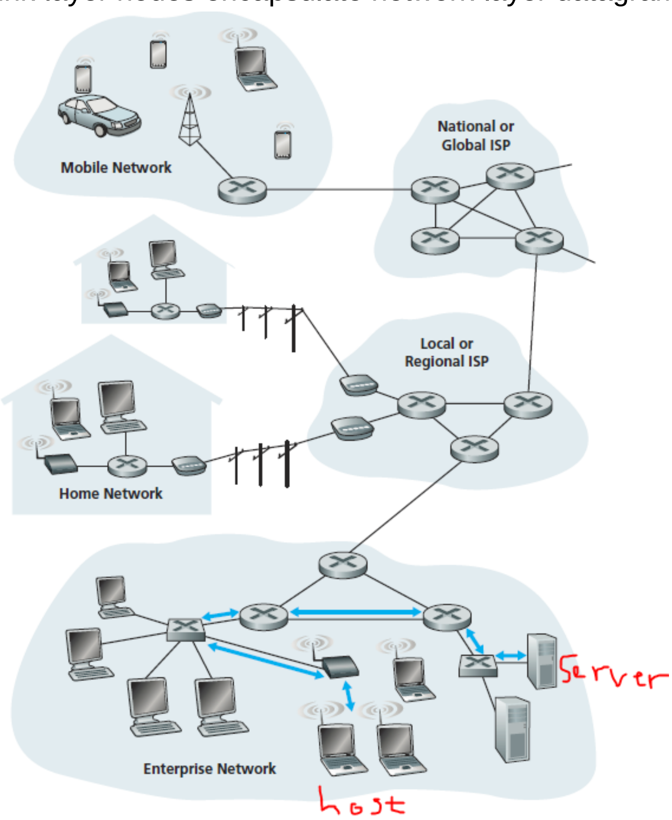
Chapter 5: Link Layer

<u>ATNLP (layers)</u> <u>(PLaNeT-A)</u>
Application
Transport
Network
Link
Physical

5.1 Introduction to the Link Layer

- **Node**: hosts, routers, switches, wifi access points
- Link-layer nodes encapsulate network-layer datagrams in **link-layer frames**



- Figure 5.1 ♦ Six link-layer hops between wireless host and server

5.1.1 The (possible) Services Provided by the Link-Layer

- Framing
  - encapsulation
- Link Access
  - protocols for shared access links
  - MAC (medium access control) protocol - rules for transmitting frames onto links
  - Depends on duplex - half / full ?
    - Refers to the transmission of data in two directions simultaneously. For example, a telephone is a **full**-**duplex** device because both parties can talk at once. In contrast, a walkie-talkie is a half-**duplex**device because only one party can transmit at a time.
- Reliable delivery

- important for media with high error rates (802.11)
- Error detection / correction

5.1.2 Where is the Link Layer implemented?

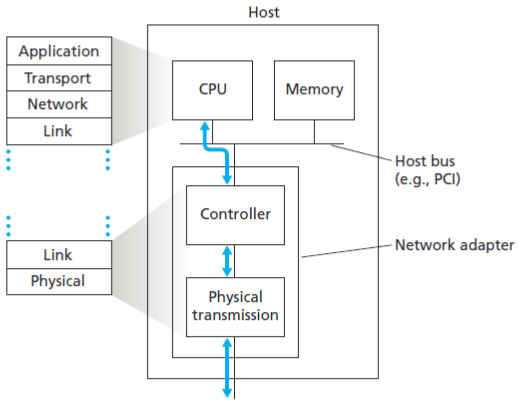
- In a router's NIC (Network Interface Card) / network adapter
- Mostly implemented in hardware
- NICs used to be separate chips - now typically built onto motherboard
- Sending side:
    - NIC takes datagram (created by higher layers) from memory and wraps it in a link-layer frame
    - (Sets error detection bits)
    - Then transmits the frame into the communication link
- Receiving side:
    - NIC receives entire frame and extracts datagram
    - (performs error detection)
- 

Figure 5.2 ♦ Network adapter: its relationship to other host components and to protocol stack functionality

