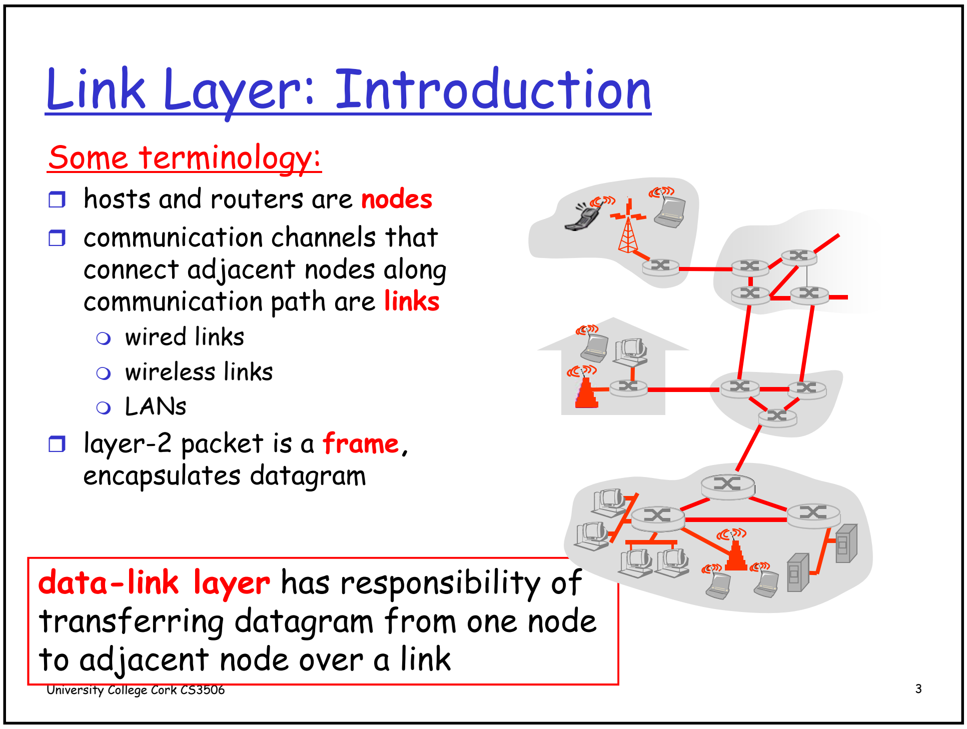
Data Link Layer

Our goals:

- understand principles behind data link layer services:
  - Link-later addressing
  - error detection
  - sharing a broadcast channel: multiple access
  - reliable data transfer, flow control: *covered in CS2505*
- instantiation and implementation of various link layer technologies

Link Layer

- Introduction and services
- Link-Layer Addressing
- Error Detection
- Multiple Access
- Ethernet
- Link-layer switches
- Link virtualization: MPLS
- Data Centre Networks
- A day in the life of a web request

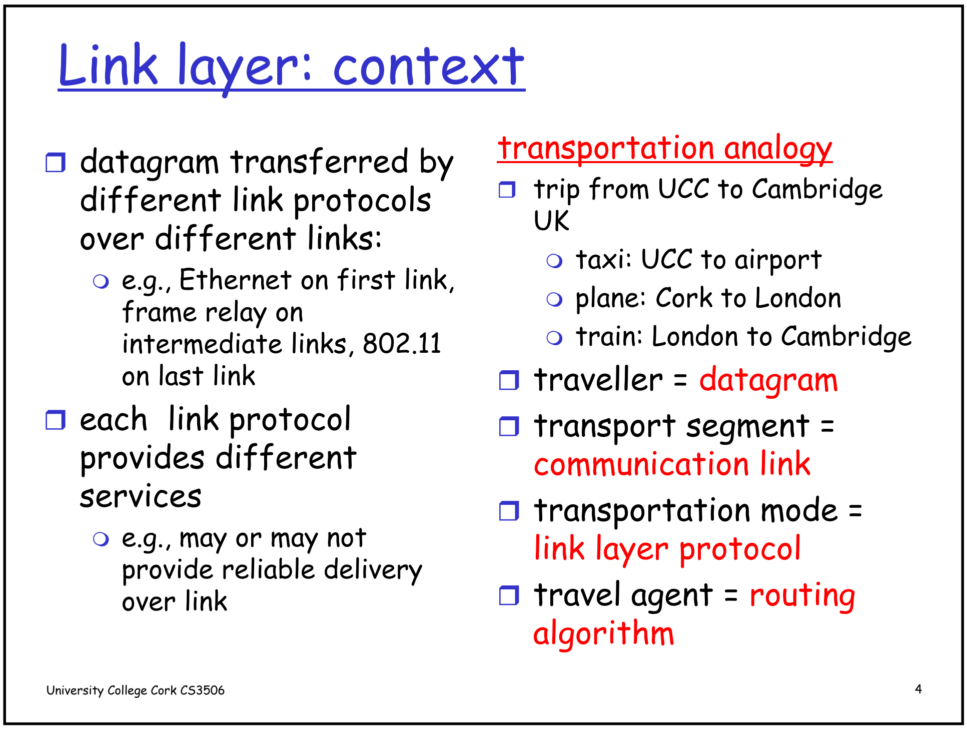


Link Layer: Introduction

Some terminology:

- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- layer-2 packet is a **frame**, encapsulates datagram

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



Link layer: context

- datagram 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 delivery over link

transportation analogy

- trip from UCC to Cambridge UK
  - taxi: UCC to airport
  - plane: Cork to London
  - train: London to Cambridge
- traveller = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

Link Layer Services

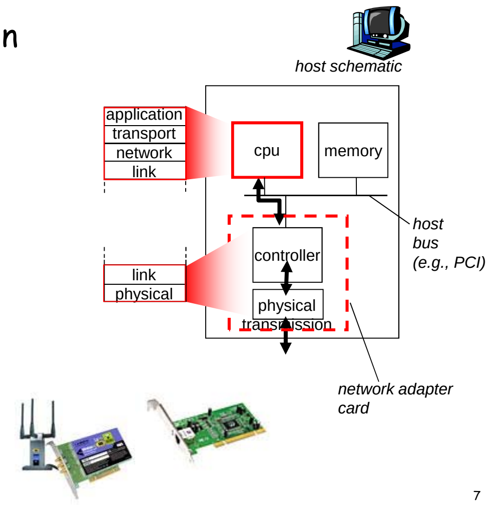
- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if it is a shared medium
  - "MAC" addresses (AKA physical addr) used in frame headers to identify source, dest
    - different from IP address!
    - MAC = Medium Access Control (feature of Link Layer)
- *reliable delivery between adjacent nodes*
  - we learned how to do this already (CS2505)
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates
    - Q: why both link-level and end-end reliability?

Link Layer Services (more)

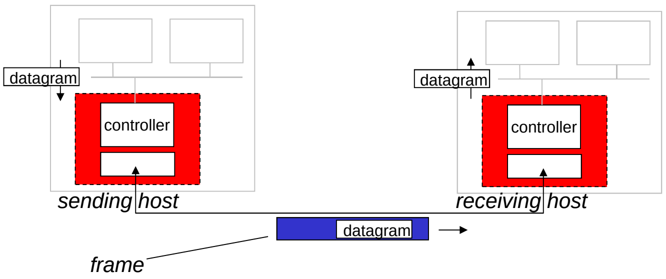
- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- error correction:
  - receiver identifies *and corrects* bit 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?

