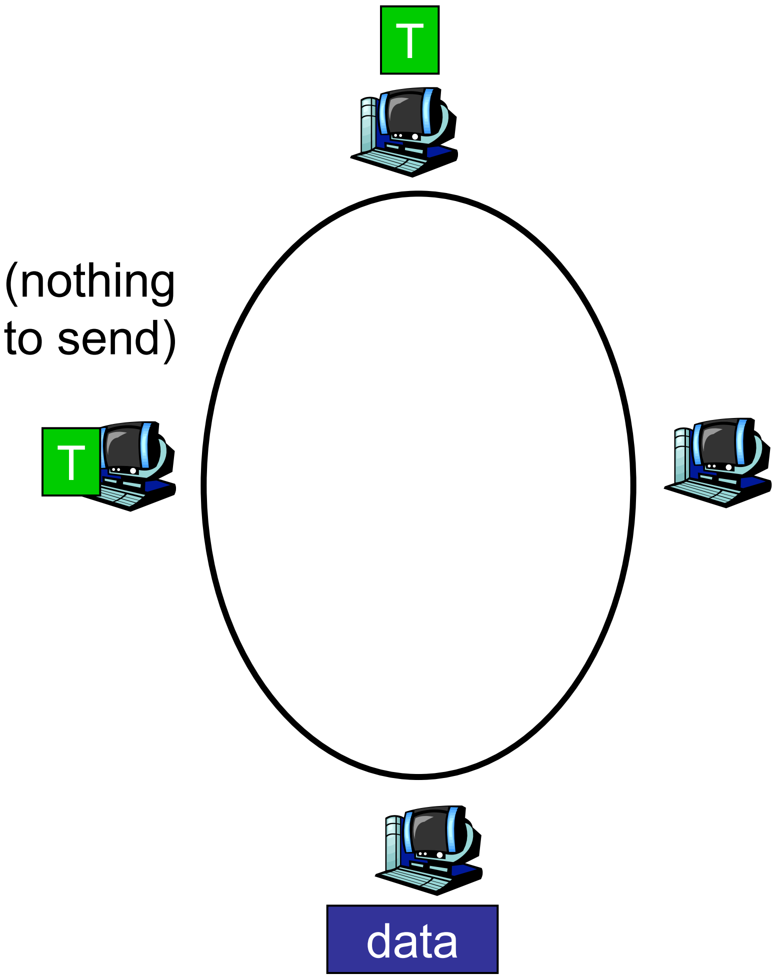
- leader node "invites" follower nodes to transmit in turn
- typically used with "dumb" follower devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (leader)
- *Not widely used these days due to overhead (fast computers!)*



"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)



Summary of MAC protocols

- *channel partitioning,* by time, frequency or code
  - Time Division, Frequency Division
  - Ex: GSM
- *random access* (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wired), hard in others (wireless)
  - Ex: CSMA/CD used in Ethernet, CSMA/CA used in 802.11.
- *taking turns*
  - polling from central site, token passing
  - Ex: Bluetooth, FDDI, token ring