5.2 Error Detection and Correction Techniques

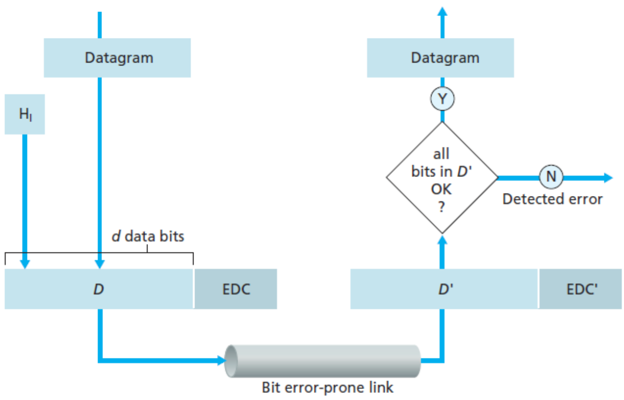
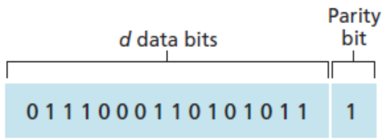
- Link-layer often provides bit-level error detection and correction
- 

Figure 5.3 ♦ Error-detection and -correction scenario
- EDC = error detection and correction bits
- **NB**: ***false positives are possible*** → error can be detected despite no error having occurred
    - ie: due to in-transit bit flips
- **NB: *errors can go undetected***

- Link-layer error detection allows the receiver to sometimes/often detect errors, <u>but not always</u>
- **<u>Three basic techniques used in link-layer error detection:</u>**
    - **Parity checks**
    - **Checksumming**
    - **Cyclic redundancy checks**

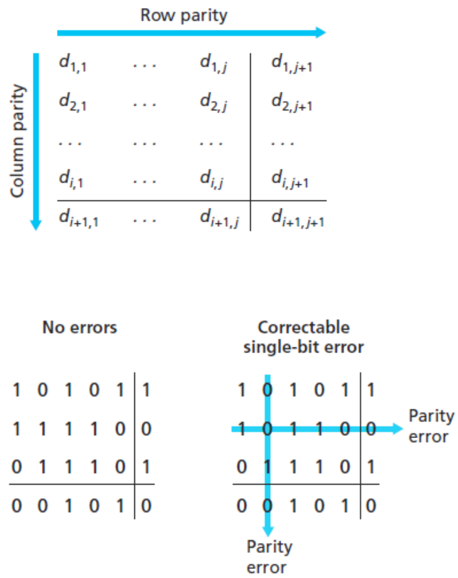
5.2.1 Parity Checks

- <u>Simplest form of error detection:</u> *single parity bit*
- Suppose message is D bits
    - In an even, single parity bit scheme, sender includes 1 extra bit
    - The value of the extra bit must make for an even number of 1s in the message, in total
    - Receiver detects an error if (D+extra bit) has an odd number of 1s
        - Knows that some *odd* number of bit errors have occurred
    - Odd parity scheme works similarly (want odd number of 1s)



- Figure 5.4 ♦ One-bit even parity

- If using even parity bit scheme, receiver will not detect errors if there is an *even* number of bit errors
- **Two-dimensional even parity:**
    - Receiver can both **detect** and **locate** exact bit errors
    - But cannot **correct** bit errors



- Figure 5.5 ♦ Two-dimensional even parity

- **Forward Error Correction (FEC):** ability of receiver to both **detect** and **correct** bit errors
    - Correction of error at receiver prevents transport-layer NAK + retransmission

5.2.2 Checksumming methods

- In checksumming techniques, the D bits comprising a message are treated as a sequence of K-bit integers
- Simple checksumming technique: USED IN TRANSPORT LAYER:
    - *Sum these K-bit integers and use the sum as the error-detection bits*
    - **Internet checksum** is based on this approach
        - Bytes of data treated as 16-bit integers and summed
        - 1s complement of this sum forms the internet checksum that is carried in the packet header
    - Receiver checks for errors by taking 1s complement of the sum of the received data
        - If any of the bits in the sum is a 0, an error is detected
- Link layer uses **cyclic redundancy check** rather than 1s complement checksumming
- Transport layer uses checksumming because it needs a simple, low overhead technique
- Link layer uses cyclic redundancy check because it can use advanced hardware operations to handle a more complex, robust method

5.2.3 Cyclic Redundancy Check (CRC)

- CRC codes (AKA polynomial codes) allow us to view a bit string as a polynomial whose coefficients are the 0 and 1 values it contains
    - **CRC → view bit string as polynomial**
- Uses remainder of long polynomial division
- Advantages of CRC:
    - Easy hardware implementation
    - Can analyse mathematically
    - Really good at error detection
- Used in Ethernet, 802.11 wifi, ATM

