12. Link Layer

---

1. Link Layer
2. Link Layer Services
3. Transmission Errors
4. Parity Bits
5. Checksumming Methods
6. Cyclic Redundancy Checks
7. Hamming Distance
8. Error Correction
9. RAC Parity
10. Error Correction with RAC Parity
11. Point-to-point Communication
12. Point-to-Point Protocol
13. Byte Stuffing
14. Multiple Access
15. Multiple Access Protocols
16. Channel Partitioning Protocols
17. Frequency Division Multiplexing
18. Time Division Multiplexing
19. Taking-Turns Protocols
20. Random Access Protocols
21. Ethernet
22. Ethernet Hub-Based Star Topology
23. CSMA/CD
24. Collisions
25. Detecting Collisions
26. Handling Collisions
27. Link-Layer Addressing
28. Address Resolution Protocol
29. How ARP Works
30. Ethernet Frame
31. Ethernet Header Example
32. Manchester Encoding
33. Manchester Encoding Preamble
34. Extending LANs - Repeaters
35. Limitations of Repeaters
36. Bridges
37. Frame Filtering
38. Switches
39. Data Centre Networks
40. Load Balancing
41. Switch Connections
42. Case Study - UPnP
43. UPnP Stages
44. Links to more information

12.1. Link Layer

- the link layer is responsible for transporting information from one host (or router) to another over a *single* link
- each network-layer datagram is encapsulated in a link-layer *frame*
- two fundamentally different types of link-layer channels:
    - *broadcast* channels
        - common in local area networks (LANs), wireless LANs, etc.
        - many hosts connected to the same communications channel
        - *medium access protocol* is needed to coordinate transmissions
    - *point-to-point* communications link
        - used between two routers or home dial-up modem and ISP router
        - coordination is trivial
        - still issues around framing, reliable transfer etc.

12.2. Link Layer Services

- *framing*: encapsulation of network datagram within a link-layer frame
- *link access*: a medium access (MAC) protocol specifies the rules by which a frame is transmitted onto the link
- *reliable delivery*: useful for links prone to high error rates; avoids cost of end-to-end retransmission at transport or application layer
- *flow control*: frames can be lost if buffering capacity is exceeded
- *error detection*: usually more sophisticated than Internet checksum and implemented in hardware
- *error correction*: possible to correct errors as well as detect them

12.3. Transmission Errors

- electro-magnetic interference can introduce unwanted electrical currents into the electronic components or wires used for communication
- severe interference (e.g. lightning) can cause permanent damage to network equipment
- more often, interference changes the electrical signal without damaging the equipment
- this can cause the receiver to misinterpret one or more bits of data
- lost, changed, or spuriously appearing bits are collectively called *transmission errors*
- handling transmission errors adds complexity to networks:
    - *detecting* errors
    - *correcting* errors

12.4. Parity Bits

- simplest form of error detection involves adding a *parity bit* to each block (e.g., byte) of data bits
- the sender computes an additional bit based on the given *d* data bits and transmits this as bit *d+1*
- the receiver performs the same computation and verifies that the parity bits agree
- this ensures that a one-bit alteration can be *detected*
- parity can be *even* or *odd*
- for even (odd) parity, the sender sets the parity bit so that the total number of 1 bits is even (odd):

Original Data	Even Parity	Odd Parity
0 0 0 0 0 0 0 0	0	1
0 1 0 1 1 0 1 1	1	0
0 1 0 1 0 1 0 1	0	1
1 1 1 1 1 1 1 1	0	1
1 0 0 0 0 0 0 0	1	0
0 1 0 0 1 0 0 1	1	0

- errors in fact often happen in "bursts", so simple parity checks are usually not enough

12.5. Checksumming Methods

- in checksumming methods, the *d* data bits are viewed as a sequence of *k*-bit integers
- the integers are summed and the resulting sum used as the error-detection bits
- recall that the Internet checksum considers the data bits as 16-bit integers and produces a 16-bit *checksum*
- recall also that checksums are not able to detect all common errors

12.6. Cyclic Redundancy Checks

- *Cyclic Redundancy Check* (CRC) can detect more errors without increasing the amount of additional information
- hardware necessary to generate CRCs is very simple
- just need an *exclusive or* (XOR) unit and a *shift register*
- CRCs are too complex to be covered here (see [Kurose and Ross] or [Tanenbaum] for further details)
- 16-bit CRC can detect all
  - single-bit errors
  - double-bit errors
  - errors involving an odd number of bits
  - *burst errors* of length 16 or less
  - as well as a number of additional errors
- electrical interference (lightning, and other sparks) often produces burst errors
- Ethernet uses a 32-bit CRC (see later)

12.7. Hamming Distance

- a fixed size unit of data comprising data and check bits is called a *codeword*
- the number of bit positions in which two codewords differ is called the *Hamming distance*
- if two codewords are distance *d* apart, it will require *d* single-bit errors to convert one to the other
- in the parity example below

Dataword	Codeword
0 0	0 0 1
0 1	0 1 0
1 0	1 0 0
1 1	1 1 1

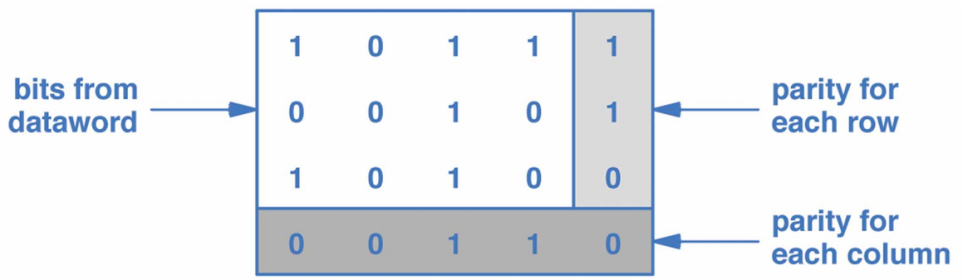
the distance between any pair of codewords is 2
- so *minimum* Hamming distance d_{min} is 2
- the maximum number e of bit errors that can be detected is: $e = d_{min} - 1$

12.8. Error Correction

- error correction can be used to reduce the number of retransmissions or with channels where retransmission cannot be requested
- but most often error detection followed by retransmission is preferred because it is more efficient
- a simple method for correcting single bit errors is to transmit each bit three times
  - `0` becomes (codeword) `000`
  - `1` becomes (codeword) `111`
- if `001`, `010`, or `100` is received, the original message was `0`
- if `110`, `101`, or `011` is received, the original message was `1`
- it is possible to do better than this method, where each codeword is 3 times the length of the original data