THE UNIVERSITY OF MELBOURNE

MAC Addressing

application
transport
network
link
physical

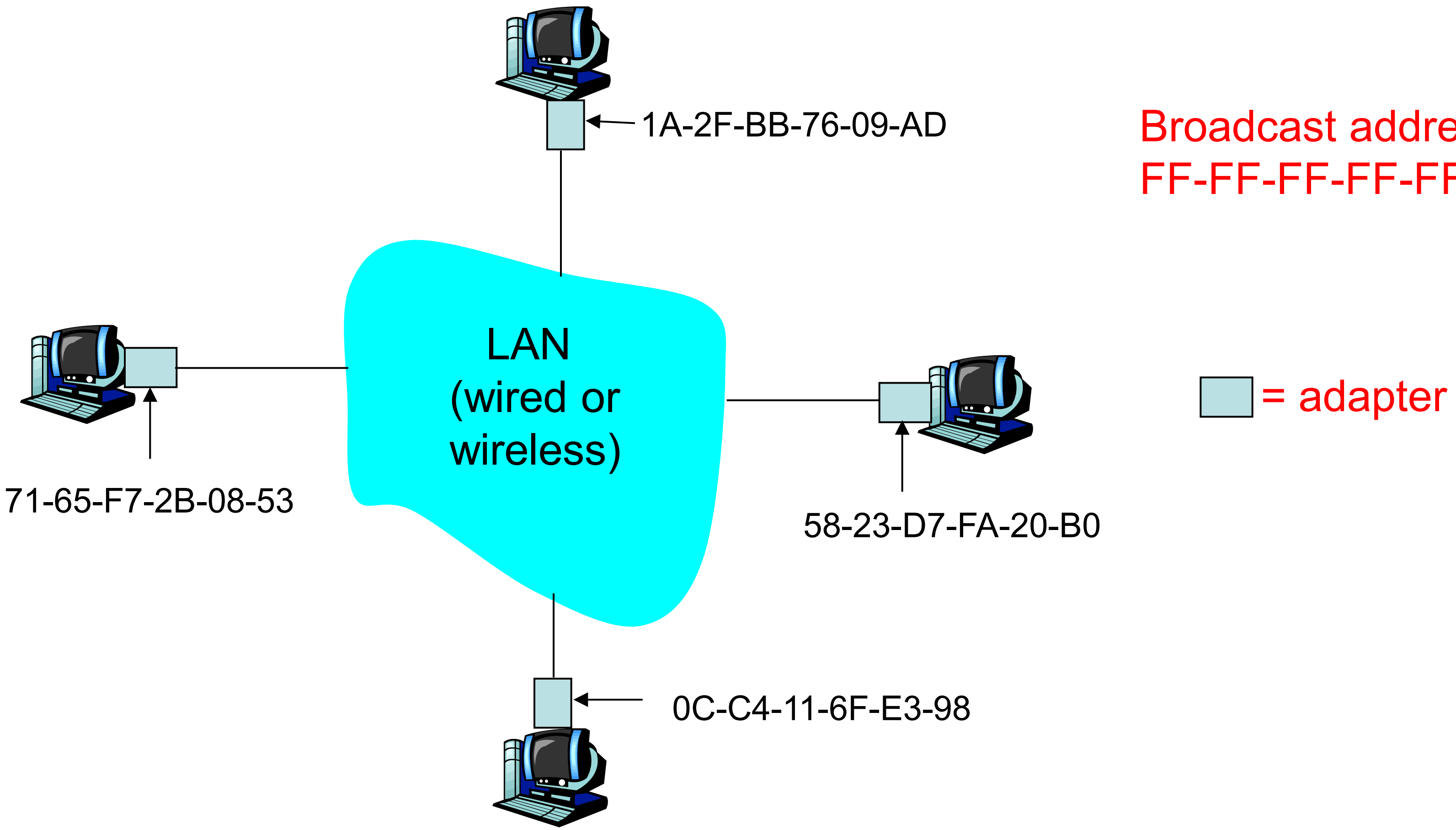
MAC Addresses

- 32-bit IPv4 address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- MAC (or LAN or physical or Ethernet) address:
  - function: *used locally to get frame from one interface to another physically-connected interface (same IP subnetwork)*
  - 48 bit MAC address (for most LANs)
    - burned in NIC ROM, also sometimes software configurable
    - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each digit {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F} represents 4 bits)

MAC Addresses

Each adapter on LAN has unique MAC (or LAN) address



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

MAC Addresses

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- **Analogy**:

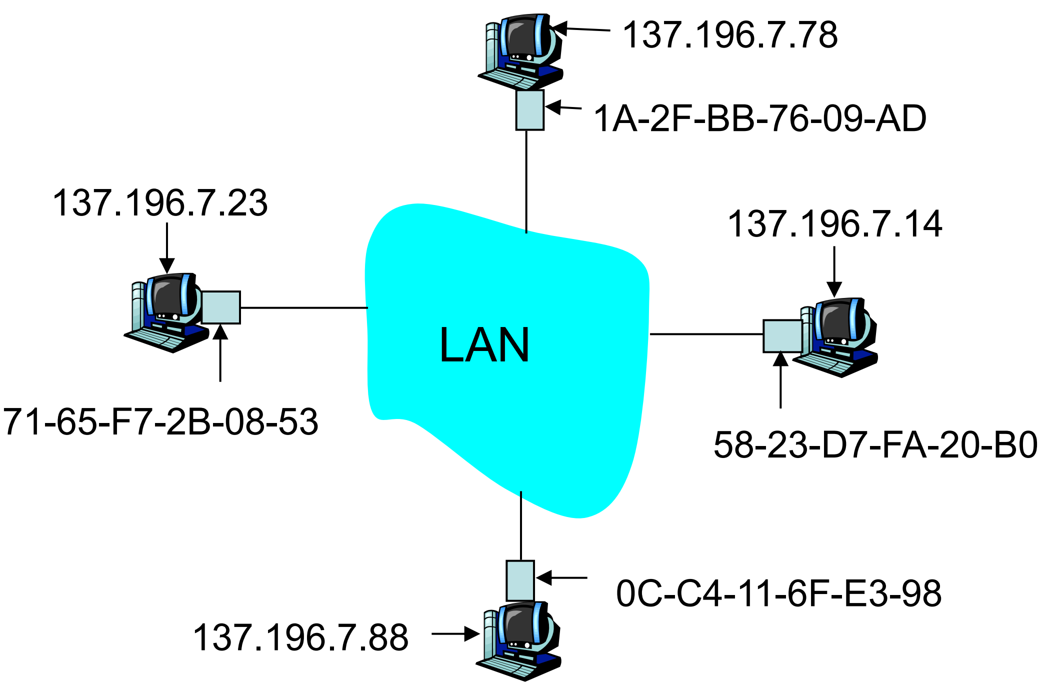
  (a) MAC address: like Citizenship Number or Tax File Number (permanent)

  (b) IP address: like postal address (temporary)

  Both are global!
- MAC flat address → portability
  - can move device (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 an interface's MAC address knowing its IP address?