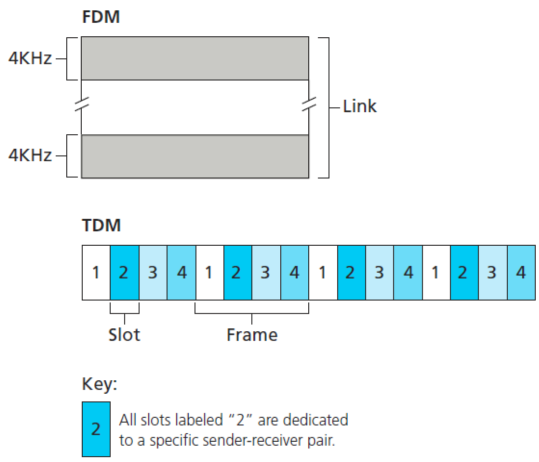
5.3 Multiple Access Links and Protocols

- **2 types of network links:**
    - **Point to point link**
        - Sender at one end, receiver at other end
    - **Broadcast link**
        - Multiple senders & receivers connected to same link
        - When a node transmits a frame, the channel broadcasts a copy of the frame to all other nodes
- **Multiple access problem:**
    - *How to coordinate a shared link with multiple sender/receiver nodes on either end?*
    - Solution = use **multiple access protocols**
- **Multiple access protocols:**
    - Regulate exchanging of messages over a shared broadcast channel/link

- *If no multiple access protocol used:*
    - Suppose 2 nodes broadcast messages simultaneously → all nodes in network receive 2 frames at once → frames ‘collide’ at receivers → become ‘tangled & jumbled’
- ***Examples of multiple access channels:***
    - Cable access network
    - WiFi
    - Satellite network
    - Cocktail party
- ***Types of multiple access protocols:***
    - Channel partitioning protocols
    - Random access protocols
    - Taking-turns protocols
- ***Desirable characteristics of a multiple access protocol:***
    - (suppose broadcast channel has rate *R* bits per second)
    - If only one node sending data, it'll have throughput of *R* bps
    - If *M* nodes sending data, each node has (average) throughput of *R/M* bps
    - Decentralised
    - Simple → inexpensive

5.3.1 Channel Partitioning Protocols

- FDM, TDM, CDMA
- (Recall) Techniques to partition bandwidth in a shared broadcast channel:
    - **TDM (Time Division Multiplexing)**
        - Divides time into frames, frames into time-slots
        - Each node assigned a slot → can only transmit packets during its time slot
        - **Pros:**
            - No collisions
            - Perfectly fair
        - **Cons:**
            - Node throughput constrained by fixed slots
            - Time wasted → a node will wait around until its timeslot
    - **FDM (Frequency Division Multiplexing)**
        - Divides channel into frequencies (each with bandwidth R/N)
            - “sub-channels”
        - Each of the N nodes assigned a frequency
            - **Pros (similar to TDM):**
                - No collisions
                - Perfectly fair
            - **Cons (similar to TDM):**
                - Node throughput constrained by fixed channel throughput
                - Time wasted?



- Figure 5.9 ♦ A four-node TDM and FDM example

- **Code Division Multiple Access (CDMA)**
  - Instead of assigning timeslots (TDM) or frequencies (FDM), assigns a **code** to each node
  - Each node uses its unique code to encode the data it sends
  - If codes are chosen well, multiple nodes can broadcast simultaneously, with receivers correctly receiving each unique message

5.3.2 Random Access Protocols

- (Second class of multiple access protocols)
- Nodes always transmit at full rate of the channel (*R* bps)
- If collision at receiver, sending nodes repeatedly retransmit until the message is received without collision
  - **Waits a random delay before retransmitting**
- **Cons:**
  - No collision avoidance
  - Wasted space + time
- Example of a random access protocol: **Slotted ALOHA**

