Link Layer

Kyunghan Lee
Networked Computing Lab (NXC Lab)
Department of Electrical and Computer Engineering
Seoul National University
https://nxc.snu.ac.kr
kyunghanlee@snu.ac.kr

Goals

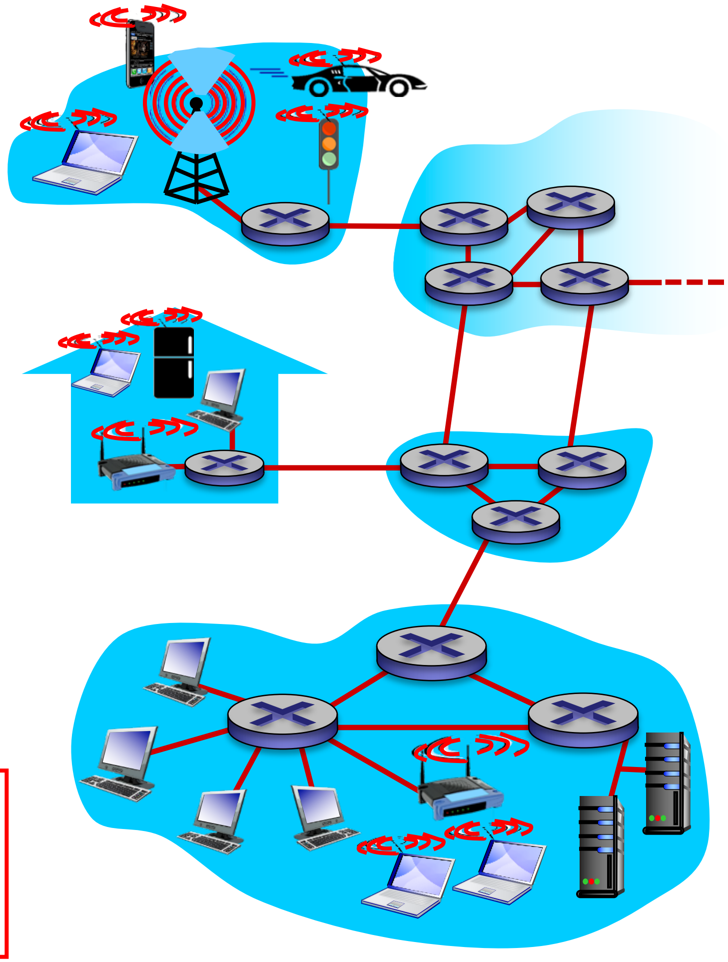
- understand principles behind link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - local area networks: Ethernet, VLANs
- instantiation, implementation of various link layer technologies

Link layer: Introduction

Terminology:

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

data-link layer has responsibility of transferring datagram from one node to *physically 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 rdt over link

transportation analogy:

- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = 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 shared medium
  - "MAC" addresses used in frame headers to identify source, destination
    - different from IP address!

- *reliable delivery between adjacent nodes*
  - we learned how to do this already (chapter 3, RDT, ARQ)!
  - 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)

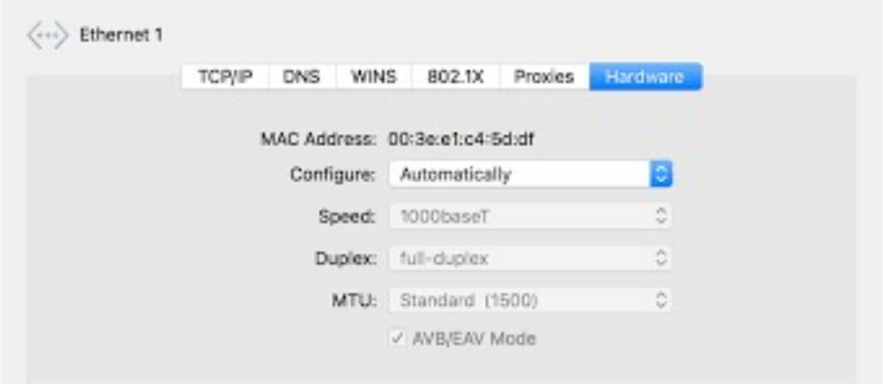
- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit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

Data link layer address (MAC address)