- **Each IP interface (host, router) on LAN has an 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)

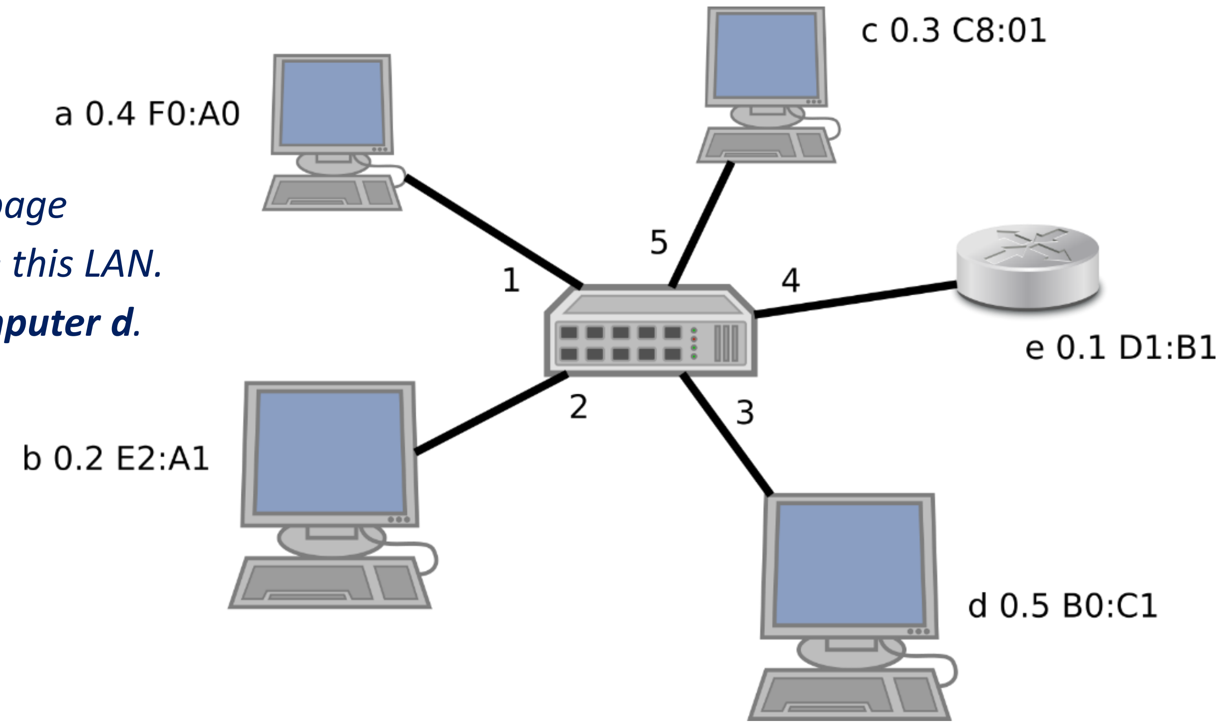
- A wants to send datagram to B, and B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - dest MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- ARP queries and responses have same format!
- 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*
- ARP is at the boundary of Layers 2 and 3!
- *ARP cache is maintained by the operating system!*

Question

An Ethernet LAN is shown below. The NICs of the **Computers** and the **Router** on the right are identified by *(ID, IP address, MAC address)*. For simplicity, only the last parts of the IP and MAC addresses are shown. The interfaces of the Ethernet **Switch** in the middle are clearly labelled as (1, 2, 3, 4, 5). For example, **(a 0.4 F0:A0)** means, **Computer a** has *IP address* **0.4** and *MAC address* **F0:A0**. **Router e** is connected to the **Switch** via *Interface* **4**.