12.9. RAC Parity

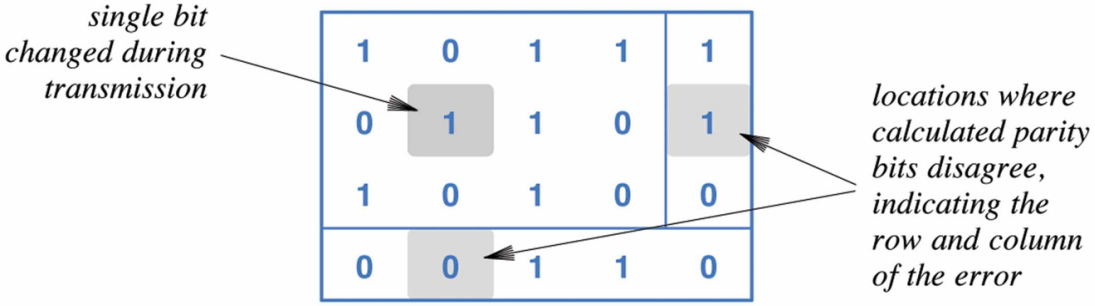
- RAC (row and column) parity (also called 2-dimensional parity) provides a more efficient encoding
- let each dataword comprise k bits
- assume r bits are added to form a codeword
- this is called an *(n,k) encoding scheme*, where $n = k + r$
- assume each dataword comprises $k = 12$ bits
- think of the bits arranged into 3 rows and 4 columns



- the example RAC is a *(20,12)* encoding

12.10. Error Correction with RAC Parity

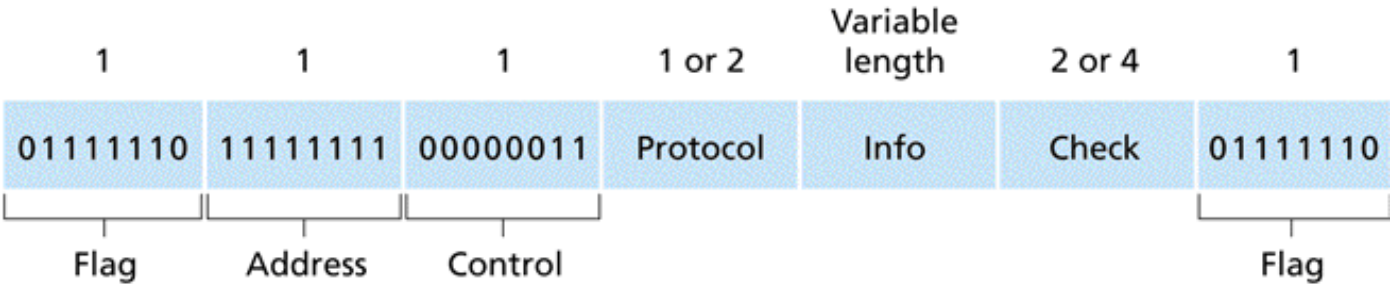
- a single-bit error will cause a parity error in both a row i and a column j
- the erroneous bit can be identified by the "cell" (i, j)
- this is shown in the following example



12.11. Point-to-point Communication

- early communication systems had each communication channel, e.g. a leased line, connecting exactly two computers
  - also the case for a dial-up link from a home computer to an ISP, and between some routers
- known as *point-to-point communication* and has three useful properties:
  - each channel is independent and can use appropriate hardware
  - the two end points have exclusive access and can decide how to send data across the channel
  - since only two computers have access to the channel, it is easy to enforce security and privacy
- one common protocol is the *point-to-point protocol* (PPP)

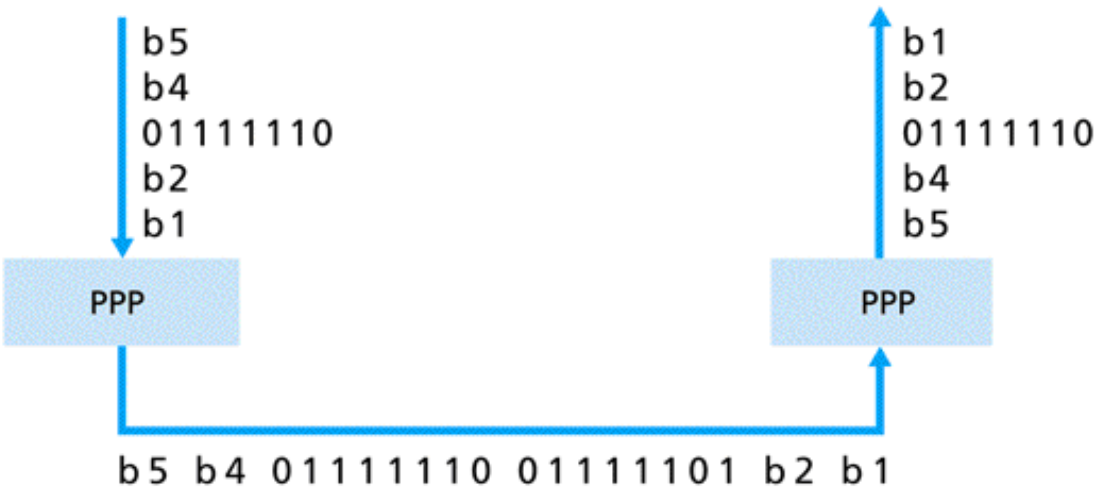
12.12. Point-to-Point Protocol



- every PPP frame starts and ends with a one-byte *flag field* with a value of `01111110` (`0x7E` hexadecimal)
- the *address* and *control* fields are always set to the same values (allows for future development)
- the *protocol* field tells the PPP receiver the upper-layer protocol to which the encapsulated data belongs (IP has value `0x21`)
- the *information* field is of variable length and carries the encapsulated data
- the *check* field is a two- or four-byte CRC

12.13. Byte Stuffing

- a protocol such a PPP uses a specific bit pattern (`0x7E`) to delimit a frame
- what happens if this pattern appears as data within the frame
- a technique called *byte stuffing* resolves this problem
- another (escape) sequence is used to mark occurrences of reserved sequences in the data
- PPP uses `01111101` (`0x7D`), as illustrated below