- Each host and router on a subnet needs a data link layer address to specify its address on the subnet
  - This address appears in the data link layer frame sent on a subnet
  - For instance, 48-bit 802.3 MAC layer frame addresses for LANs

00:01:AB:01:02:03

Manufacturer identifier　　Random digits allocated by manufacturer



IP and MAC addr serve different purposes

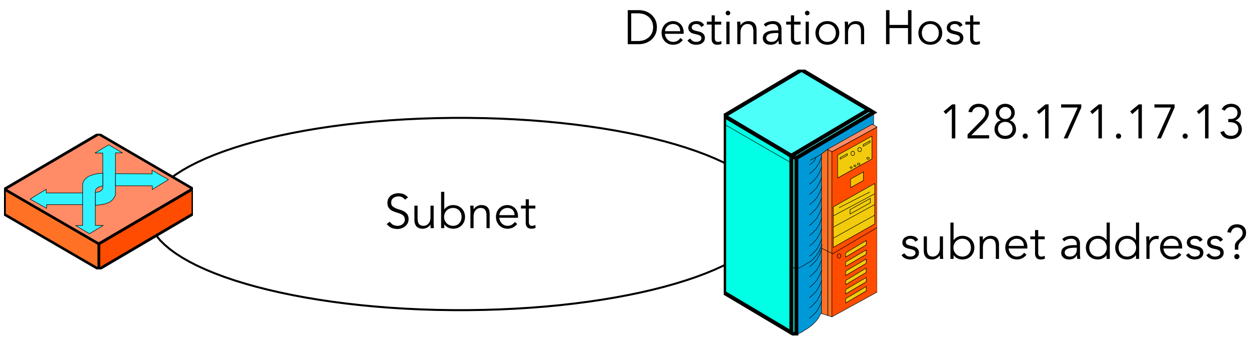
- IP address
  - To guide delivery to destination host across the Internet (across multiple networks)
- MAC Address (subnet address)
  - To guide delivery between two hosts, two routers, and a host and router within a single subnet
  - Same LAN, Frame Relay network, etc.

Analogy

- In company, each person has a company-wide ID number (like IP address)
- In company, person also has a local office number in a building
- Paychecks are made out to ID numbers
- For delivery, also need to know office number

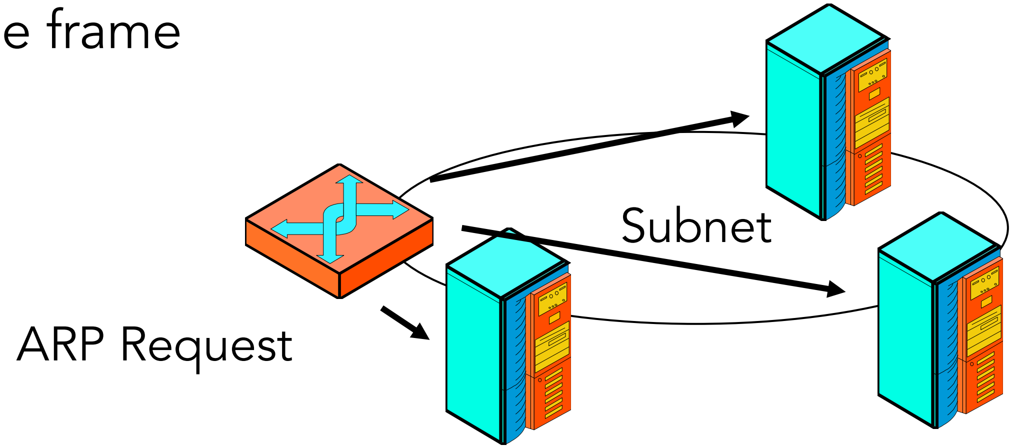
Address resolution

- Problem
  - Router knows that destination host is on its subnet based on the IP address of an arriving packet
  - Does not know the destination host's subnet address, *so cannot deliver the packet across the subnet*