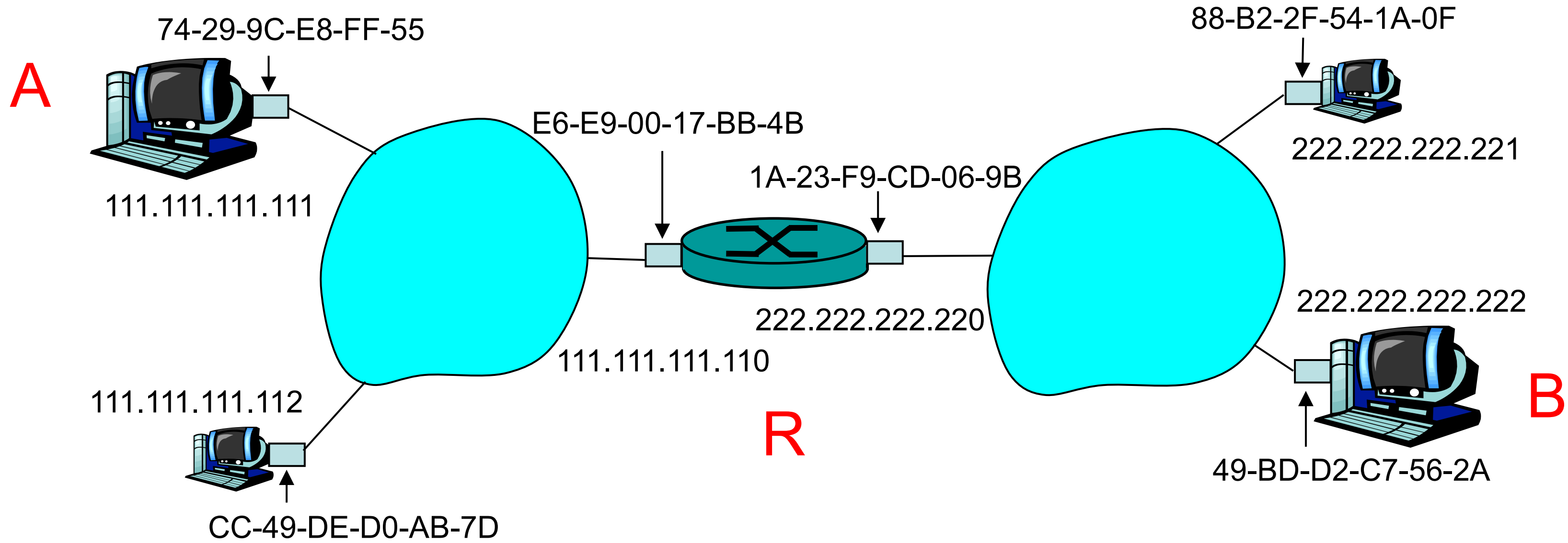
Provide the relevant *ARP tables*, all filled with the appropriate entries <u>after the following events are completed</u>:

1. *Initially, all the ARP tables are empty.*
2. ***Computer a** downloads a webpage from an external server outside this LAN.*
3. ***Computer b** sends a file to **Computer d**.*



Addressing: routing to another LAN

walkthrough: send datagram from A to B via R
assume A knows B's IP address

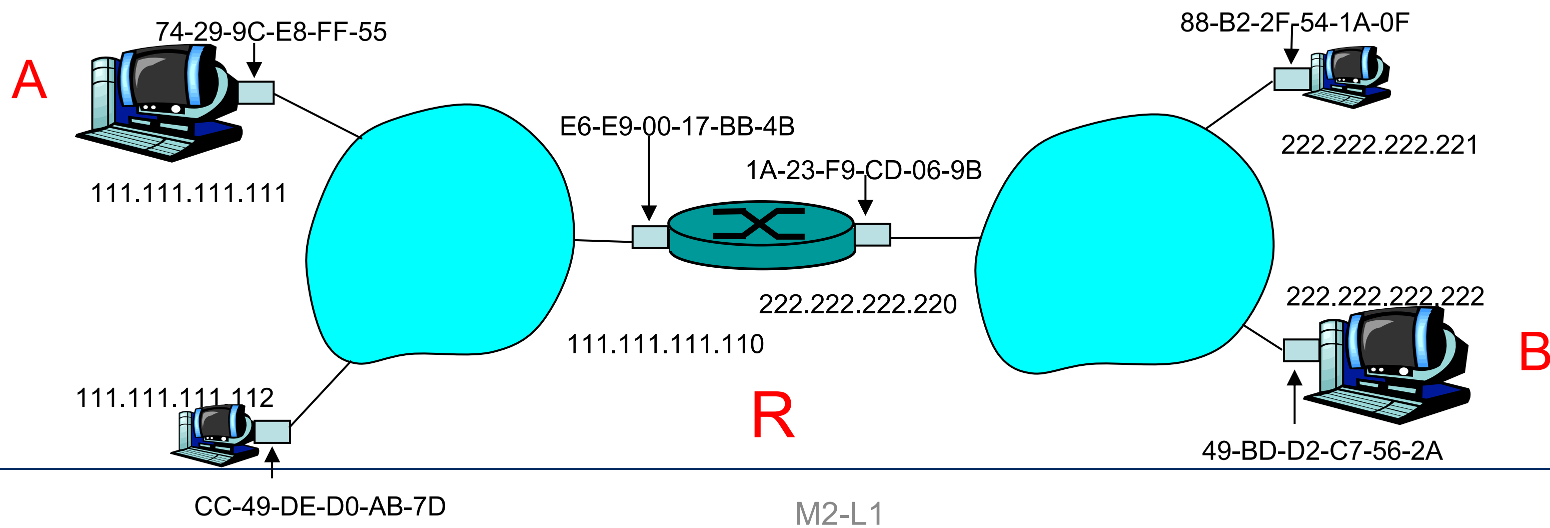


two ARP tables in router R, one for each IP network (LAN)