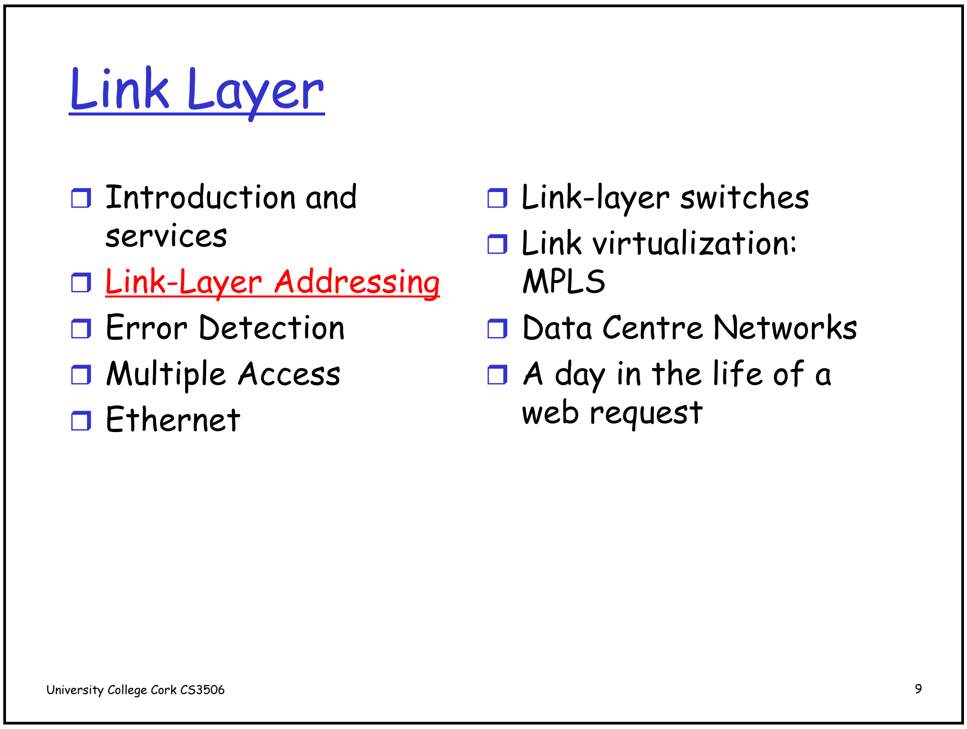
- ❒ in each and every host
- ❒ link layer implemented in "adaptor" (aka *network interface card* NIC)
  - ❍ Ethernet card, PCMCI card, 802.11 card
  - ❍ implements link, physical layer
- ❒ attaches into host's system buses
- ❒ combination of hardware, software, firmware



Adaptors Communicating

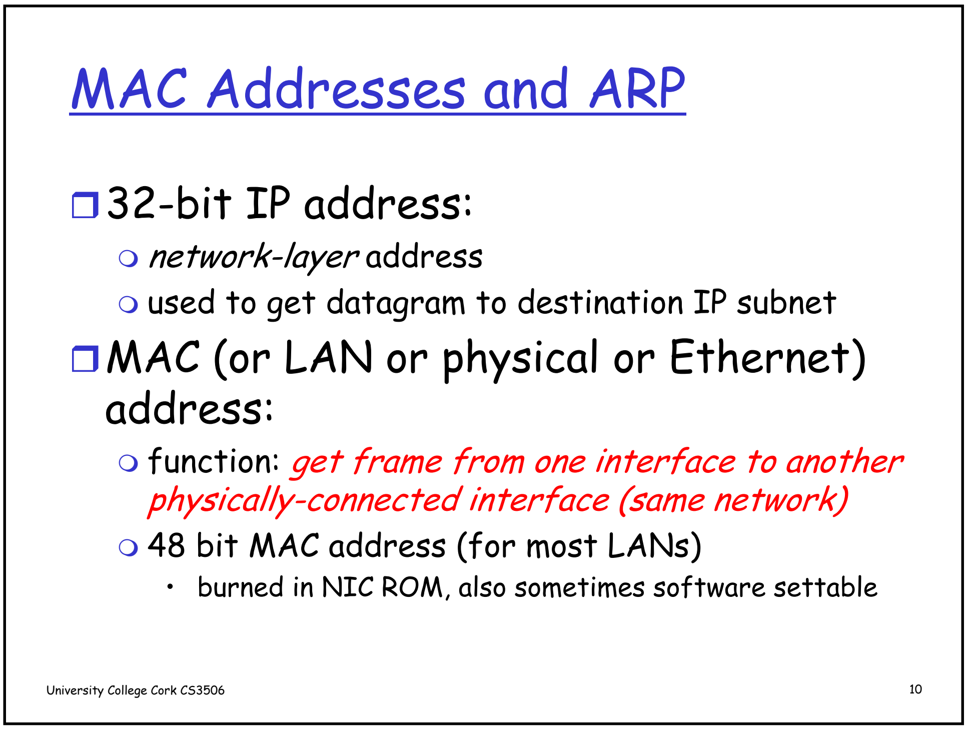


- ❒ sending side:
  - ❍ encapsulates datagram in frame
  - ❍ adds error checking bits, reliability, flow control, etc.
- ❒ receiving side
  - ❍ looks for errors, reliability, flow control, etc
  - ❍ extracts datagram, passes to upper layer at receiving side



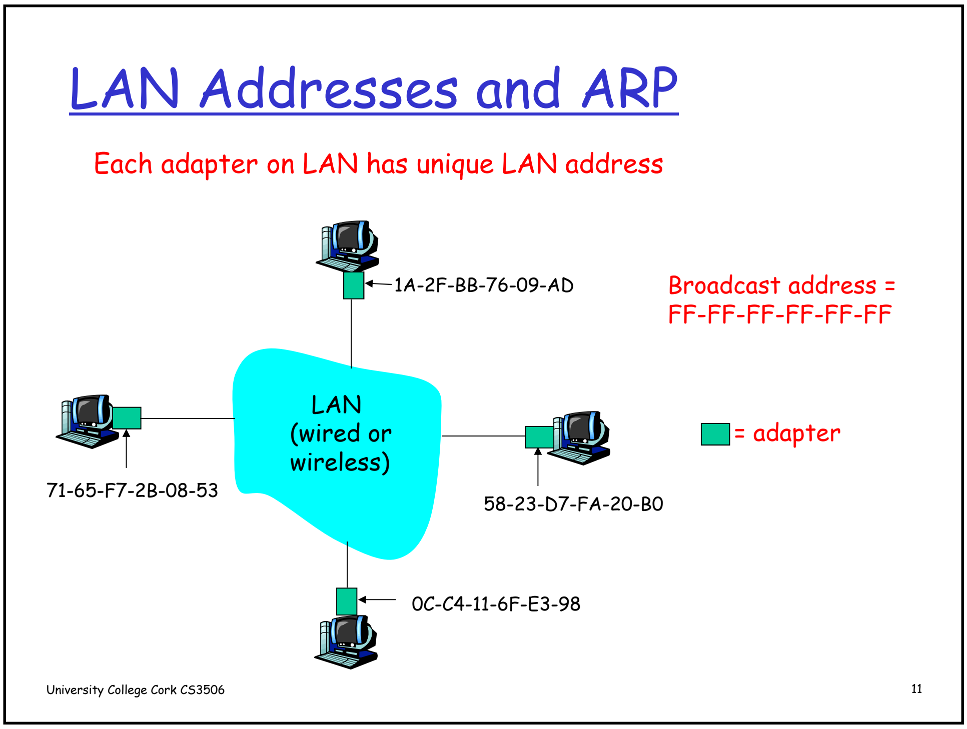
Link Layer

- ❐ Introduction and services
- ❐ Link-Layer Addressing
- ❐ Error Detection
- ❐ Multiple Access
- ❐ Ethernet
- ❐ Link-layer switches
- ❐ Link virtualization: MPLS
- ❐ Data Centre Networks
- ❐ 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