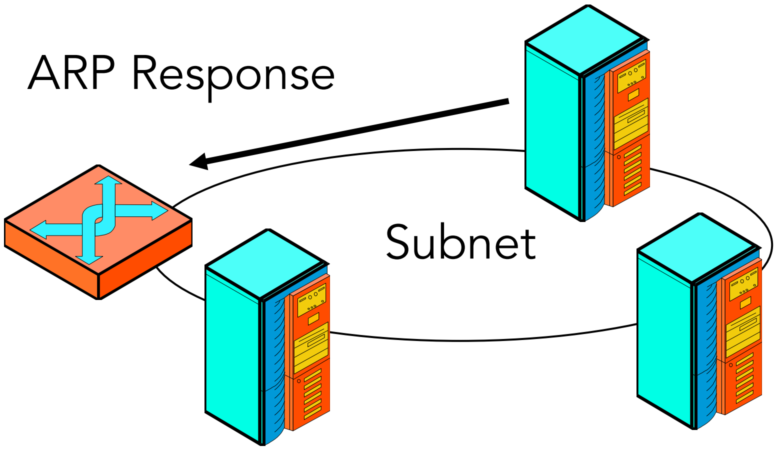
Address Resolution Protocol (ARP)

- Router creates an ARP request message to be sent to all hosts on the subnet.
  - ARP message asks "Who has IP address 128.171.17.13?"
  - Passes ARP request to data link layer process for delivery
- Data link process of router broadcasts the ARP Request message to all hosts on the subnet.
  - On a LAN, MAC address of 48 ones tells all stations to pay attention to the frame



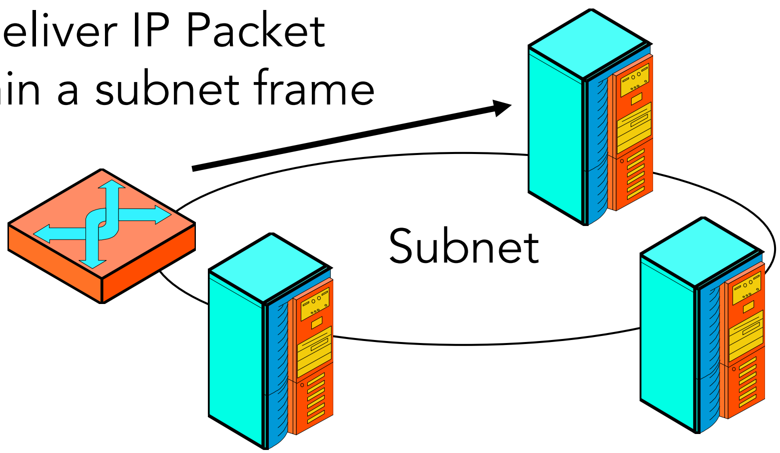
Address Resolution Protocol (ARP)

- Host with IP address 128.171.17.13 responds
  - Internet process creates an ARP response message
  - Contains the destination host's subnet address (48-bit MAC address on a LAN)



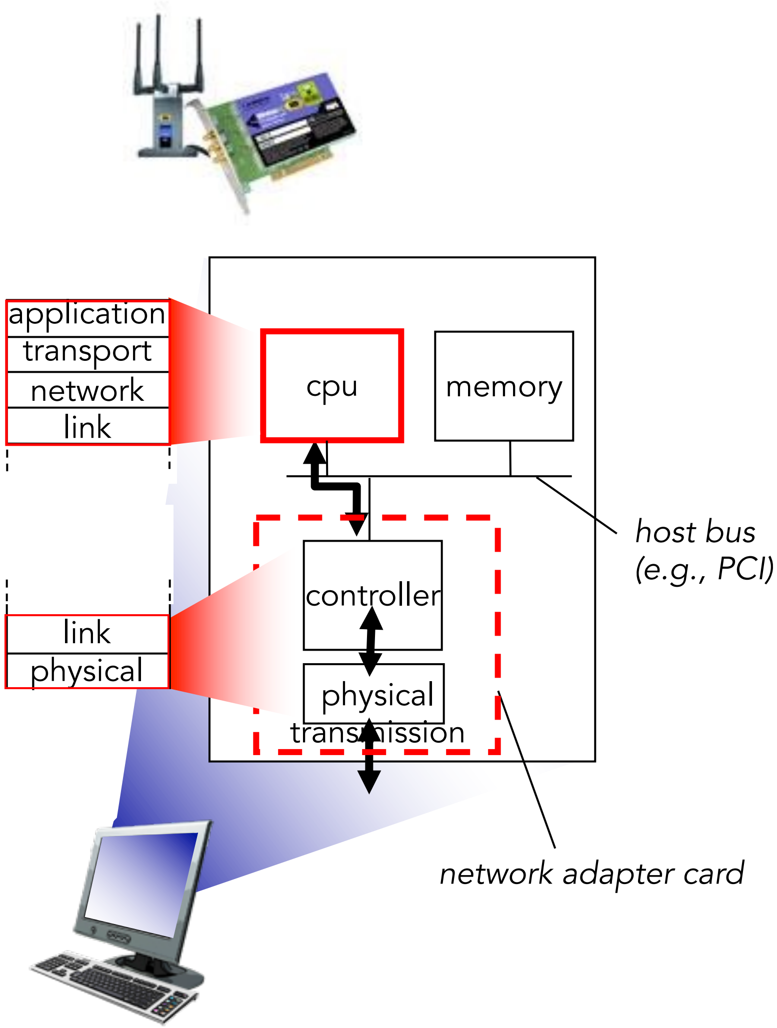
Address Resolution Protocol (ARP)