Addressing: routing to another LAN

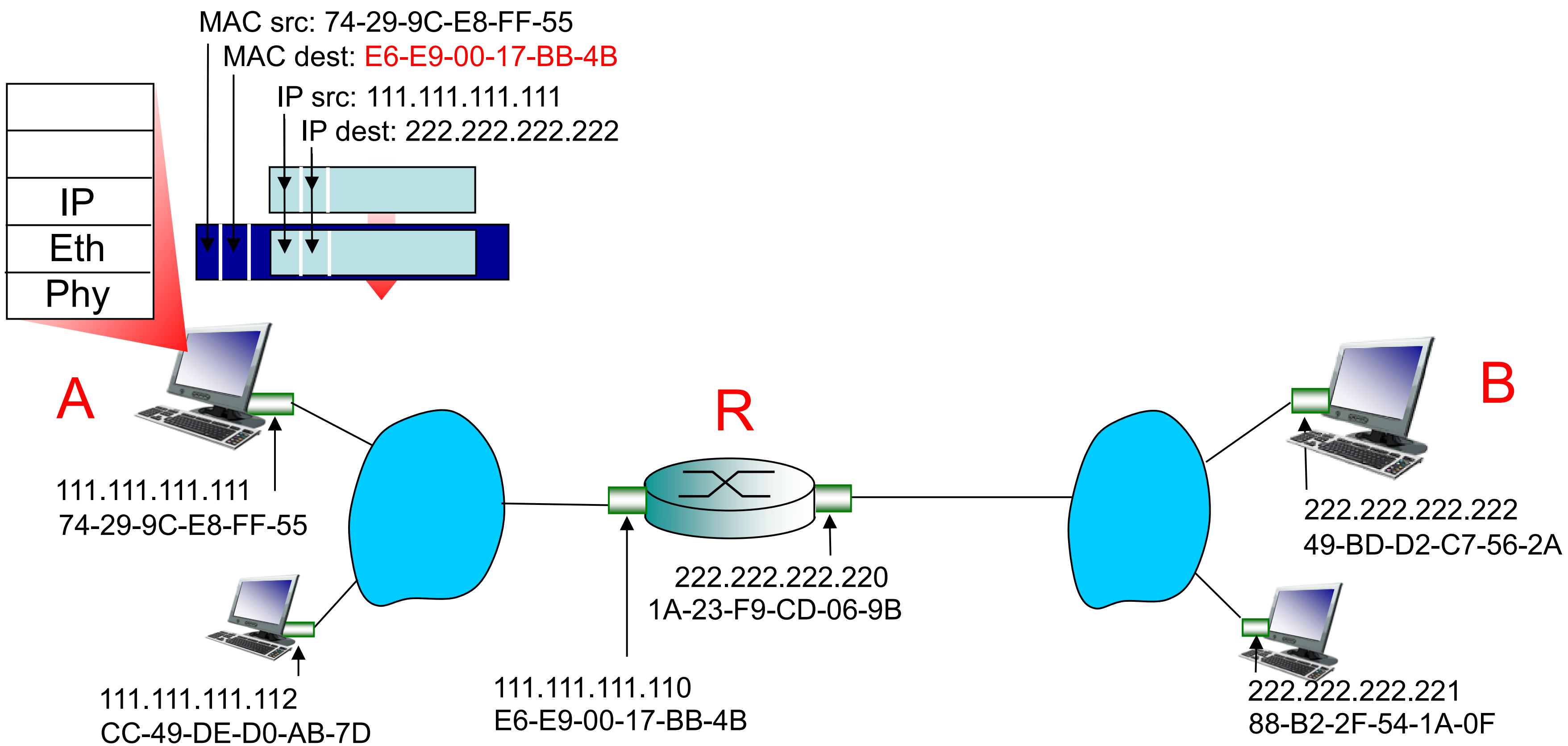
- A creates IP datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as destination, frame contains A-to-B IP datagram
- A's NIC sends frame
- R's NIC receives frame
- R removes IP datagram from Ethernet frame, sees destination is B, uses forwarding table to select output interface
- R uses ARP table on outward interface to get B's MAC address
- R creates frame re-encapsulating IP datagram, sends to B