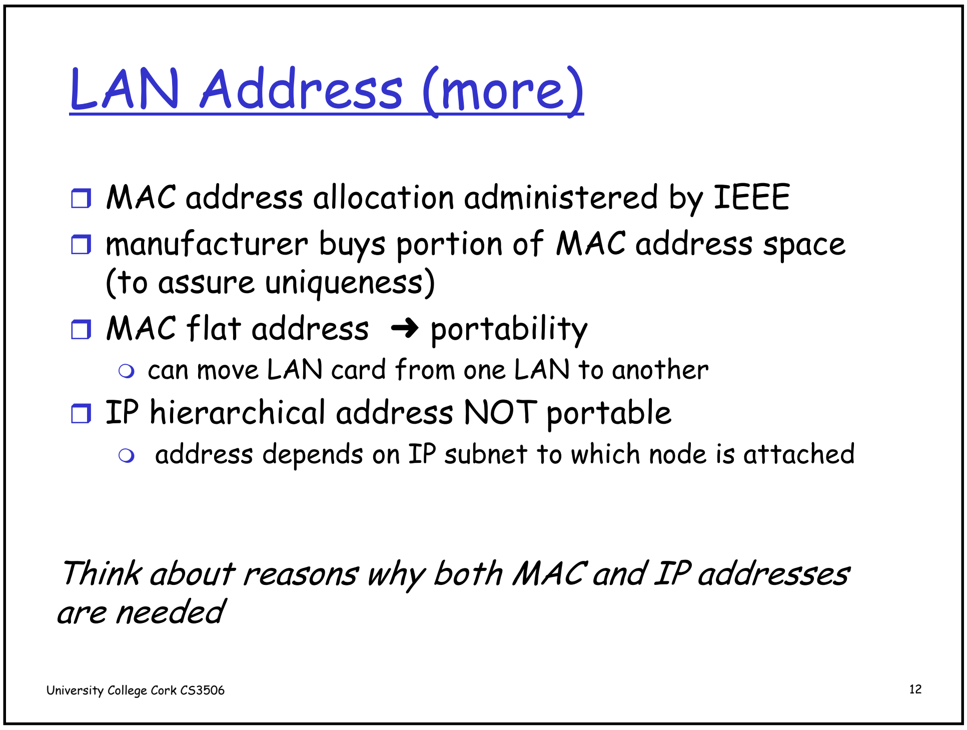


LAN Addresses and ARP

Each adapter on LAN has unique LAN address

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

🟩 = adapter



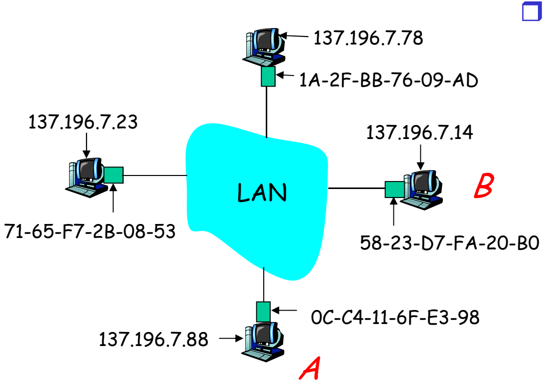
LAN Address (more)

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

Think about reasons why both MAC and IP addresses are needed

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)

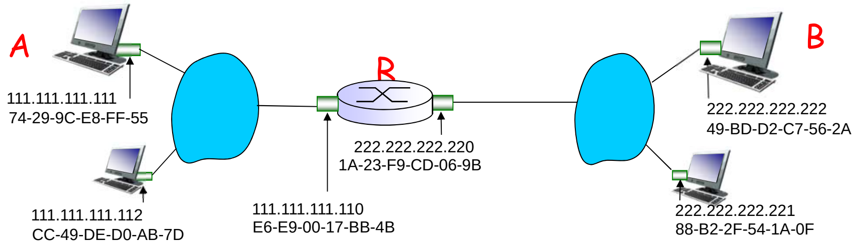
ARP protocol: Same LAN (network)

- ❒ 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 IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows R's MAC address (how?)



A

111.111.111.111
74-29-9C-E8-FF-55

111.111.111.112
CC-49-DE-D0-AB-7D

R

111.111.111.110
E6-E9-00-17-BB-4B

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

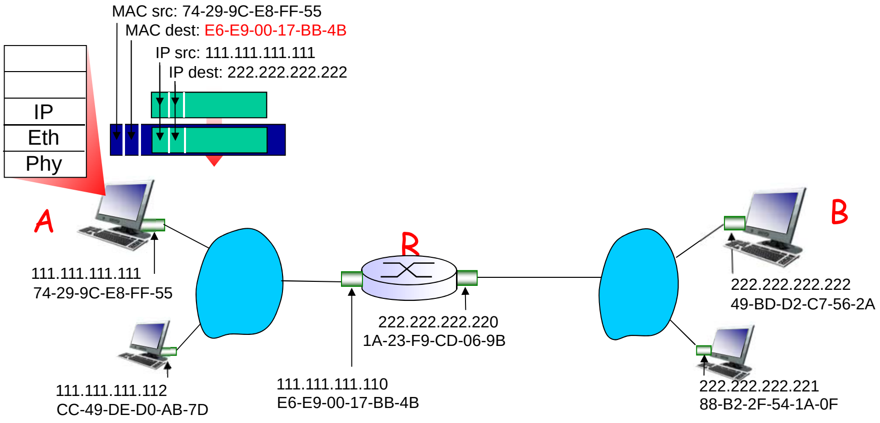
B

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

- 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



MAC src: 74-29-9C-E8-FF-55
MAC dest: E6-E9-00-17-BB-4B
IP src: 111.111.111.111
IP dest: 222.222.222.222