- Router delivers the IP packet to the destination host
  - Places the IP packet in the subnet frame
  - Puts the *destination host's subnet address* in the destination address field of the frame

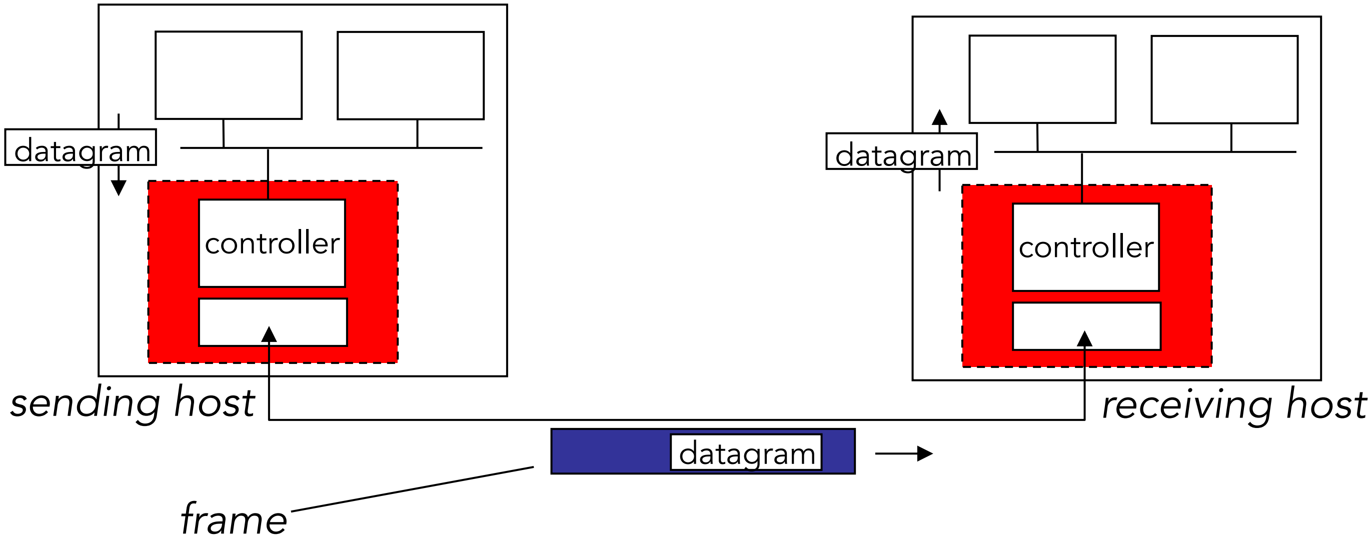


Where is the link layer implemented?

- In each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - 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, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc.
  - extracts datagram, passes to upper layer at receiving side

Link layer header

- Header format depends on its own standard

Ethernet (802.3) Frame Format

7 bytes	1 byte	6 bytes	6 bytes	2 bytes	42 to 1500 bytes	4 bytes	12 bytes
Preamble	Start of Frame Delimiter	**Destination MAC Address**	**Source MAC Address**	Type	Data (payload)	CRC	Inter-frame gap

Data payload is IP packets carrying TCP or UDP data

WiFi (802.11) Frame Format

2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0 to 2312 bytes	4 bytes
Frame Control	Duration	MAC Address 1 (Destination)	MAC Address 2 (Source)	MAC Address 3 (Router)	Seq Control	MAC Address 4 (AP)	Data (payload)	CRC