5.3.3 Taking-Turns Protocols

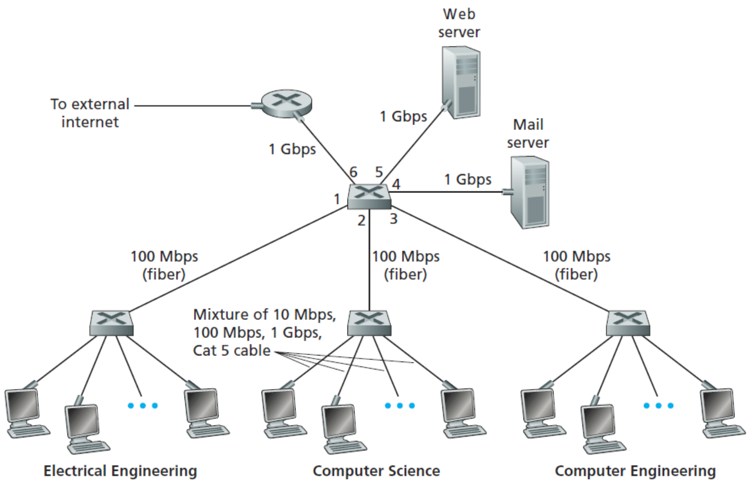
- Allows for a single node's (for M=1) throughput to be R bps
  - As well as an average throughput of R/M bps for each node if M>1
- **Types of taking-turns protocols:**
  - **Polling protocol**
    - Master node 'polls' other nodes in round-robin fashion
    - Master node messages other nodes to tell them when they're allowed to transmit, and how many frames they can transmit
    - **Pros:**
      - No collisions, wasted space or time
    - **Cons:**
      - Polling delay
      - Master node dead → whole channel dead
  - **Token-passing protocol**
    - No master node

        - **‘Token frame’** is passed from node to node in a fixed order
        - If a node is holding the token frame, it’s allowed to transmit (there is a cap on how much it can transmit)
        - **Pros:**
            - Decentralised
            - Highly efficient
        - **Cons:**
            - One node dying can still crash channel
            - Nodes can ‘forget’ to pass on the token → expensive recovery procedure to get the token back on track

5.3.4 DOCSIS: The Link-Layer Protocol for Cable Internet Access

- Cable access network connects several thousand residential modems to CMTS
- DOCSIS (data over cable service interface specifications)
    - Specifies cable network architecture + protocols
    - Uses FDM
    - Frames transmitted from CMTS on downstream channel are received by all residential modems on that channel *without collisions* (one-to-many)
    - Frames transmitted from residential modems on upstream channel are *not always received by the CMTS without collisions* (many-to-one)
- **How does DOCSIS control / prevent collisions on the upstream?**
    - CMTS sends MAP messages on downstream → tells modems which **time slots** they’re allowed to transmit during
    - Modems send ‘time slot request frames’
        - Random order
        - Can collide with each other at CMTS
        - If no response to request frame → assume there was a collision and wait a while before retransmitting

5.4 Switched Local Area Networks (LANs)



- **Figure 5.15** ♦ An institutional network connected together by four switches

5.4.1 Link-Layer Addressing and ARP

- Hosts and routers have link-layer addresses (completely separate from network layer / IP addresses)

- (it's actually the **interfaces** of these hardware devices that have link-layer addresses)
- These addresses are called **MAC addresses**
- Address Resolution Protocol (ARP) used to translate IP addresses to (link-layer) MAC addresses

MAC Addresses

- Hosts and routers have **link-layer addresses** corresponding to their **interfaces**
- **NB:** link-layer switches don't MAC addresses
  - Since link-layer switches have the sole purpose of forwarding datagrams
- A MAC address is 6 bytes long → 2^48 possible addresses
  - Not structured hierarchically like an IP address → flat structure
- No 2 adapters have the same MAC address → doesn't matter where manufactured in the world
  - This is because IEEE manages the MAC address space
  - IEEE allocates fixes the first 24 bits of MAC addresses, allows vendors to set the rest
- When an adapter wants to send a frame to another adapter in the LAN, it inserts the destination adapter's MAC address into the frame and then transmits the frame into the LAN
- Special MAC **broadcast address:**
  - If inserted in a frame, the frame will be broadcasted to **all** adapters in the LAN
  - FF-FF-FF-FF-FF-FF → hexadecimal

Address Resolution Protocol (ARP)

- We need a way to convert/map between network-layer (IP) addresses and link-layer (MAC) addresses
  - Solution = ARP

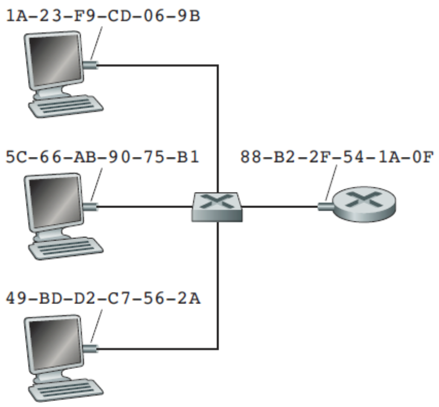


Figure 5.16 ♦ Each interface connected to a LAN has a unique MAC address

-
- At the network-layer, IP addresses are used to send datagrams between hosts. But a host's sending adapter needs the link-layer MAC address of the adapter at the destination host
  - ARP allows this to happen by mapping between IPs and MACs

How does ARP work (*within an internal subnet)*?