Network Layer



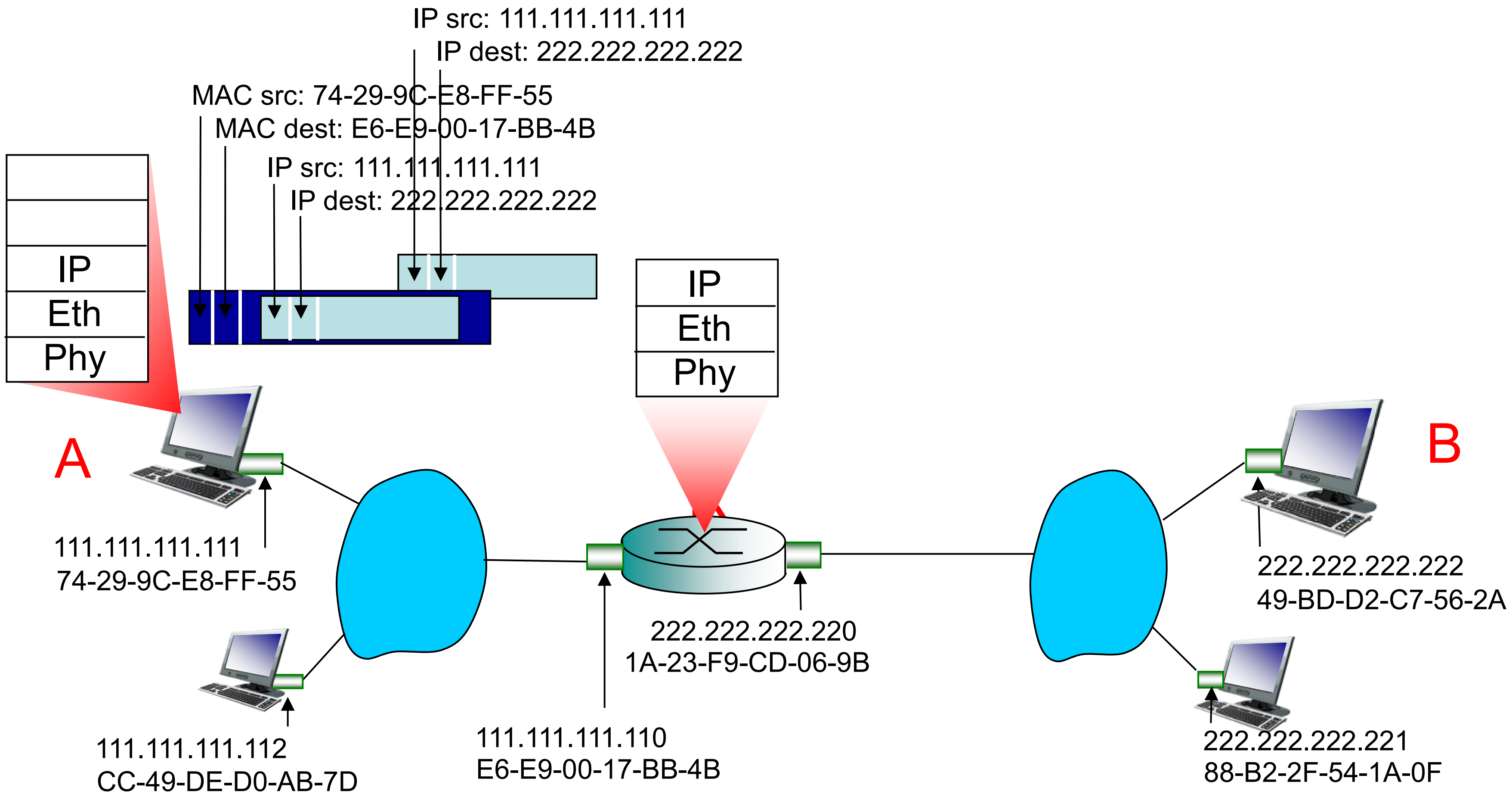
Addressing: routing to another LAN

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



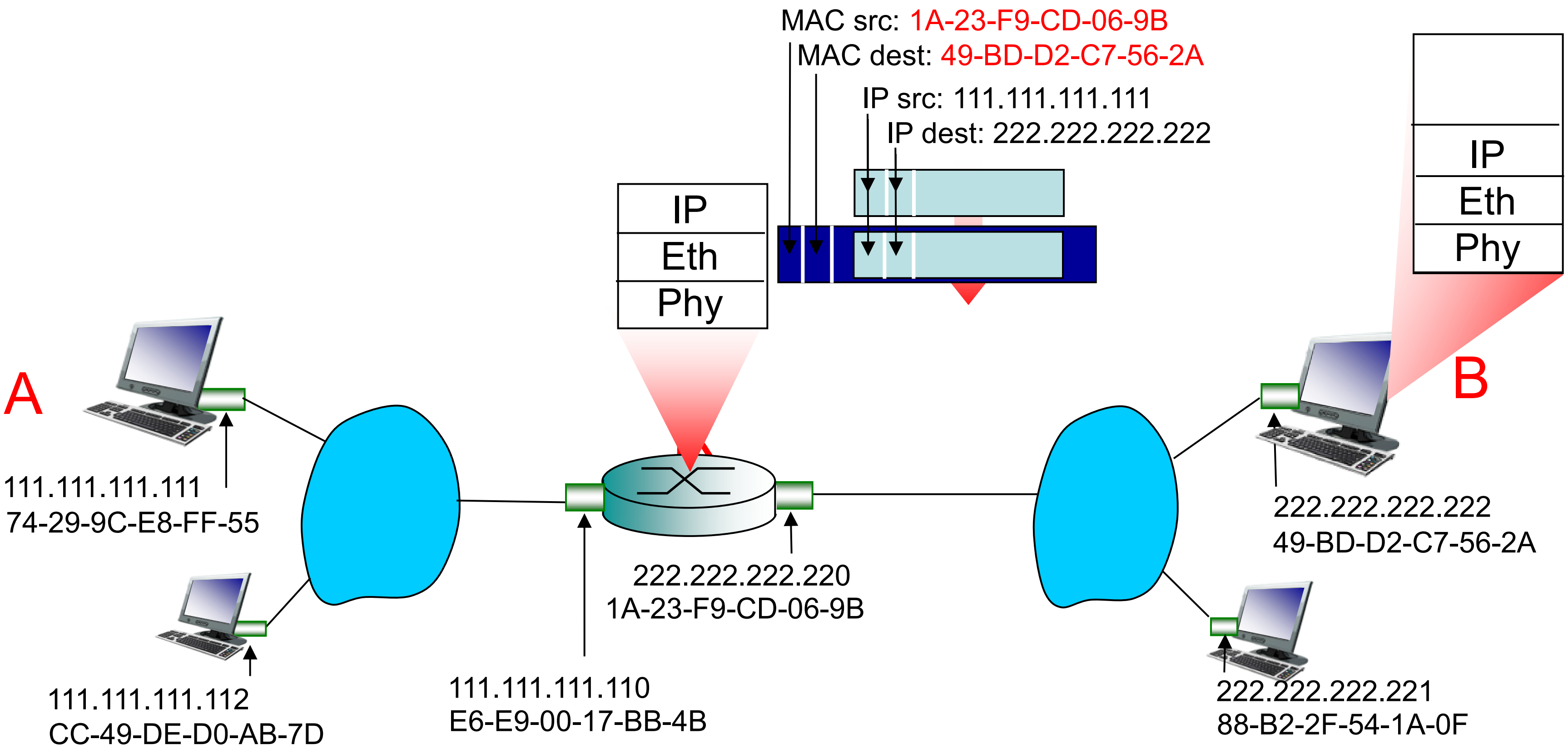
Addressing: routing to another LAN

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