- if `0x7D` appears in the original data, it must also be preceded by `0x7D`

12.14. Multiple Access

- recall that an alternative to a point-to-point link is a broadcast link
- as typically used for LANs
- the problem now is how to coordinate the access of multiple sending and receiving nodes to a shared broadcast channel
- this is called the *multiple access problem*
- co-ordination requires communication, and the time to communicate depends on distance
- so this mechanism does not scale over long distances

12.15. Multiple Access Protocols

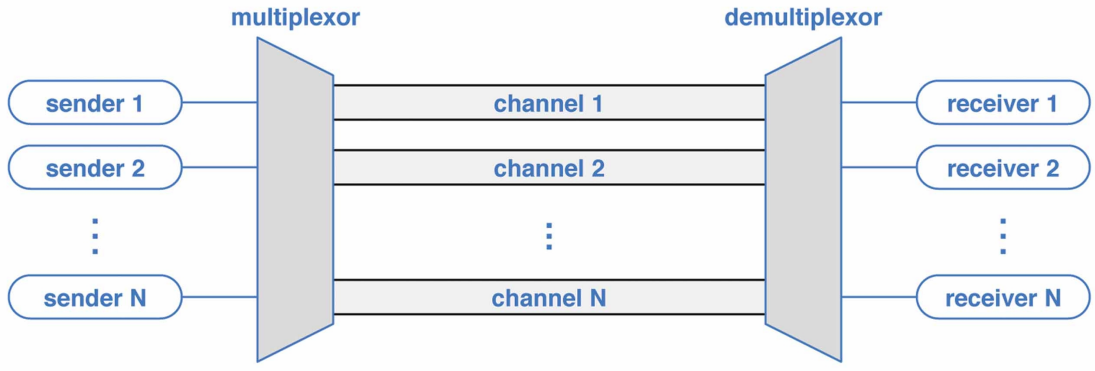
- many multiple access protocols have been developed over the years
- can essentially be divided into three categories:
  - *channel partitioning protocols*
  - *random access protocols*
  - *taking-turns protocols*
- ideally, a multiple access protocol for a broadcast channel of rate R bits per second should have the following characteristics:
  - when only one node has data to send, that node has throughput R bps
  - when M nodes have data to send, each has a throughput of R/M bps
  - the protocol is decentralised, i.e., no master node
  - the protocol is simple

12.16. Channel Partitioning Protocols

- a channel can be partitioned using *multiplexing*
- four basic approaches to multiplexing:
  - frequency division multiplexing
  - time division multiplexing
  - wavelength division multiplexing (used for optical fibre)
  - code division multiplexing (used for mobile phones)
- we will consider only the first two

12.17. Frequency Division Multiplexing

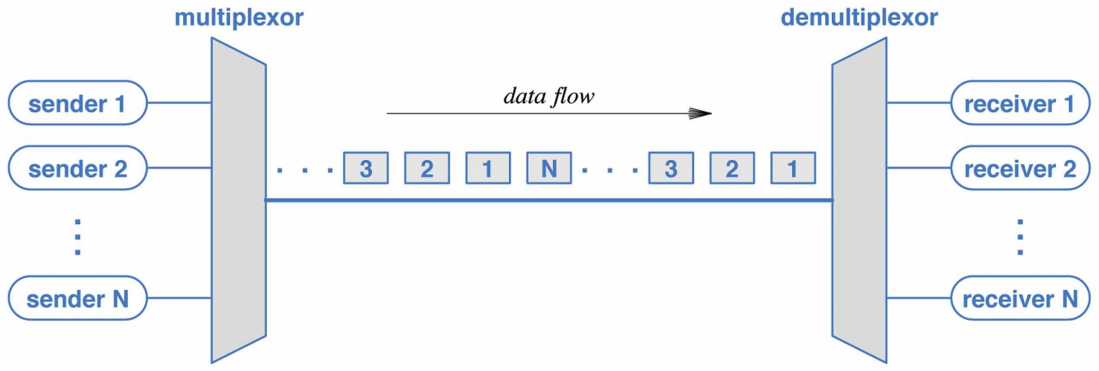
- *Frequency Division Multiplexing* (FDM) is used for broadcast radio
- radio stations can transmit signals simultaneously
- provided they each use a separate *channel* (i.e. carrier frequency)
- same principle can be used for data communications, as illustrated below



- there is a practical limit on set of frequencies that can be used for channels
- if they are too close, interference can occur
- a set of carrier frequencies is chosen with a gap, known as a *guard band*, between them
- the problem is that the throughput drops when fewer than *N* nodes are sending data

12.18. Time Division Multiplexing

- *Time Division Multiplexing* (TDM) is simpler than FDM
- simply means transmitting an item from one source, then from another, and so on, as shown below



- figure shows items being sent in a *round-robin* order
- problem:
  - the throughput drops when fewer than *N* nodes are sending data
  - each node must wait for its turn

12.19. Taking-Turns Protocols

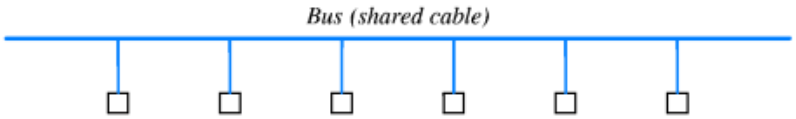
- we will mention just two such protocols
- in the *polling protocol*
  - one node is designated as a master node
  - the master node polls each of the nodes in a round-robin manner
  - each node is offered the chance to send some maximum number of frames
  - problems: polling delay and master can fail
- in the *token-passing protocol*
  - no master node
  - a special frame known as a *token* is passed among the nodes in some fixed order
  - a node holds onto the token if it has frames to send
  - otherwise it forwards the token to the next node
  - problems: if a node fails or neglects to pass the token

12.20. Random Access Protocols

- in a random access protocol, each node transmits at the full rate of the channel
- now transmitted frames can *collide* if nodes transmit at the same time
- when there is a collision, none of the receiving nodes can make sense of any of the transmitted frames
- all the frames involved in a collision are lost
- the broadcast channel is wasted during the collision interval
- when there is a collision, each node retransmits after a *random delay*
- there are many random access protocols, but we will consider only *Ethernet*