- Each host and router has an ARP table in memory
    - Contains IP<->MAC mappings, as well as a TTL for each mapping (indicates when the mapping will be deleted)

IP Address	MAC Address	TTL
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

    - **Figure 5.18** ♦ A possible ARP table in 222.222.222.220
- If a sending host doesn't have the receiver host's MAC address (in addition to its IP address), it must send an **ARP packet** to the destination via the ARP protocol, requesting the MAC address corresponding to that destination host's IP address
- Sender's adapter puts the ARP packet in a link-layer frame, uses the broadcast address as the frame's destination, and transmits the frame into the LAN
    - Each adapter in the network receives the broadcasted ARP packet and checks to see if the contained destination IP address is its own IP
    - The adapter with a match sends back an ARP packet containing its IP<->MAC mapping (in a normal frame, not broadcast, since it has the IP address of the sender), which the original sending host can put in its ARP table
- ARP is mostly a link-layer protocol, but deals with IP addresses in addition to MAC addresses, so is somewhat of a mix between a link-layer and network-layer protocol

How does ARP work *(when datagrams are being sent across subnets)*?

- More complicated than the case of internal subnet ARP operations

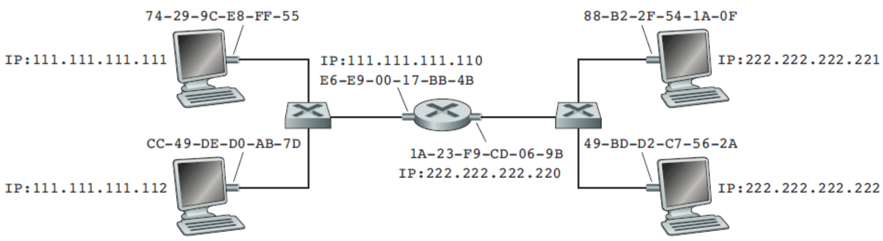


Figure 5.19 ♦ Two subnets interconnected by a router

-
- Each host has one IP address and one adapter
- Each router has an IP address for each of its interfaces
- The router in 5.19 has two interfaces and thus:
    - An IP for each interface
    - An ARP module for each interface
    - An adapter for each interface
        - A MAC address for each adapter
- Suppose that 111.111.111.111 wants to send a datagram to 222.222.222.222:
    - 111....111 passes the datagram to its adapter, and tells the adapter to send the datagram to the first-hop router with IP=111...110 and MAC=E6-E9….BB-4B
        - Datagram encapsulated in a frame and sent to the first-hop router
        - MAC address obtained using ARP
        - **NB** that the first MAC destination for the packet is the first-hop router in the current subnet, **NOT** the mac address of the destination host's adapter interface in the destination subnet

- After first-hop router has received the datagram, it checks its forwarding table and thus sends the datagram via the interface 222.222.222.220
- This interface passes the datagram to its adapter, which puts the datagram in a frame and sends the frame into subnet 2
- The destination MAC of the frame will now be, thanks to ARP, the actual destination in subnet 2.
- Thus, the datagram will be sent to 222.222.222.222

5.4.2 Ethernet

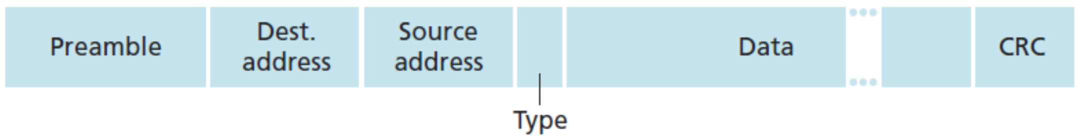
- Most prevalent wired LAN technology
- **Hub:** physical layer device that acts on individual bits rather than frames
  - Ethernet hub broadcasts arriving bits onto all other interfaces
  - If hub receives frames from multiple different interfaces simultaneously, collision can occur
- **Switch:** collision-free alternative to hub → replaced ethernet hubs in 2000s
  - While routers operate up till layer 3, switches only operate up till layer 2
- 