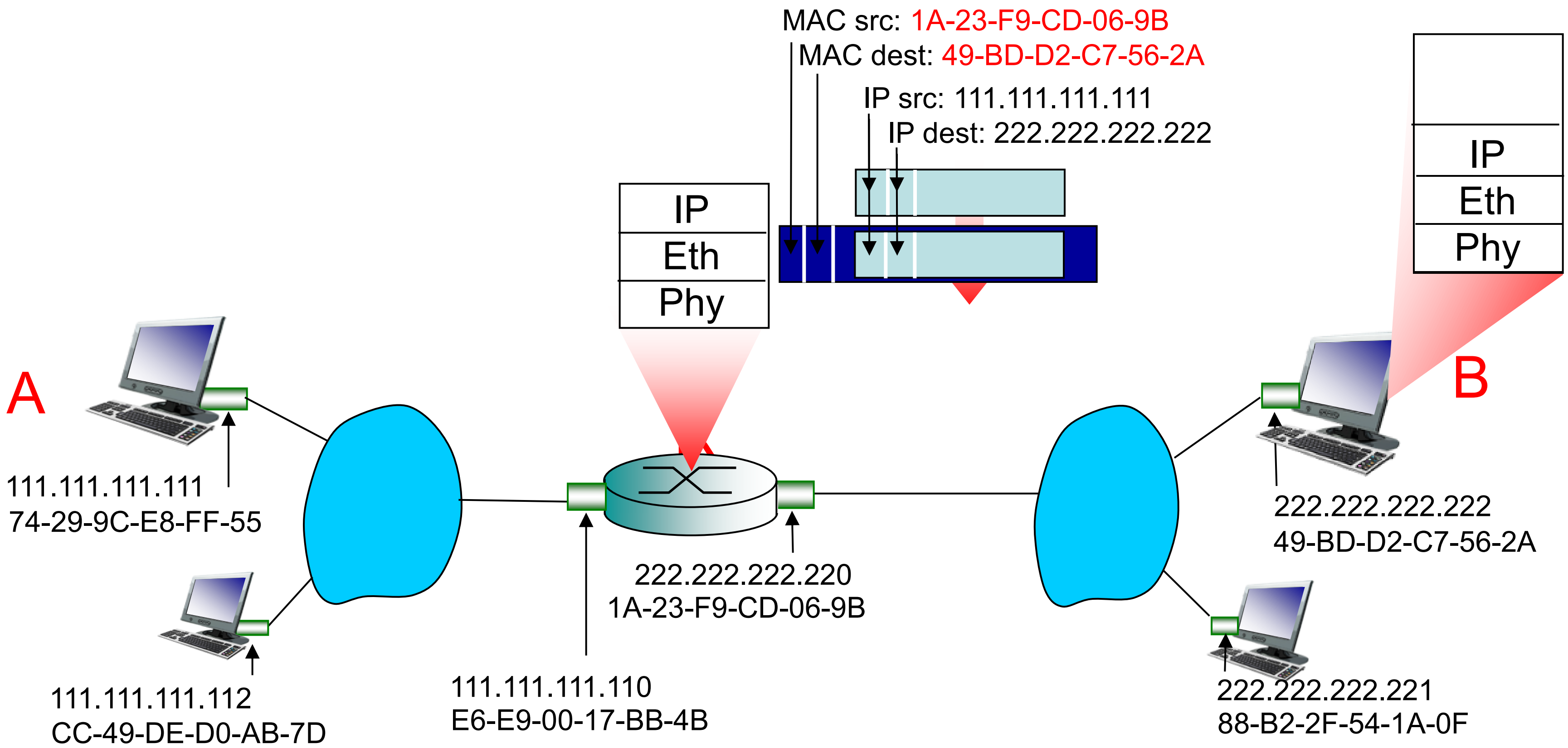
Addressing: routing to another LAN

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



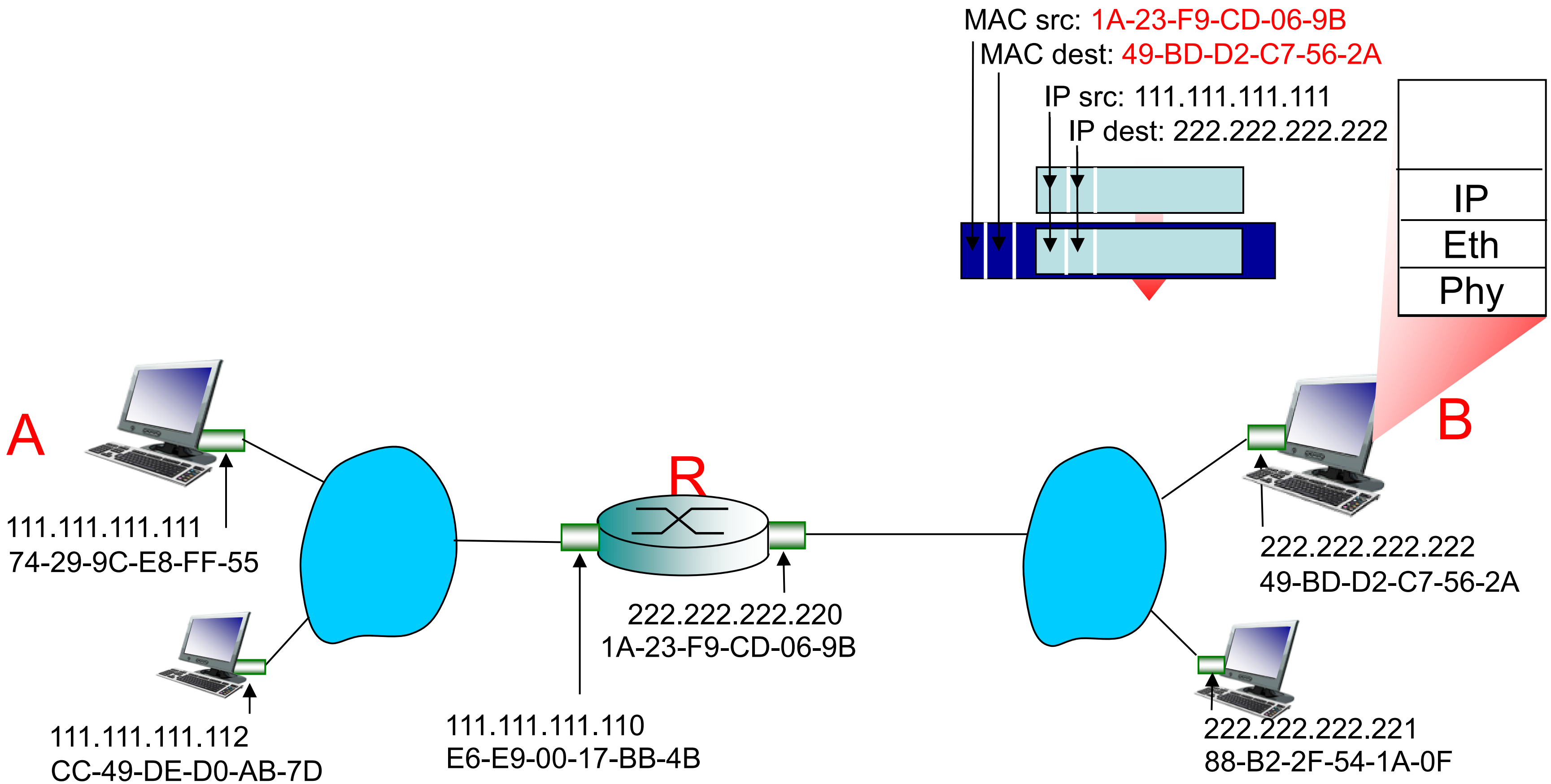
Addressing: routing to another LAN

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



Learning Objectives

You should now know about

- Link Layer
- Error detection and correction
- Media Access Control (MAC) and MAC protocols
- Aloha, CSMA, collision detection, hidden terminal problem
- Aloha throughput analysis
- MAC Addressing