12.21. Ethernet

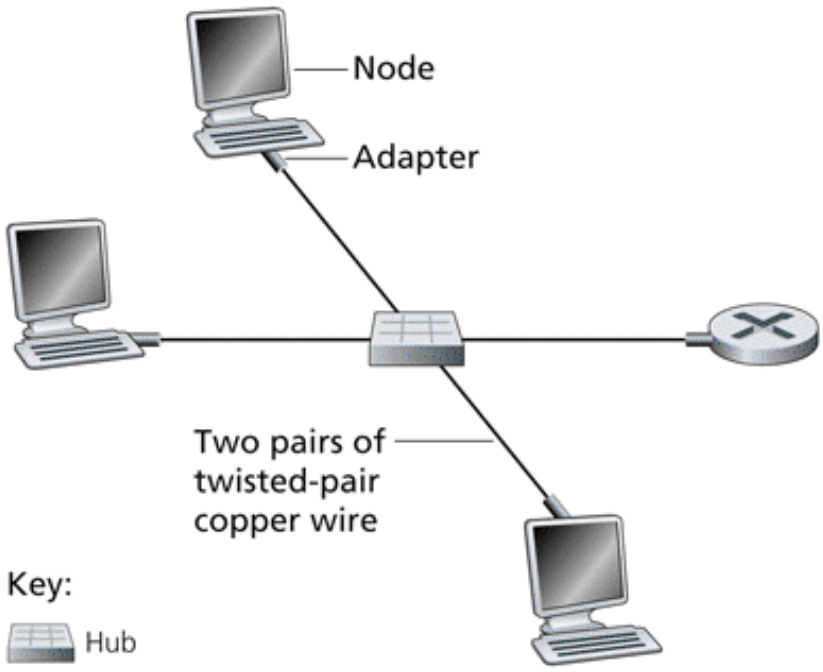
- original standard was published by Digital Equipment Corporation, Intel Corporation, and Xerox Corporation in 1982
- IEEE currently controls Ethernet standards
- in its original form, an Ethernet LAN consisted of a single coaxial cable called the *ether*, but often referred to as a *segment*
- thus Ethernet used a so-called *bus* topology:



- a segment was limited to 500 m in length, with a minimum separation of 3 m between each pair of connections
- it operated at 10 Mbps
- newer versions operate at up to 40 Gbps

12.22. Ethernet Hub-Based Star Topology

- by the late 1990s most organisations had moved to a hub-based star topology



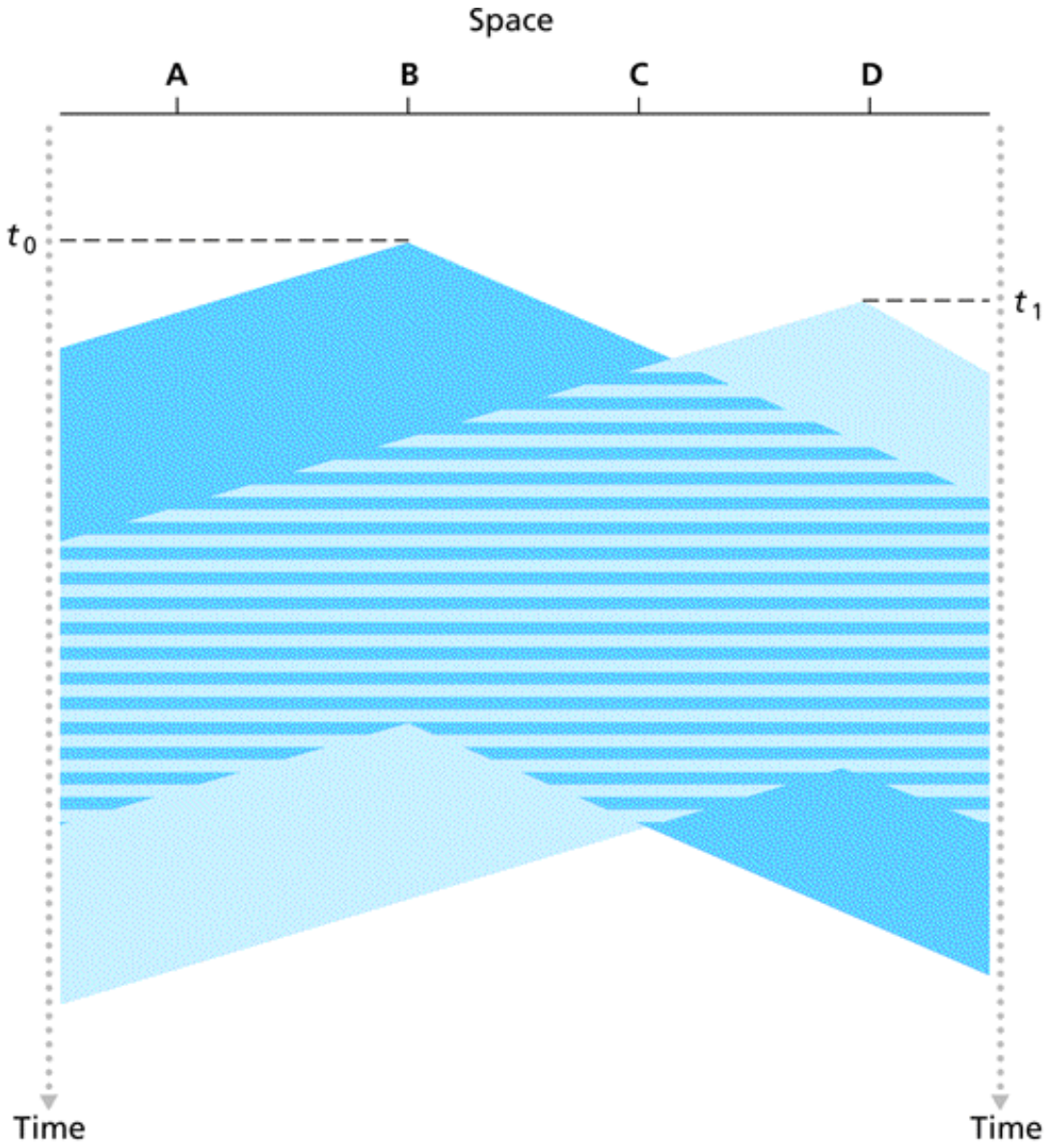
- each host and router is directly connected to a hub
- hub is a physical-layer device that acts on bits rather than frames
- when a bit arrives on one interface, it is transmitted on all other interfaces
- so this is also a broadcast LAN environment
- nowadays organisations use Ethernet *switches* (see later)