  Figure 5.20 ♦ Ethernet frame structure

Ethernet Frame Structure

- Sending adapter encapsulated IP datagram within Ethernet frame
- Passes frame to physical layer
- Receiving adapter receives frame from physical layer
- Extracts IP datagram → passes to network layer
- **Ethernet frames have 6 fields**
  - Data field → MTU = 1500 bytes, otherwise fragmented datagram
  - Destination address
  - Source address
  - Type field → ethernet can support different network-layer protocols
  - Cyclic redundancy check (CRC) → detect bit errors
  - Preamble → used to synchronise receiver clocks with sender clock
- **Ethernet provides connectionless service to network layer**
  - → no adapter handshaking required
- **Ethernet provides unreliable service to network layer**
  - → no ACKs/NAKs sent
  - If CRC check detects error, frame is simply discarded
  - <u>Consequence:</u> Ethernet is cheap and simple, but streams of datagrams may have gaps
- **<u>Ethernet's CSMA/CD protocol solves the 'multiple access problem'</u>**
- *Repeater*: physical-layer device which receives signal on input side and regenerates the signal on the output side
- **Ethernet can be carried over a number of different physical media:**

    - Coaxial cable
    - Copper wire
    - Fiber
- *Ethernet's frame format hasn't changed since the protocol's invention, despite all the other upgrades*

5.4.3 Link-Layer Switches

- *Role of a switch:* receive incoming link-layer frames and forward them onto outgoing links
- A switch is **transparent** to hosts and routers → even though their frames pass through it, they don't specifically address frames to the switch
- Just like with routers, the rate at which frames arrive can exceed the output capacity of a switch's outgoing interface
    - So switches, again like routers, have buffers at their interfaces
- Switches are **plug-and-play devices** → require no human intervention other than connecting LAN cables to it
- Switches are **full-duplex** → any switch interface can send and receive frames simultaneously

Forwarding and Filtering

- **Filtering:**
    - Switch function which decides whether a frame is forwarded to an interface or dropped
- **Forwarding :**
    - Switch function which transmits frames to the appropriate interfaces
- **Switch table:**
    - Used by *filtering* and *forwarding* functions
    - Contains entries for (some or all) hosts and routers on the LAN
        - Each entry contains a MAC address, the outgoing switch interface pointing to that MAC address, and the creation time of the entry

Address	Interface	Time
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
....	....	....

Figure 5.22 ♦ Portion of a switch table for the uppermost switch in Figure 5.15

        -
    - **NB difference between switch forwarding and router forwarding functions:**
        - Routers forward datagrams, switches forward frames
        - Routers forward packets to IP addresses, switches forward packets to MAC addresses
    - **How does switch filtering and forwarding work?**
        - Suppose a frame with dest MAC DD...DD arrives at the switch's interface x
        - Switch will check its table for DD...DD
        - *3 possible scenarios:*

- There is no entry in the table for DD-DD-DD-DD-DD-DD. In this case, the switch forwards copies of the frame to the output buffers preceding *all* interfaces except for interface *x*. In other words, if there is no entry for the destination address, the switch broadcasts the frame.
- There is an entry in the table, associating DD-DD-DD-DD-DD-DD with interface *x*. In this case, the frame is coming from a LAN segment that contains adapter DD-DD-DD-DD-DD-DD. There being no need to forward the frame to any of the other interfaces, the switch performs the filtering function by discarding the frame.
- There is an entry in the table, associating DD-DD-DD-DD-DD-DD with interface $y \neq x$. In this case, the frame needs to be forwarded to the LAN segment attached to interface *y*. The switch performs its forwarding function by putting the frame in an output buffer that precedes interface *y*.