IP
Eth
Phy

A

111.111.111.111
74-29-9C-E8-FF-55

111.111.111.112
CC-49-DE-D0-AB-7D

R

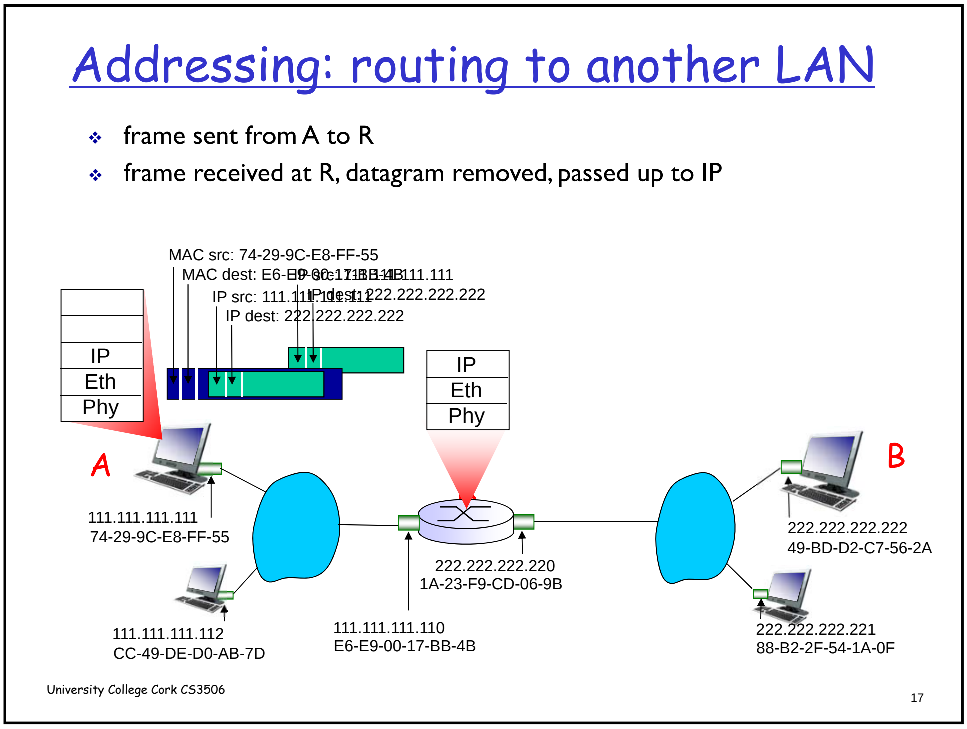
111.111.111.110
E6-E9-00-17-BB-4B

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

B

222.222.222.222
49-BD-D2-C7-56-2A

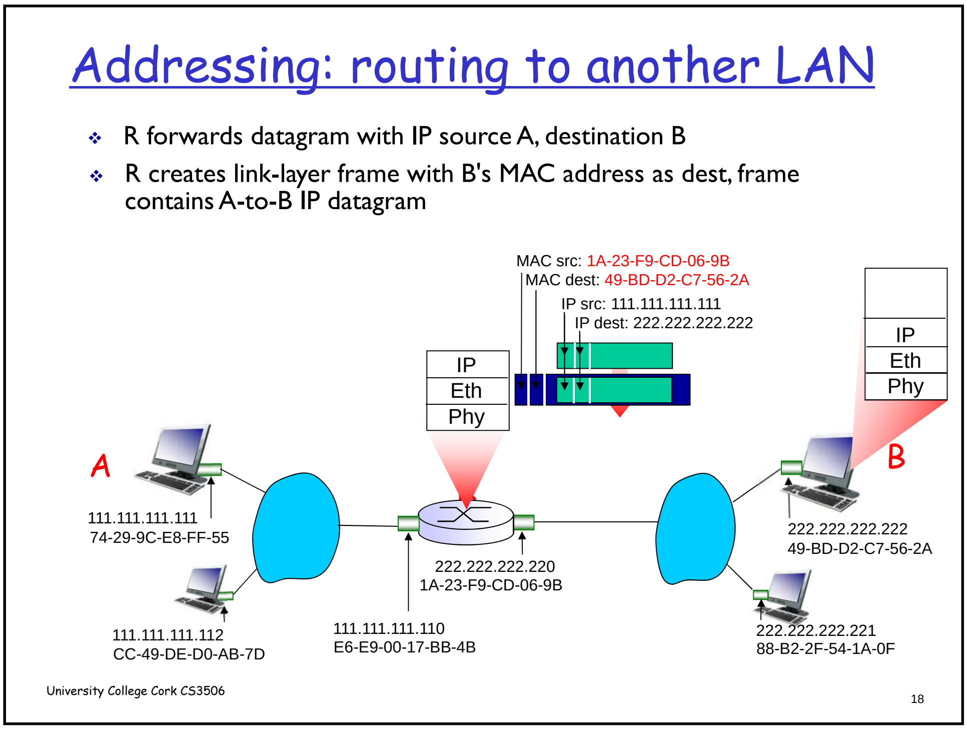
222.222.222.221
88-B2-2F-54-1A-0F



Addressing: routing to another LAN

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

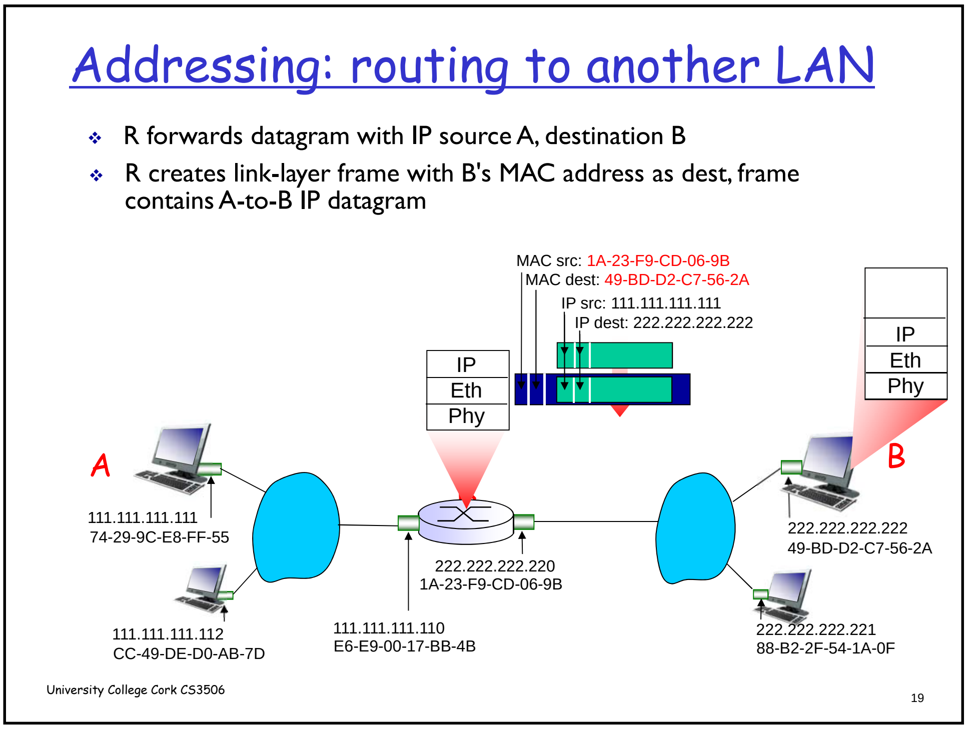
University College Cork CS3506



Addressing: routing to another LAN

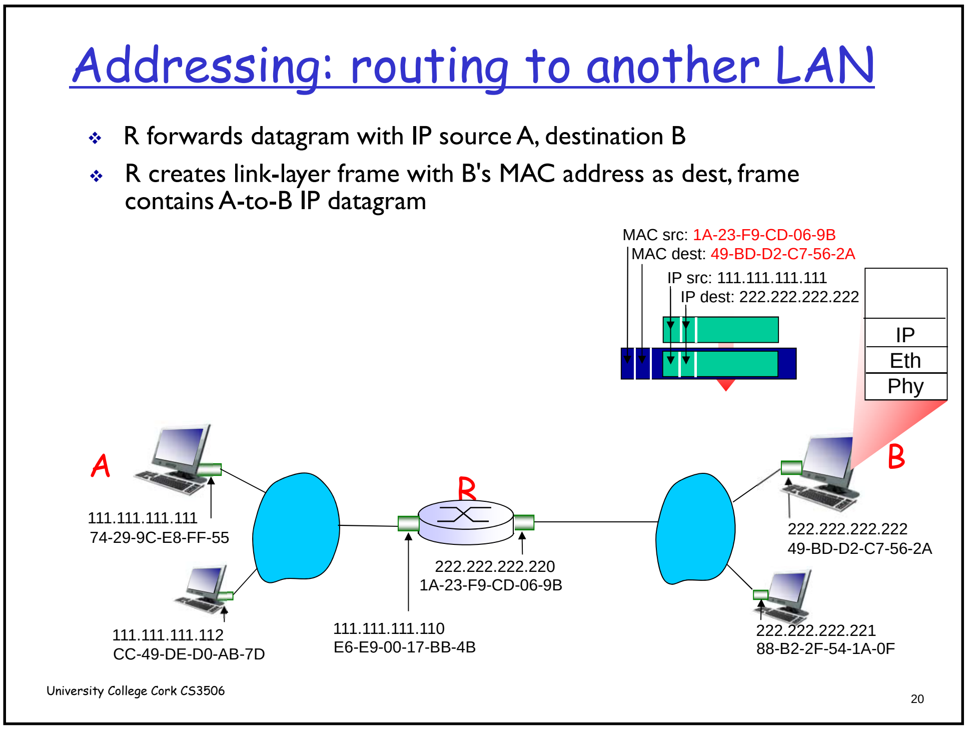
- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram

University College Cork CS3506



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

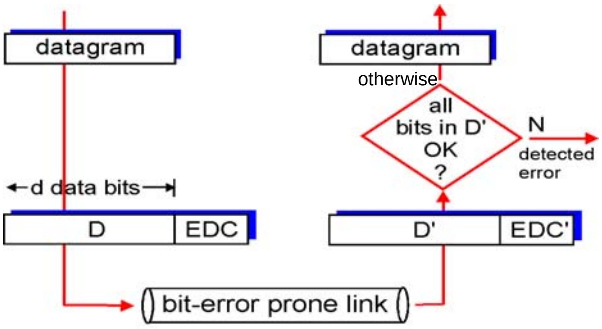
Link Layer

- ❒ Introduction and services
- ❒ Link-Layer Addressing
- ❒ Error Detection
- ❒ Multiple Access
- ❒ Ethernet
- ❒ Link-layer switches
- ❒ Link virtualization: MPLS
- ❒ Data Centre Networks
- ❒ A day in the life of a web request

Error Detection

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

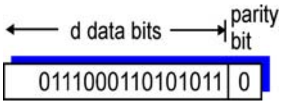
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity Checking

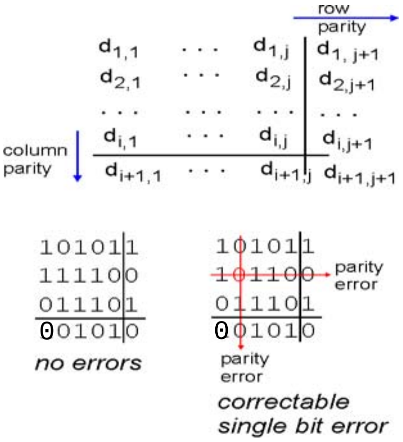
Single Bit Parity:

Detect single bit errors



Two Dimensional Bit Parity:

Detect *and correct* single bit errors



row parity
$d_{1,1}$ . . . $d_{1,j}$ $d_{1,j+1}$
$d_{2,1}$ . . . $d_{2,j}$ $d_{2,j+1}$
. . . .
$d_{i,1}$ . . . $d_{i,j}$ $d_{i,j+1}$
column parity
$d_{i+1,1}$ . . . $d_{i+1,j}$ $d_{i+1,j+1}$
101011
111100
011101
001010
no errors
101011
101100
011101
001010
parity error
parity error
correctable single bit error

Internet checksum (review)

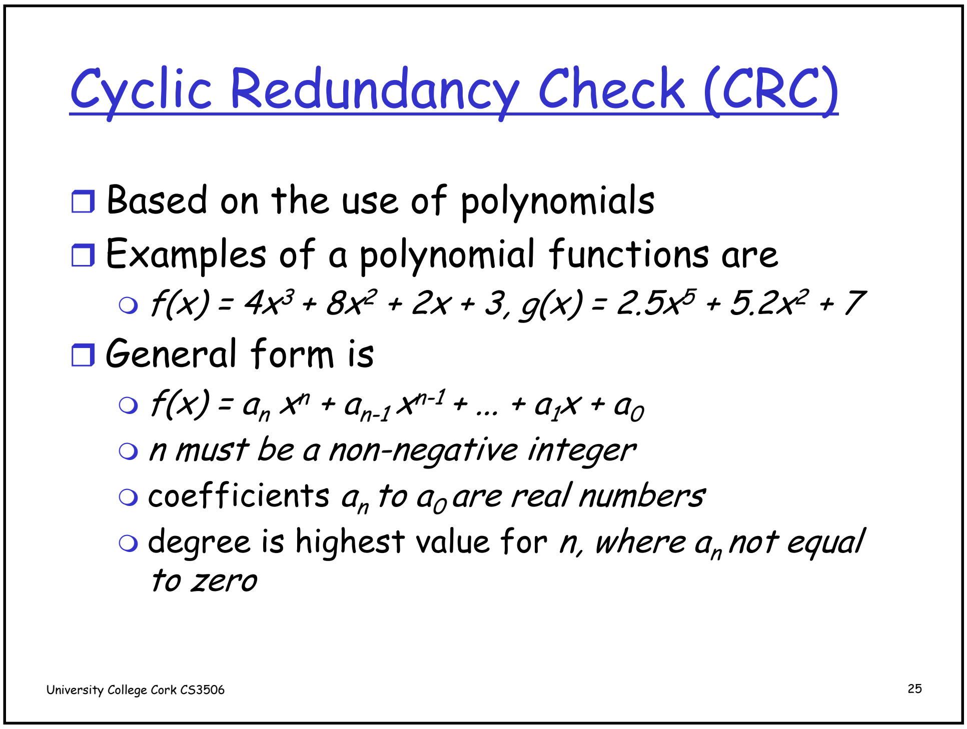
Goal: detect "errors" (e.g., flipped bits) in transmitted packet (as used at transport layer)

Sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

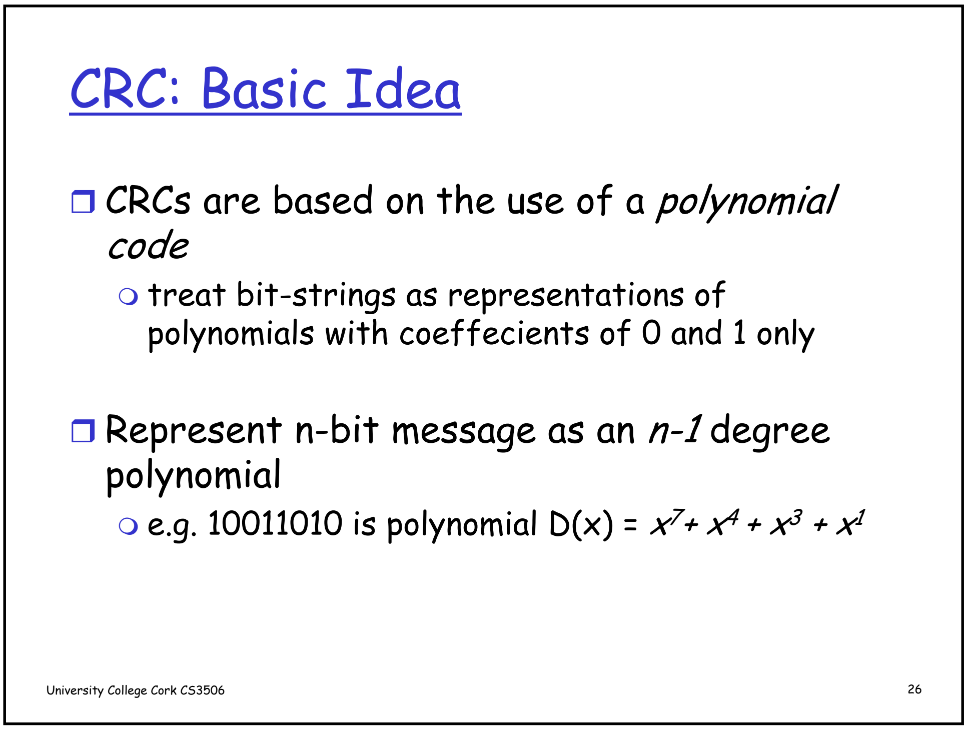
Receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?*