12.23. CSMA/CD

- all computers attached to the Ethernet use CSMA/CD to co-ordinate their activities
- CSMA/CD = *Carrier Sense Multiple Access/Collision Detection*
- a computer wishing to transmit checks for electrical activity on the cable, informally called a *carrier*
- if there is no carrier, the computer can transmit
- if a carrier is present, the computer waits for the sender to finish before proceeding

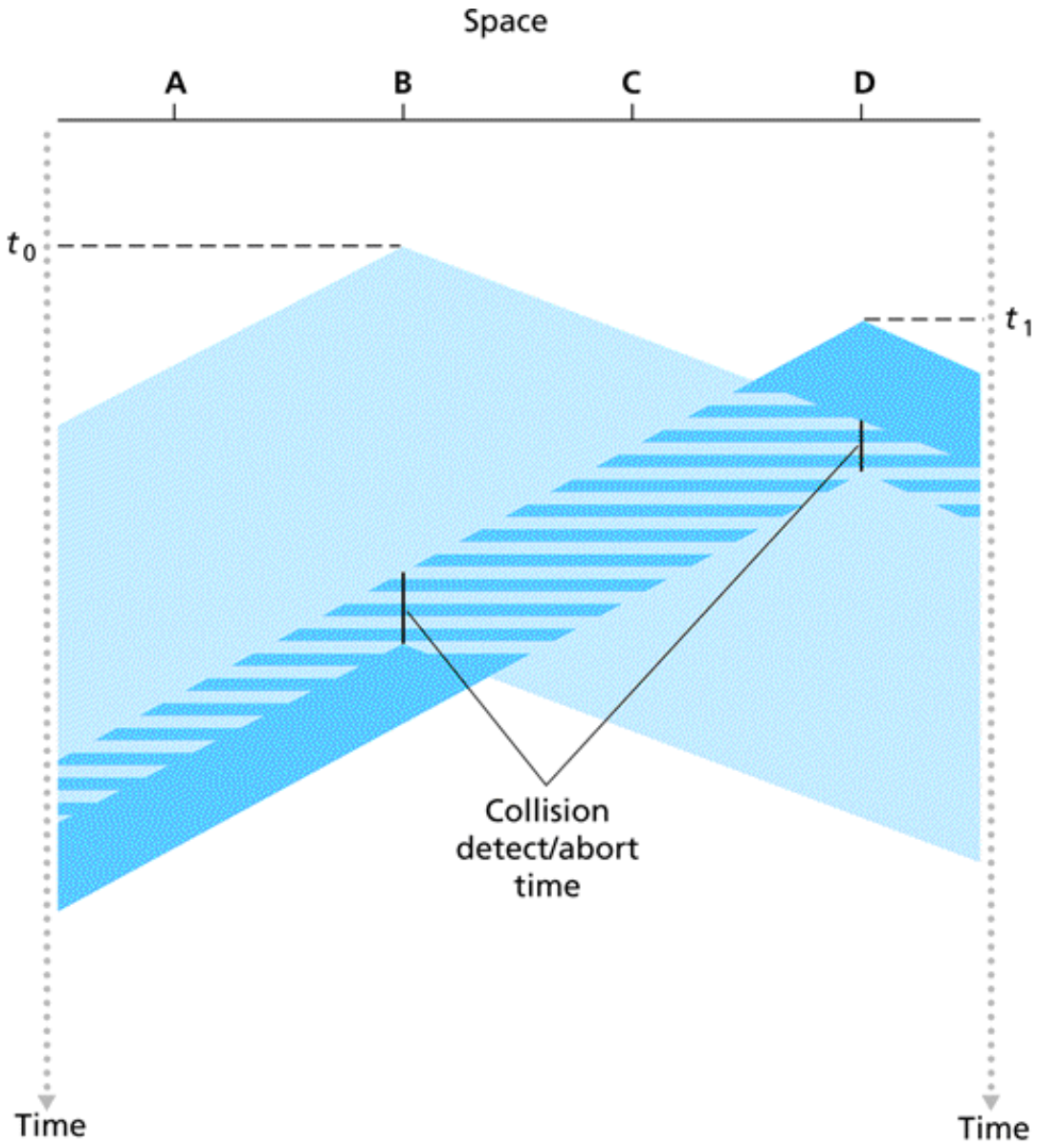
12.24. Collisions

- it is possible for two or more computers to detect the lack of carrier and start transmission (almost) simultaneously
- the signals then interfere with one another
- this interference is called a *collision*:



12.25. Detecting Collisions

- a sending computer monitors the signal on the cable
- if the signal differs from the signal it is sending, then a collision has occurred and the computer stops transmitting:

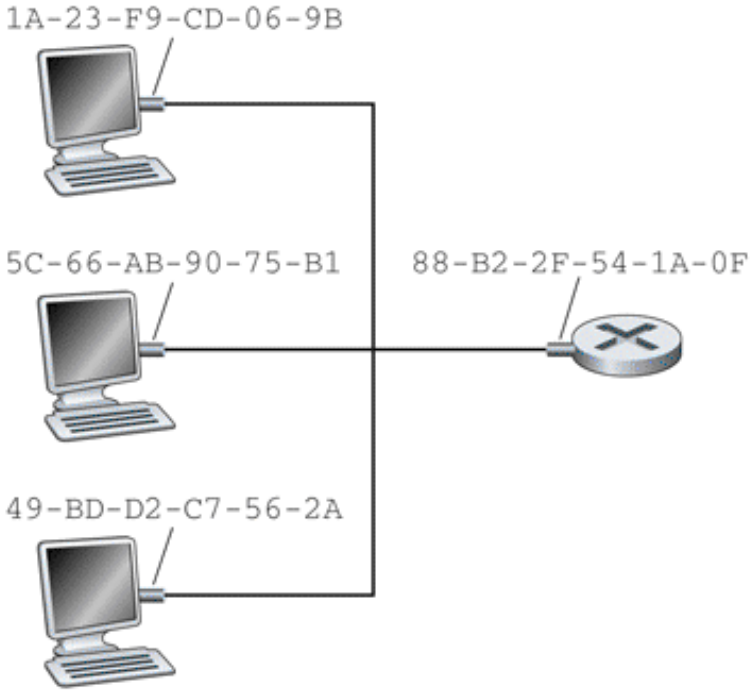


12.26. Handling Collisions

- following a collision, a computer waits for the cable to become idle before retransmitting
- however, if the computers start transmitting as soon as the cable becomes free, another collision will occur
- Ethernet requires each computer to *delay* after a collision
- the standard specifies a maximum delay, d, and requires each computer to choose a random delay less than d
- in this case, the computer choosing the shortest delay will transmit first
- if subsequent collisions still occur, the computers double the maximum delay ($2d$, $4d$, ...) until the range is large enough for one computer to choose a short delay and transmit without a collision
- this technique is called *binary exponential back-off*