Self-learning: automatic switch table configuration

- Table initially empty
- For each incoming frame on an interface, switch stores in its table the:
  - MAC address from frame's *source* field
  - Interface which sent the frame
  - Current time
- Switch deletes a table entry if, after the **aging time** period, no frame with that entry's MAC address as the source address has been received
  - If a PC is replaced by another PC, the original PC's MAC address will be discarded from the table after the aging time

Properties of link-layer switching:

- *No collisions*
  - Frames are buffered → never forward multiple frames on an outgoing interface simultaneously
- *Heterogenous links*
  - Different links in LAN can operate at different speeds → mix old and new equipment
- *Automatic network management*
  - Can automatically disconnect a malfunctioning router, for example
- Security
  - Less vulnerable to sniffing than hubs and wireless LANs

Switches vs Routers

- **Router =** store-and-forward packet switch which forwards packets using network-layer addresses
- **Switch =** store-and-forward packet switch which forwards packets using MAC addresses
- **Pros and cons of switches:**

Pros	**Cons**
Plug and play	Uses spanning tree topology → hard to construct
High throughput	Large network = large ARP tables = heavy processing

	Broadcast storms → one host can go crazy and broadcast loads of frames → network collapse

- **Pros and cons of routers:**
  -

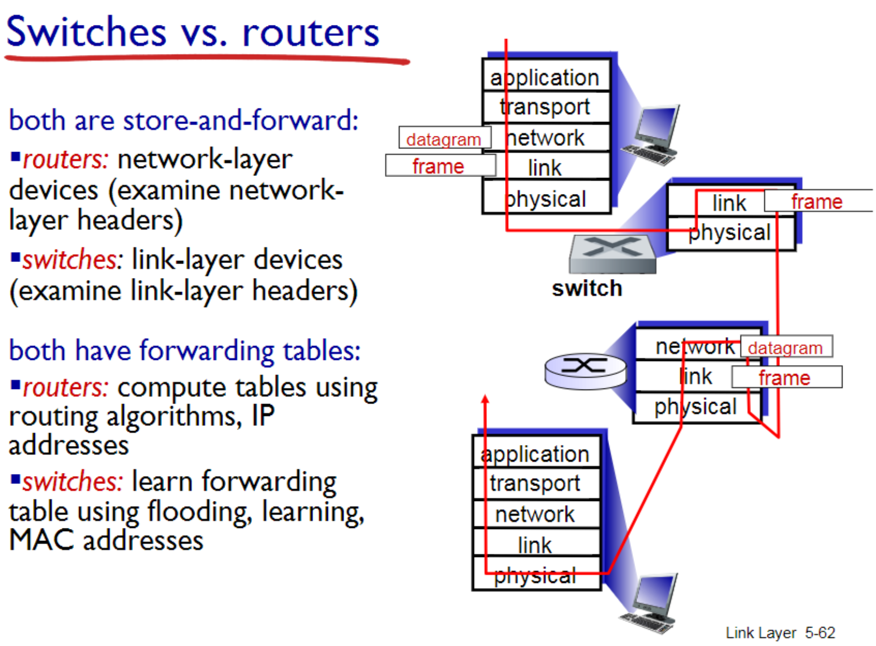
Pros	Cons
Hierarchical addresses → no cycling loops → not limited to spanning tree topology	Not plug and play → manual IP configuration (or DHCP..?)

-

	Hubs	Routers	Switches
Traffic isolation	No	Yes	Yes
Plug and play	Yes	No	Yes
Optimal routing	No	Yes	No

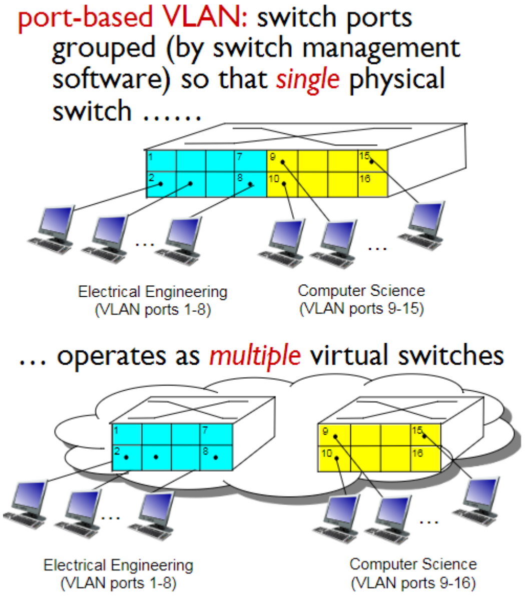
Table 5.1 ♦ Comparison of the typical features of popular interconnection devices

-



5.4.4 Virtual Local Area Networks (VLANs)

- Institutional LANs often configured hierarchically
  - Each department has its own switched LAN connected to the other switched LANs
  - Not always feasible IRL → has drawbacks
- Alternative = **VLAN**