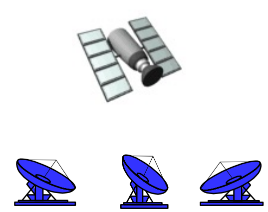
Multiple access links, protocols

Two types of "links":

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch, host
- *broadcast (shared wire or medium)*
  - old-fashioned Ethernet
  - 802.11 wireless LAN

shared wire
(e.g., 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

An Ideal Multiple Access Protocols

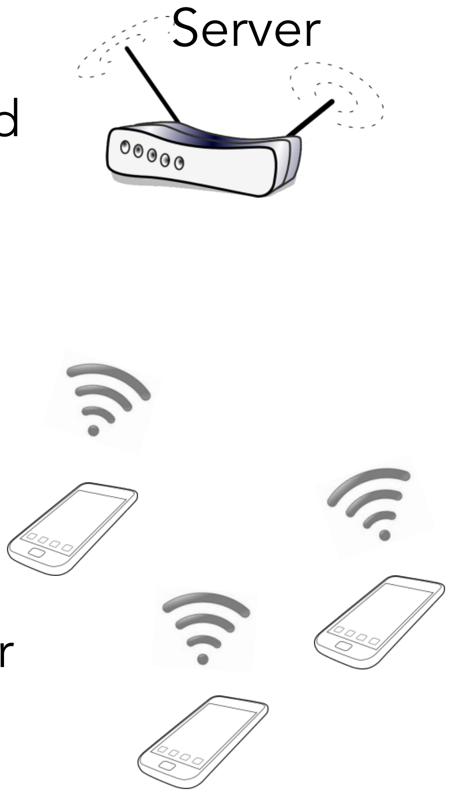
given: broadcast channel of rate R bps

desiderata (to be):

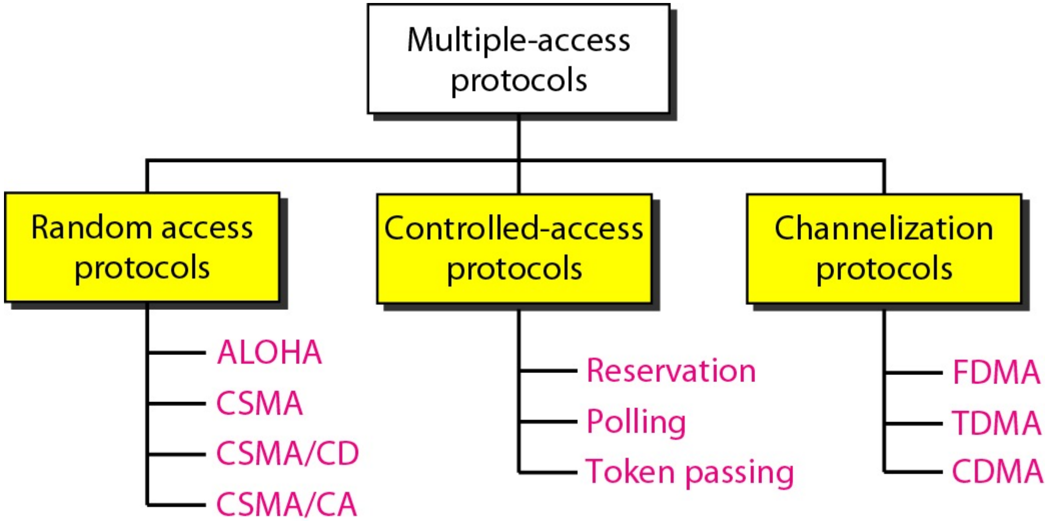
1. when one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple

Multiple Access Communications

- Multi-access system
  - Received signal is the sum of the attenuated transmitted signal from a set of other nodes, corrupted by delay and noise
  - Conceptually: View it as each node having a queue of packets to be transmitted, and the multi-access channel is a common server
  - Ideally: Server should view waiting packets as one combined queue to be served by some queueing discipline (e.g. FIFO)
  - Reality: Server does not know which nodes contain packets, also each node is unaware of packets at other nodes
- Question:
  - How this knowledge about state of system can be distributed?