12.27. Link-Layer Addressing

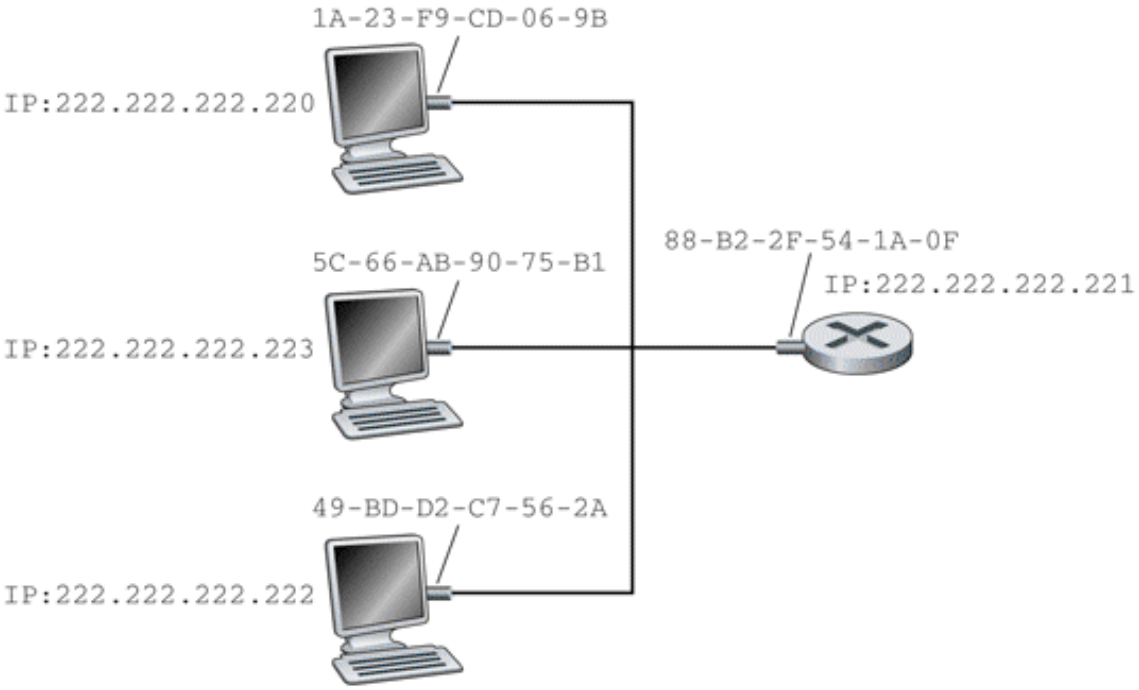
- each computer (interface) on the LAN is assigned a unique *physical address*
- also called a *hardware address* or *Media Access address* (MAC address)
- MAC addresses are 6 bytes (48 bits) long, usually written in hexadecimal, e.g., `1A-23-F9-CD-06-9B`



- IEEE manages the MAC address space, allocating blocks of addresses to manufacturers

12.28. Address Resolution Protocol

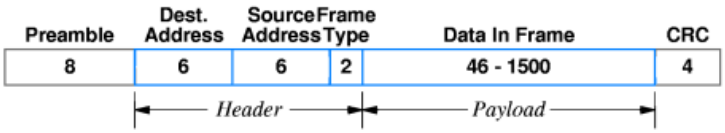
- addressing on LANs is via MAC addresses
- addressing at the network layer is via, e.g., IP addresses
- so we need to be able to translate between them
- this is done by the *Address Resolution Protocol* (ARP) for IPv4 (IPv6 uses ICMPv6)
- ARP takes an IP address *on the same LAN* and returns the corresponding MAC address



12.29. How ARP Works

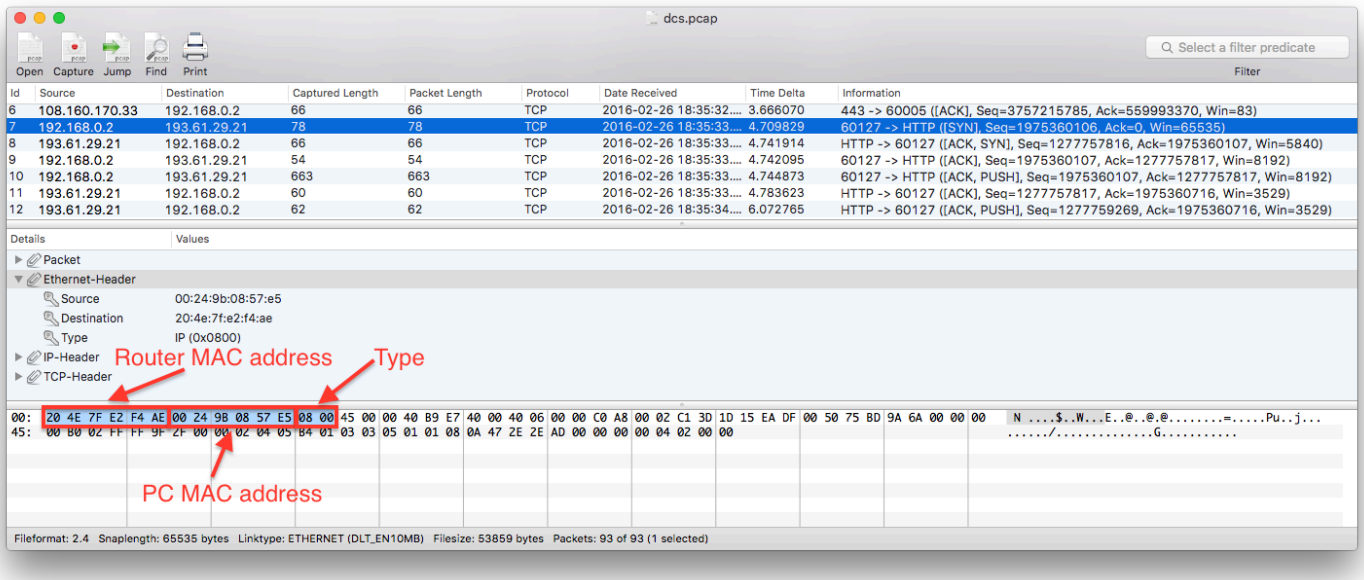
- each node (host or router) has an *ARP table* in memory, containing mappings
- how does the node determine the mappings?
  - it sends an ARP query packet to the broadcast MAC address `FF-FF-FF-FF-FF-FF`
  - the packet contains the desired IP address and the MAC address of the sender
  - the one node with the matching IP address sends a response ARP packet with its MAC address
- how does the node know a machine is on the same LAN (subnet)?
  - it found out the subnet mask from the DCHP server
  - the network prefix of the destination will match its own
- if the destination machine is not on the same LAN, the node uses the MAC address of the router (default gateway identified by DCHP)