- VLANS allow *virtual LANs* to be defined within a single switch
    - Hosts within in a switch’s VLAN think that they are the only ones connected to the switch → meanwhile, there can be other VLANs configured in the switch
- **Port-based VLAN:**
    - The switch’s ports (interfaces) are divided into groups, and each group of interfaces belongs to a different VLAN within the switch
    - 

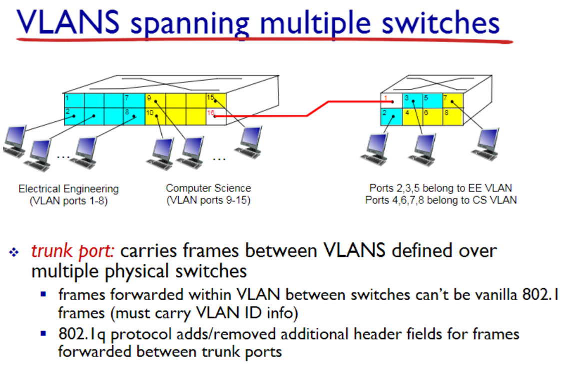
Problem:

- A switch contains VLAN 1 and VLAN 2. How can packets be exchanged between VLAN 1 and 2?

Solution 1:

- **VLAN switch port**
- port/interface which is shared by both VLANs
- <u>Not a scalable approach</u>
    - Not feasible if frames need to be exchanged between VLANs that are defined over multiple physical switches

Solution 2:

- **VLAN trunking**
- Uses ‘VLAN tag’
- <u>Scalable approach</u>
    - Can be used to exchange frames between VLANs that are defined over multiple physical switches
    - 

5.6 Data Centre Networking (for cloud applications)

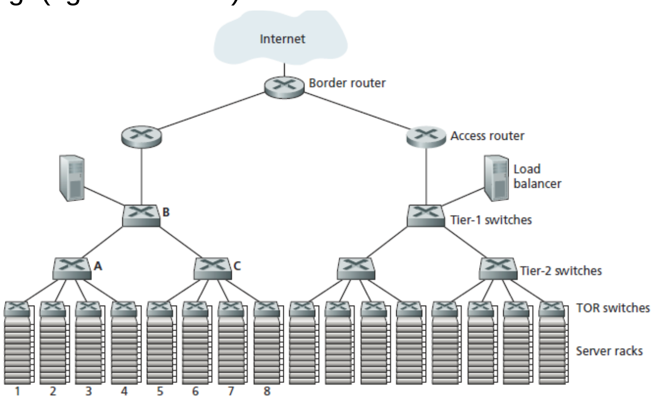
- Google, Apple, etc. have massive data centres → each has a data centre network
- **Blades =** hosts in a data centre
  - Resemble ‘pizza boxes’ (...?)
- Blades/hosts are stacked in racks
  - At top of each rack → **TOR (top of rack) switch**
  - **TOR** connects hosts in rack to each other
- Data centre network supports 2 types of traffic:
  - Traffic between internal hosts/blades and external internet
  - Traffic between internal hosts/blades
- **Border routers** used to handle external<->internal traffic

Load Balancing

- Consider a Google datacentre
- Provides many applications: search, email, video, etc.
- Each of these applications has a publicly visible IP address which external clients can use to access the service/application
  - External requests are managed by the datacentre’s **load balancer (layer 4 switch)**
    - → distributes requests to the hosts

Hierarchical architecture

- Small data with simple network:
  - Border router
  - Load balancer
  - Single ethernet switch connecting 50 or so racks of blades/hosts
- But big data centre needs a **hierarchy of routers and switches** → deal with scale
  - eg: (fig 5.30 below):



  - **Figure 5.30** ♦ A data center network with a hierarchical topology
  - Multiple access routers under main border router, multiple switches under each access router, each switch is in charge of a bunch of racks (containing blades/hosts), each with TOR switches
  - May also have to use VLAN subnets

- Conventional hierarchical architecture **solves scale problem** but **suffers from *limited host-to-host capacity problem***
    - Constrains rate of flow between hosts/blades in different racks
    - **Solution:** higher-rate switches
        - But this is expensive

Trends in Data Centre Networking

- (1) Replace hierarchy of switches with a **fully connected topology**
- (2) Put mini data centres in shipping containers and ship them to different locations around the world