Cyclic Redundancy Check (CRC)

- Based on the use of polynomials
- Examples of a polynomial functions are
  - $f(x) = 4x^3 + 8x^2 + 2x + 3$, $g(x) = 2.5x^5 + 5.2x^2 + 7$
- General form is
  - $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$
  - *n must be a non-negative integer*
  - coefficients a_n *to* a_0 *are real numbers*
  - degree is highest value for *n, where* a_n *not equal to zero*



CRC: Basic Idea

- CRCs are based on the use of a *polynomial code*
  - treat bit-strings as representations of polynomials with coeffecients of 0 and 1 only
- Represent n-bit message as an *n-1* degree polynomial
  - e.g. 10011010 is polynomial D(x) = $x^7 + x^4 + x^3 + x^1$

CRCs: Basic Idea

- Sender and receiver agree (ahead of time) on a generator polynomial G(x)
  - convert D(x) to P(x), with P(x) exactly divisible by G(x)
  - receiver knows that message should be exactly divisible by G(x) – it divides the received message bits by G(x) to detect errors
- Let k be the degree of some divisor polynomial G(x)
  - e.g., *G(x) = x3+ x2 + 1, degree=3*

CRC Mechanism

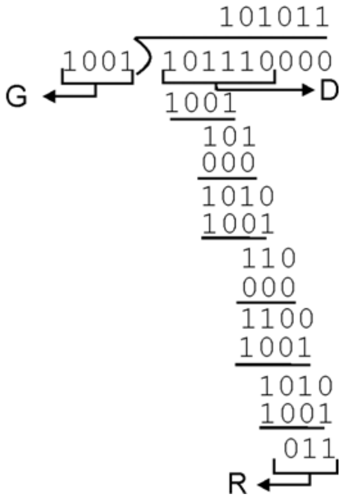
- Transmit polynomial P(x) that is evenly divisible by G(x), and receive polynomial P(x) + E(x)
  - E(x) represents corrupted bits and E(x)=0 implies no errors
- Recipient divides (P(x) + E(x)) by G(x)
  - the remainder will be zero in only two cases
    - E*(x) was zero (i.e. there was no error)*
    - or *E(x) is exactly divisible by C(x)*
  - Choose *G(x) to make second case extremely rare*

CRC Mechanism

- ❒ Use polynomial arithmetic
  - ❍ addition = exclusive OR (XOR)
    - e.g. 10011011 + 11001010 = 01010001
  - ❍ subtraction = exclusive OR (XOR)
  - ❍ "goes into" = has the same number of bits
- ❒ Calculate *P(x)*
  - ❍ shift D(x) left by k bits i.e. the degree of G(x), then
  - ❍ divide P(x) by G(x) to produce a remainder term R(x)

CRC Example

- ❒ D(x) = 101110
- ❒ G(x) = 1001 (degree = 3)
- ❒ Shift D(x) by 3 to produce message P(x)
- ❒ Polynomial long division by G(x) to yield remainder R(x) = 011
- ❒ So, the message minus R(x) would be exactly divisible by C(x)



CRC Mechanism

- Thus,
  - *P(x) = D(x) shifted left k bits, minus R(x)*
  - In example: *P(x) = 101110 - 011 = 101110011*
- Sender sends P(x) as the frame
- Receiver reads frame *P(x) + E(x)*
  - divides by *G(x); zero remainder => no errors*
  - yields *D(x) by shifting right k bits*
- Widely used, e.g. Ethernet CRC is $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$

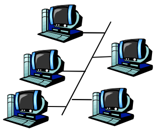
Link Layer

- Introduction and services
- Link-Layer Addressing
- Error Detection
- Multiple Access
- Ethernet
- Link-layer switches
- Link virtualization: MPLS
- Data Centre Networks
- A day in the life of a web request

Multiple Access Links and Protocols

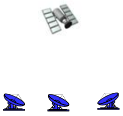
Two types of "links":

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - old-fashioned Ethernet
  - upstream HFC (Hybrid Fibre Coax)
  - 802.11 wireless LAN (WiFi)



shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Multiple Access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - 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 node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

Ideal Multiple Access Protocol

Broadcast channel of rate R b/s

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

MAC Protocols: a taxonomy

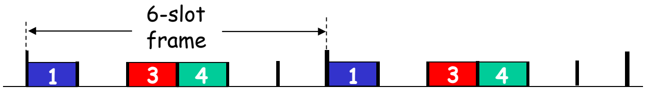
Three broad classes:

- ❒ Channel Partitioning
  - divide channel into smaller "pieces" (typically time slots)
  - allocate piece to 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

Channel Partitioning MAC protocol: TDMA

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 pkt, slots 2,5,6 idle

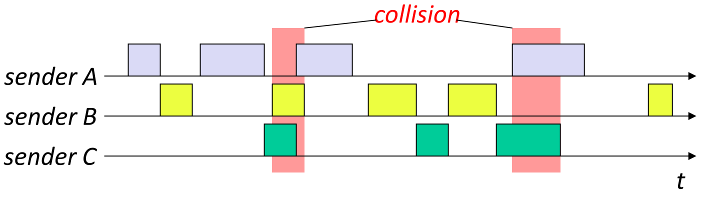


Random Access Protocols

- ❒ When 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:
  - ❍ ALOHA
  - ❍ Carrier Sense Medium Access (CSMA)

ALOHA

- ❒ Very simple; no coordination; no synchronisation
- ❒ when frame first arrives at sender's MAC layer
  - ❍ transmit immediately
  - ❍ This can cause collisions if other stations send their frame in time that overlaps
- ❒ ALOHA suffers from very poor link utilisation



CSMA (Carrier Sense Multiple Access)

CSMA: listen before transmit:

If channel sensed idle: transmit entire frame

- ❒ If channel sensed busy, defer transmission

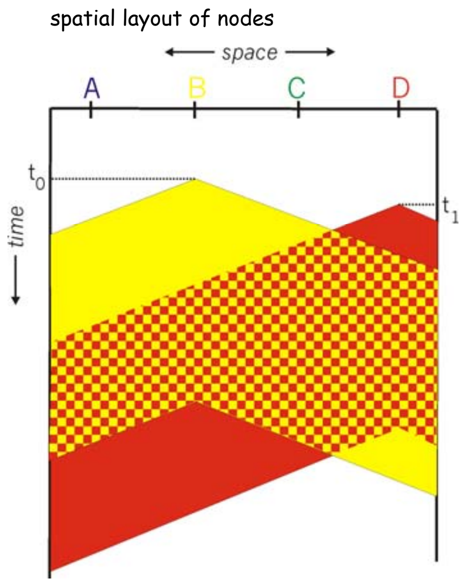
- ❒ human analogy: don't interrupt others!