12.30. Ethernet Frame



- the 64-bit *preamble* consists of alternating ones and zeros allowing the receiver to synchronise with the incoming signal
- followed by the header consisting of a 48-bit *destination address*, a 48-bit *source address*, and a 16-bit *frame type*
- payload can vary in length from a minimum of 46 bytes to a maximum of 1,500 bytes
- payload is followed by a 32-bit CRC
- Ethernet standard specifies the values that can be used in the header fields; for example
  - 48-bit address `FF:FF:FF:FF:FF:FF` (hexadecimal), i.e. all 1s, is reserved for broadcast
  - frame type value of `0800` (hexadecimal) is reserved for IPv4 traffic

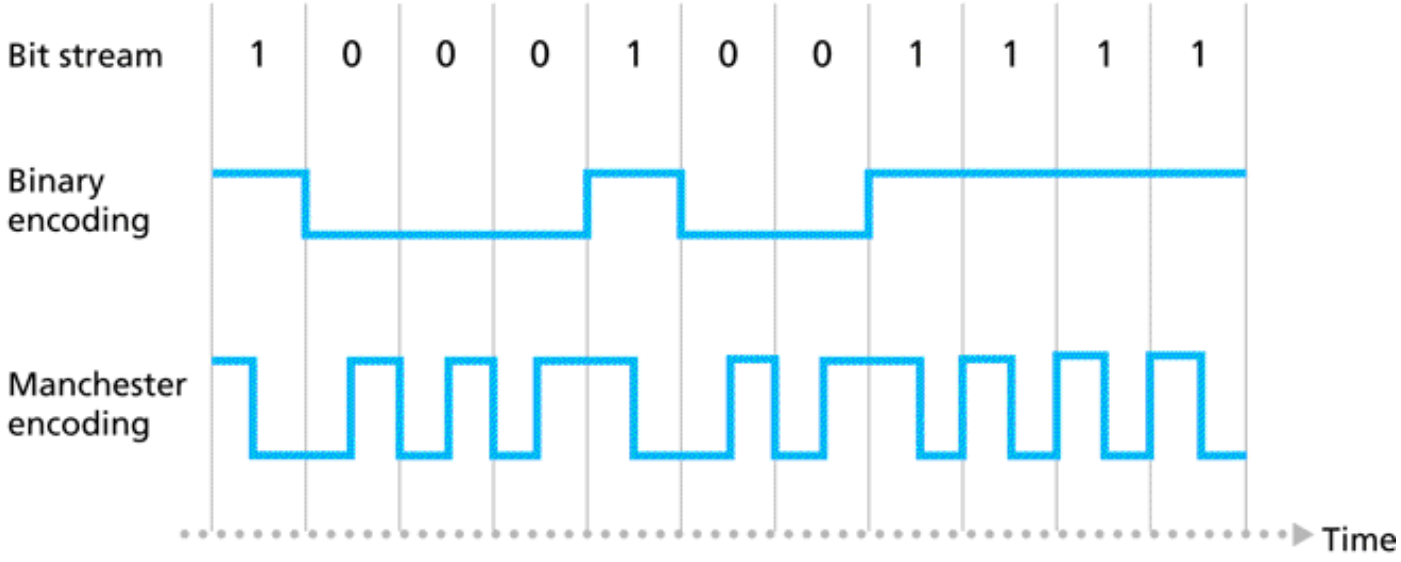
12.31. Ethernet Header Example



12.32. Manchester Encoding

- 10 Mbps Ethernet frames are transmitted using *Manchester Encoding*
- faster versions require more sophisticated and complex encodings
- Manchester Encoding uses the fact that hardware can detect a *change* in voltage more easily than a fixed value
- technically, the hardware is *edge triggered*, with the changes known as *rising* or *falling* edges
- the sender transmits
    - a falling edge to encode a 1
    - a rising edge to encode a 0

  as illustrated below



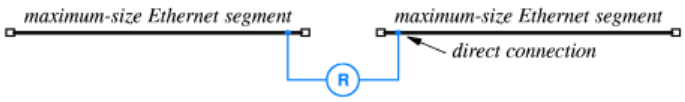
- the voltage change that encodes a bit occurs exactly half-way through the time slot
- if two contiguous bits have the same value, an additional change in voltage occurs at the edge of the time slot

12.33. Manchester Encoding Preamble

- Manchester Encoding uses the *preamble* to allow for synchronisation
- preamble consists of 64 alternating 1s and 0s sent before the frame
- these produce a square wave with transitions exactly in the middle of each slot
- receiving hardware uses the preamble to synchronise with the time slots
- the last two bits of the preamble are both 1s to signal the end of the preamble

12.34. Extending LANs - Repeaters

- electrical signals become weaker as they travel along a cable
- some LAN technologies allow two cables to be joined together by a device called a *repeater*
- when a repeater detects a signal on one cable, it transmits an amplified signal on the other cable, as illustrated below