Multiple Access Protocols



- Random Access
  - "Free for all" approach:
    - Nodes send new packets immediately, hoping for no interference from other nodes
  - Question:
    - When and how packets are retransmitted when collisions occur?

MAC protocols: taxonomy

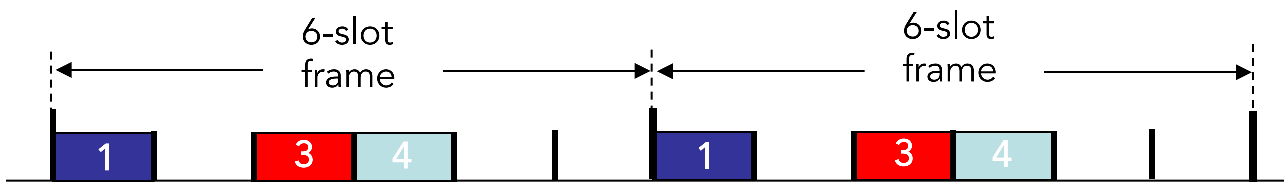
Three broad classes:

- *channel partitioning*
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - 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 protocols: TDMA

TDMA: time division multiple access

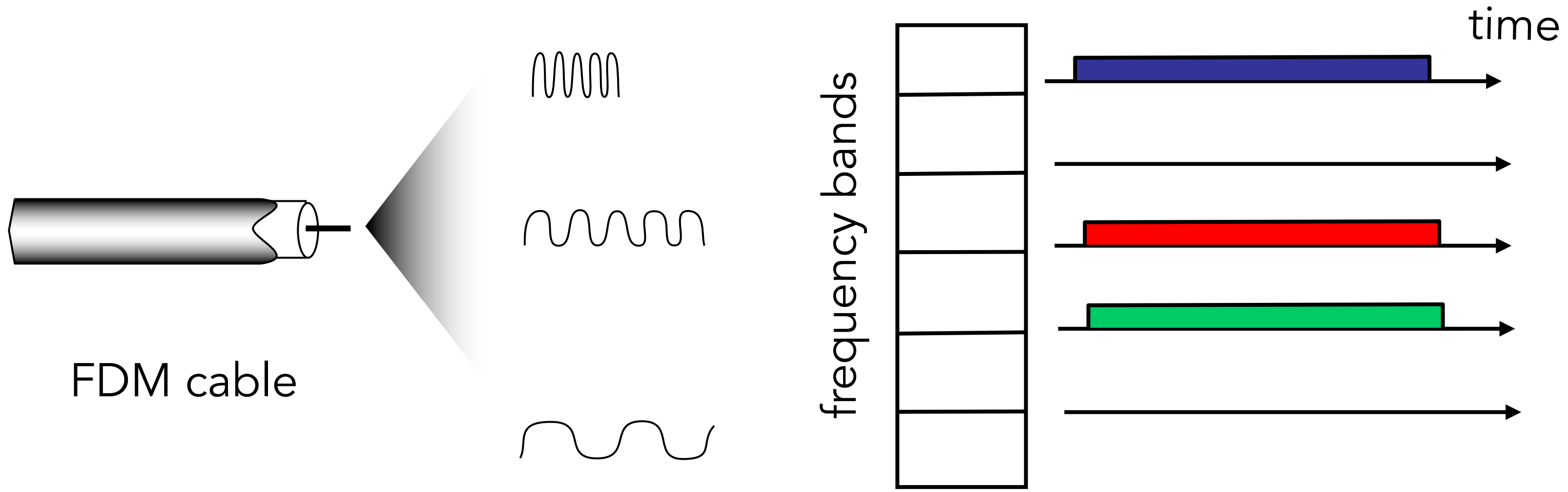
- access to channel in "rounds"
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle



Channel partitioning MAC protocols: FDMA

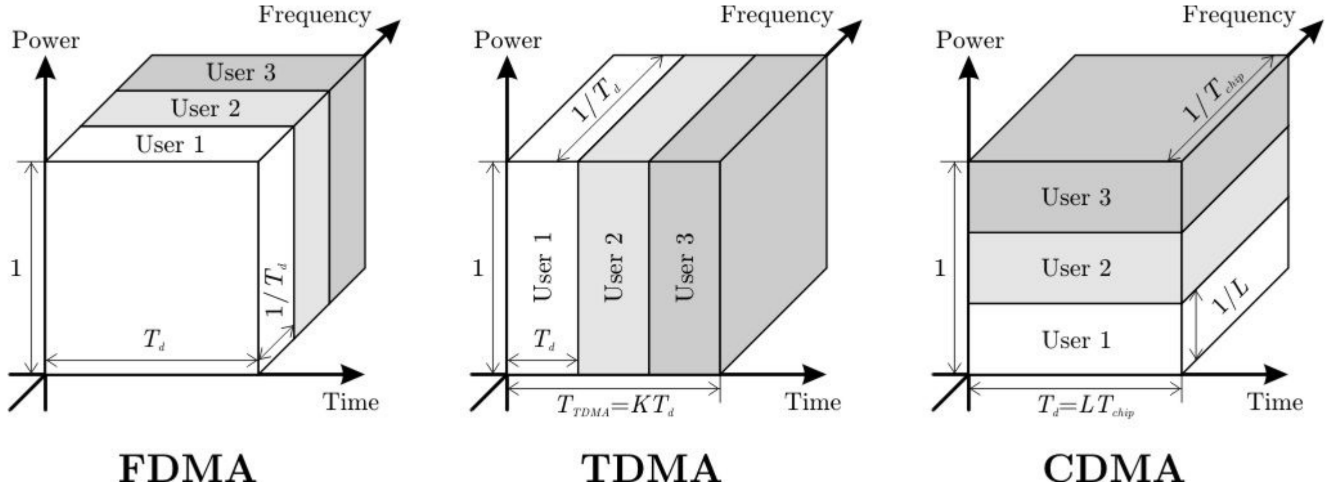
FDMA: frequency division multiple access

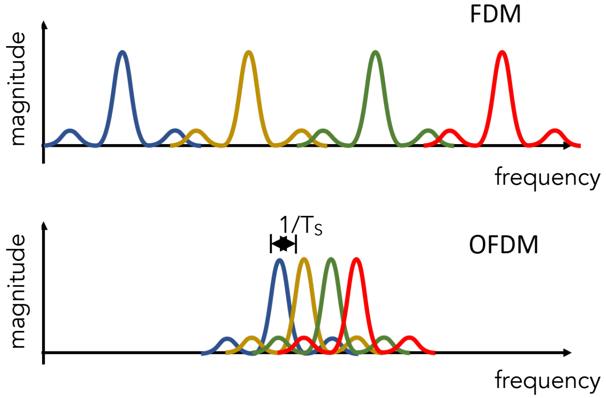
- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle

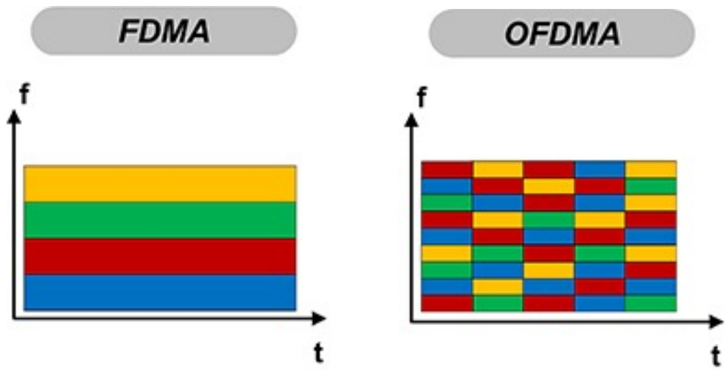


FDMA / TDMA / CDMA

- Resources can be divided in various dimensions (frequency, time, and code) in a disjoint manner







Random Access Protocols

- when a node has packets 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

Random Access Protocols

- Random access schemes are completely *decentralized*

- Variety of different random access schemes:
  - Users transmit at will – *Pure Aloha*
  - Users are constrained to transmit in certain time intervals only – *Slotted Aloha*
  - Users "listen" to channel before they transmit and they transmit only if there is an idle medium – *802.11 (Wi-Fi)*
  - Users may use random access to request permission or reservation to transmit a full message at some later time (reservation technique) – *3G/LTE*

Pure Aloha

- Developed at Univ. of Hawaii in the early 1970's
  - It was developed to communicate over various islands in a wireless manner
  - Fore-runner of all random access schemes proposed / adopted since

- A user may transmit information at any time