CSMA 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



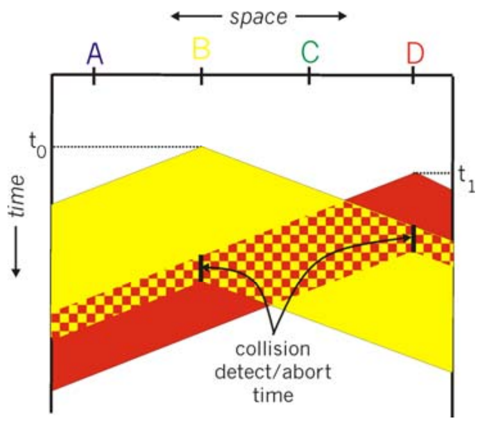
CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- 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



CSMA/CD in Classic Ethernet

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** - waits for a specific period of time and then returns to Step 2

Backoff helps avoid repeated collisions

Ethernet's Exponential Backoff

- *Goal*: adapt retransmission attempts to estimated current load
  - Attempt to "spread" retransmission attempts
  - heavy load: random wait will be longer
- Algorithm
  - NIC waits (K x slot duration) times; K chosen at random from given range
  - after 1st collision: choose K from {0,1}
  - after 2nd collision: choose K from {0,1,2,3}…
  - after *m*th collision, NIC chooses *K* from {0,1,2,…, 2^m-1}

CSMA/CD efficiency

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

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

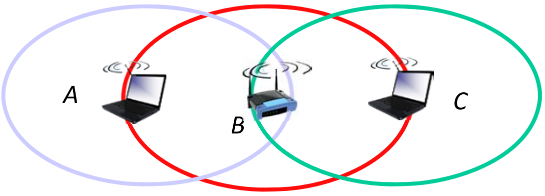
- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!

CSMA in Wireless Links

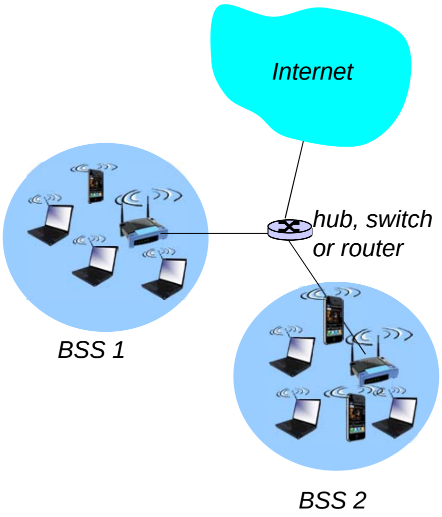
- ❒ Problems in wireless networks for CDMA/CD
  - ❍ signal strength decreases proportional to (the square) of the distance between sender and receiver
  - ❍ the sender could apply Carrier Sense and CD, but the collisions happen *at the receiver*
  - ❍ it might likely be the case that a sender cannot “hear” other transmissions, i.e., Carrier Sense does not work
  - ❍ it might likely be the case that a sender cannot “hear” the collision, i.e., CD does not work

Hidden Node Problem

- ❒ A sends to B; C cannot receive A
- ❒ C wants to send to B, C senses a “free” medium (Carrier Sense fails)
- ❒ collision at B, A cannot detect the collision (CD fails)
- ❒ A is “hidden” for C



IEEE 802.11 (WiFi)



- ❒ Wireless Local Area Network (LAN)
- ❒ Devices in a cell communicate with Access Point (aka base station)
  - ❍ or ad-hoc network, device-to-device
  - ❍ formally a Basic Service Set, with an SSID
  - ❍ BSSs can share an SSID, eg eduroam

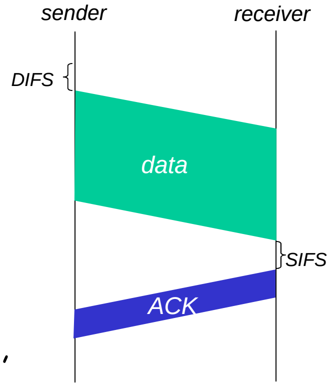
Wireless Communication

- ❒ *important* differences from wired link ….
  - *decreased signal strength:* radio signal signal strength decreases proportional to (the square) of distance between sender and receiver, and also due to absorption
  - *interference from other sources:* standard WiFi operates in same radio frequencies as Bluetooth and other devices
  - *multipath propagation:* radio signal reflects off objects and ground, arriving at destination at slightly different times

…. make communication across (even a point to point) wireless link much more "difficult"

802.11 Basics

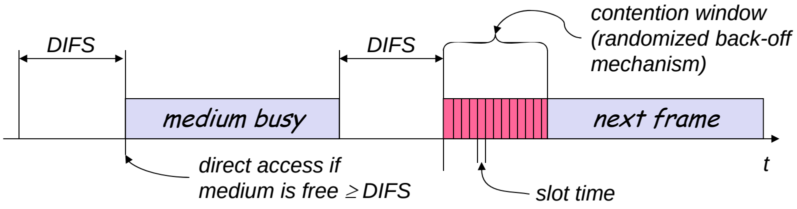
- Inter Frame Spaces are delays prior to transmitting a frame
  - Different IFSs allow control over the importance of a given frame
  - DIFS is for regular data frames, SIFS is a shorter period for ACKs etc



Why is ACK needed?

802.11 CSMA Basics

- 802.11 uses CSMA/CA (Collision Avoidance)
  - Basic mechanism uses *Clear Channel Assessment* to listen to the medium prior to transmission
  - Plus a random exponential backoff



802.11 CSMA/CA Procedure

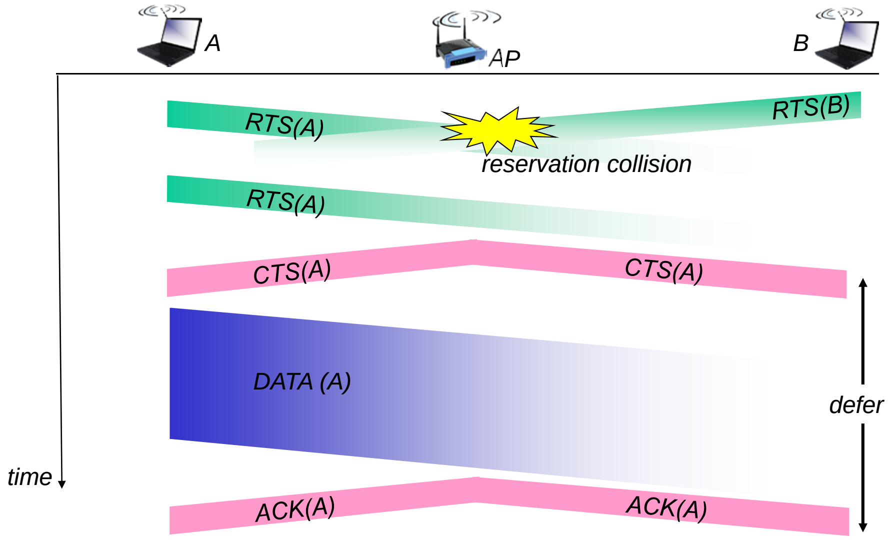
- station ready to send starts sensing the medium using CCA
- if the medium is free for the duration of an Inter-Frame Space (IFS), the station can start sending (IFS depends on service type)
- if the medium is busy, the station has to wait for a free IFS, then the station must additionally wait a random back-off time (collision avoidance, multiple of slot-time)
- if another station occupies the medium during the back-off time of the station, the back-off timer stops (helps enforce fairness)

CSMA/CA based on Reservation

idea: allow sender to "reserve" channel rather than random access of data frames: avoid collisions for long data frames

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

Collision Avoidance: RTS/CTS



"Taking Turns" MAC protocols

channel partitioning MAC protocols:

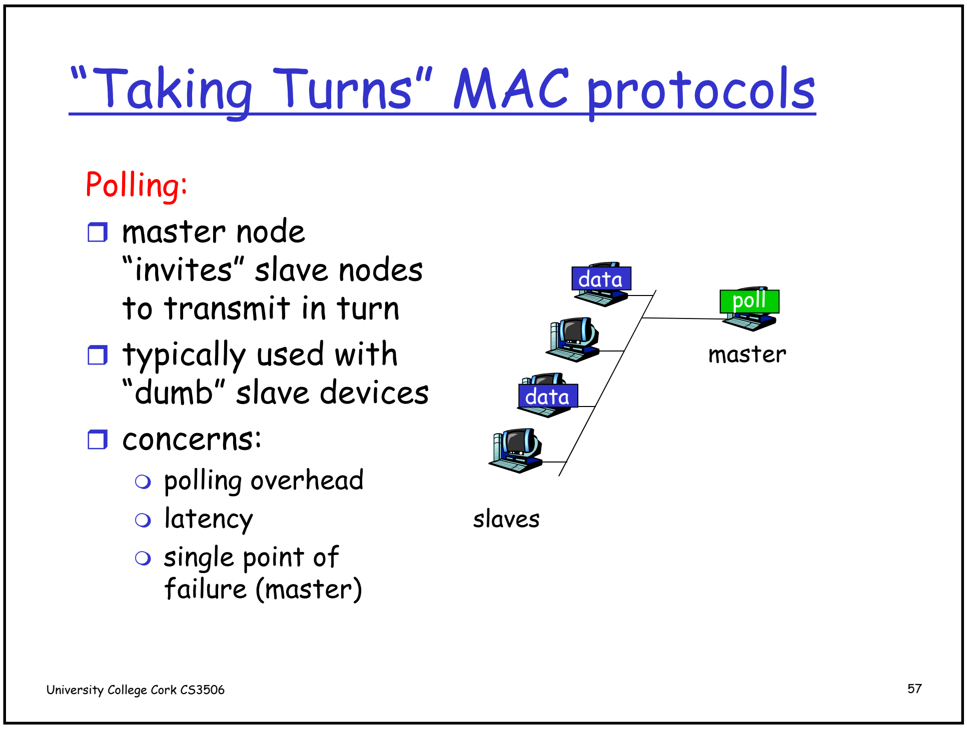
- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

Random access MAC protocols

- efficient at low load: single node can fully utilise channel
- high load: collision overhead

"taking turns" protocols

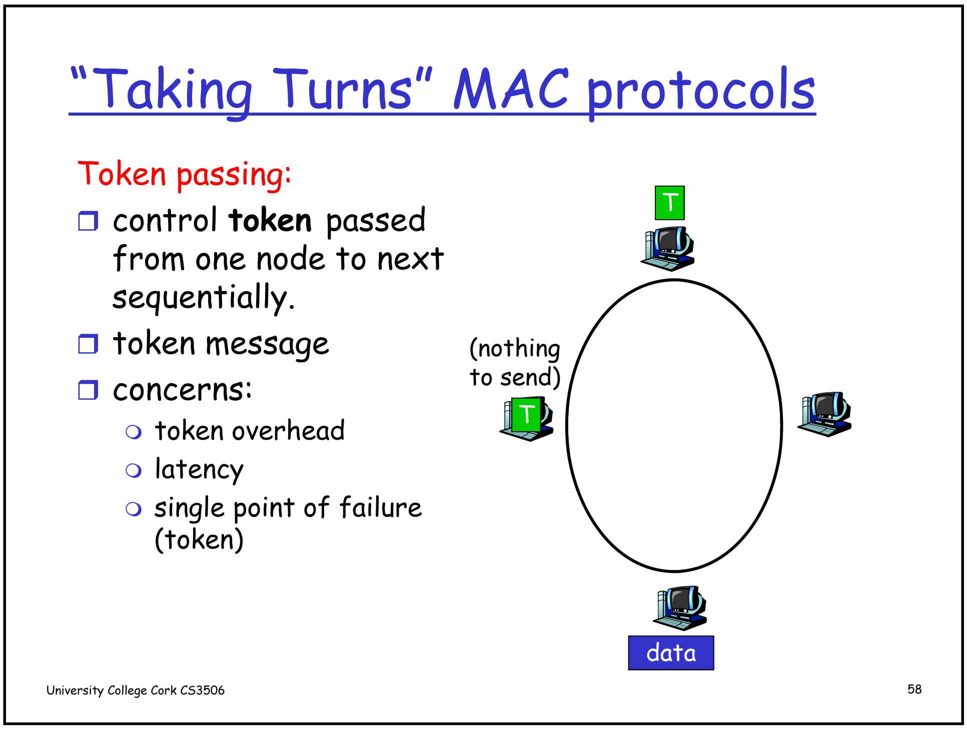
look for best of both worlds!



"Taking Turns" MAC protocols

Polling:

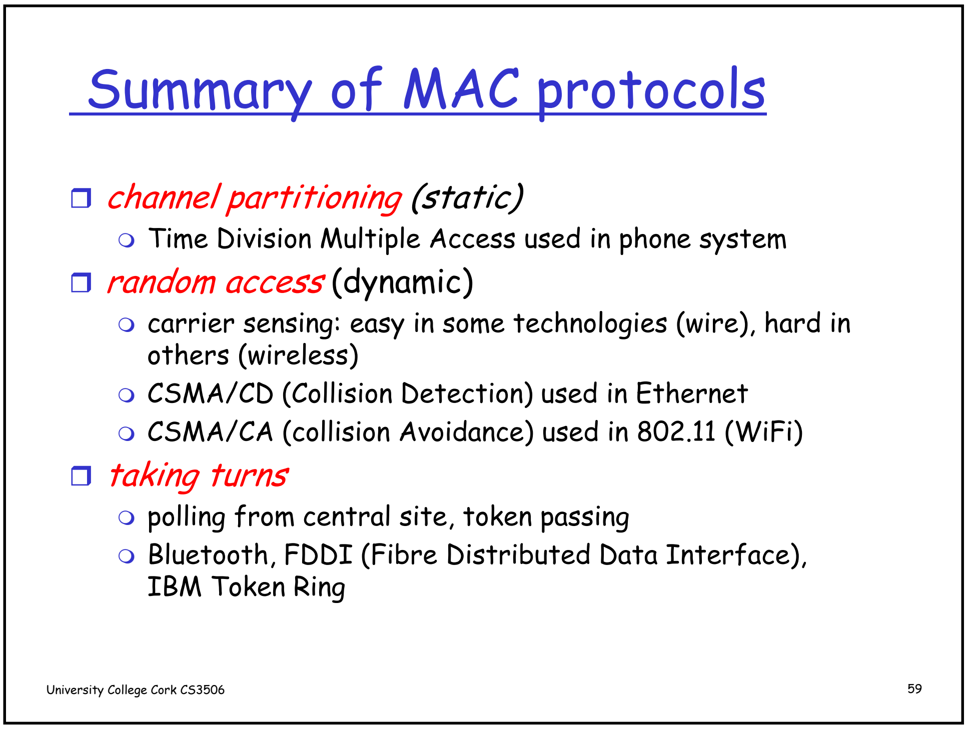
- ❒ master node "invites" slave nodes to transmit in turn
- ❒ typically used with "dumb" slave devices
- ❒ concerns:
  - polling overhead
  - latency
  - single point of failure (master)



"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 (static)*
  - Time Division Multiple Access used in phone system
- *random access* (dynamic)
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD (Collision Detection) used in Ethernet
  - CSMA/CA (collision Avoidance) used in 802.11 (WiFi)
- *taking turns*
  - polling from central site, token passing
  - Bluetooth, FDDI (Fibre Distributed Data Interface), IBM Token Ring