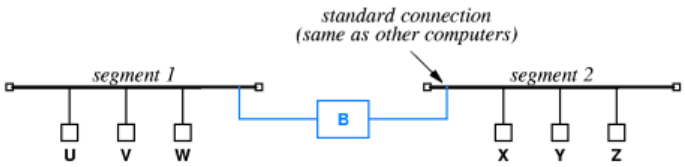


12.35. Limitations of Repeaters

- unfortunately, adding repeaters cannot be continued indefinitely
- the Ethernet standard requires low delay for CSMA/CD to work
- if the delay is too large, the scheme fails
- the most important disadvantage of a repeater is that it does not understand frames; it simply amplifies the electrical signal
- if a collision or electrical interference occurs on one segment, repeaters cause the same problem to occur on all other segments

12.36. Bridges

- a *bridge* connects two LAN segments using conventional interfaces and therefore handles complete frames
- the bridge listens to each segment in *promiscuous mode*:
  - means that it processes every frame, not just those addressed to it
- when a bridge receives a frame, it verifies that it arrived intact and forwards it to the other segment if necessary
- thus, two LAN segments connected via a bridge behave like a single LAN:



- bridges only forward complete, correct frames and not collisions or electrical interference

12.37. Frame Filtering

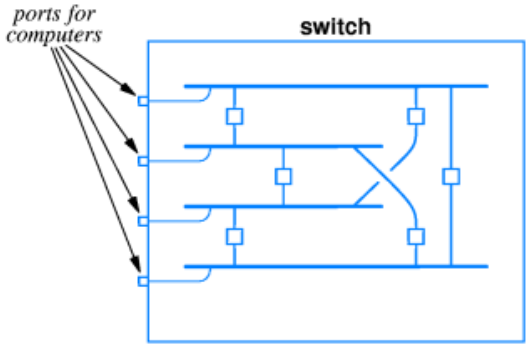
- a bridge also performs *frame filtering* - it only forwards frames if necessary
- it uses the source MAC addresses to build up a table of computers attached to each segment
- figure below illustrates how such an *adaptive* or *learning* bridge learns the locations of computers

Event	Segment 1	Segment 2	Frame Sent
Bridge boots	±	±	±
A sends to B	A	±	Both Segments
B sends to A	A, B	±	Segment 1 only
X broadcasts	A, B	X	Both Segments
Y sends to A	A, B	X, Y	Both Segments
Y sends to X	A, B	X, Y	Segment 2 only
X sends to Z	A, B	X, Y	Both Segments
Z sends to X	A, B	X, Y, Z	Segment 2 only

- A, B and C are on segment 1; X, Y and Z are on segment 2
- the first time computer A sends to B, the bridge has to forward the frame to segment 2
- when B sends to A, the bridge discovers that they are both on the same segment so does not forward the frame

12.38. Switches

- today, people mostly use Ethernet *switches*
- a switch has multiple *ports* that each attach to a single computer or a LAN segment
- a switch simulates a bridged LAN, as illustrated below

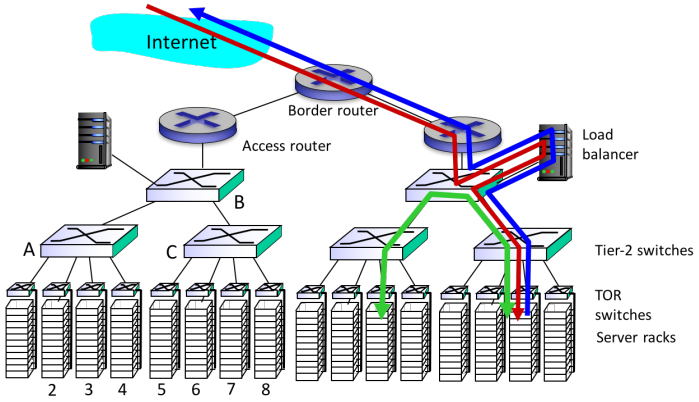


- the switch provides each computer with the illusion of a separate LAN segment connected to other segments via bridges
- one advantage of a switched LAN is parallelism:
  - a switch with *N* ports can transfer up to *N/2* frames simultaneously, provided they are independent
  - a switch also learns which computers are on which port
  - if only a single computer on each port, then no need for CSMA/CD

12.39. Data Centre Networks

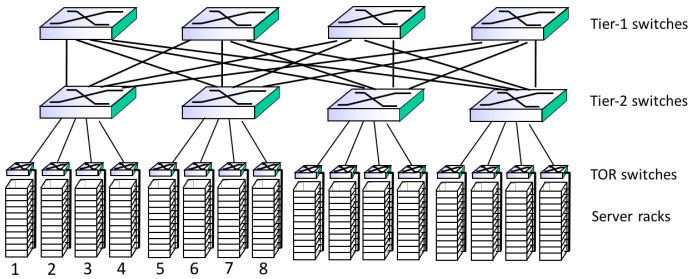
- 10's to 100's of thousands of machines in close proximity
- e.g., Amazon, Google, YouTube
- want to manage and balance load
- prevent bottlenecks
- machines are in racks, with many layers of switches

12.40. Load Balancing



- load balancing - application-layer routing
- receives external client requests
- directs workload within data centre
- returns results to external client (hiding data centre internals from client)
- TOR = top of rack

12.41. Switch Connections



- rich interconnection among switches, racks:
  - increased throughput between racks (multiple routing paths possible)
  - increased reliability via redundancy

12.42. Case Study - UPnP

- UPnP - Universal Plug and Play
- allows *devices* to join a network and offer/consume services
- example devices include:
  - printers
  - audio/video entertainment
  - home automation - heating/lighting
  - smart appliances
- also includes *control points* which send actions to devices

12.43. UPnP Stages

- Addressing - uses DHCP (and AutoIP):
  - device is assigned an address on the network
- Discovery - uses SSDP (Simple Service Discovery Protocol):
  - control points find devices
  - SSDP based on HTTP and UDP
- Description - uses XML over SOAP (Simple Object Access Protocol):
  - control points learn types and capabilities of devices
  - e.g., a media player and the formats it can handle
- Control - uses SOAP and XML for actions and responses:
  - e.g., increase the volume
- Eventing - uses GENA (Generic Event Notification Architecture):
  - control points listen for state changes in devices
  - e.g., device has stopped playing
  - GENA based on XML, HTTP and UDP

12.44. Links to more information

- The companion web site for Tanenbaum's book, Chapters 3 and 4
- UPnP specifications and resources

See Chapter 6 of [Kurose and Ross], Chapters 13 to 15 and 17 of [Comer] and parts of Chapters 3 and 4 of [Tanenbaum].