Contention Protocols and Networks

- Contention Protocols
  - CSMA/CD
- Network Topologies
- Ethernet

Contention Protocols

- Can be compared to conventional automobile traffic
  - Without any (traffic) rules, there would be countless collisions and chaos
  - Traffic lights can regulate traffic, and ensure a continual flow
- Network traffic follows a similar model

History of Ethernet

- Ethernet is a development of the Aloha radio network in Hawaii from 1970, for communications with remote nodes in difficult terrain
- Aloha protocol was one of the earliest contention protocols
  - Was designed to establish communication among the islands using a packet radio system

Aloha Protocol

- A single central node transmits packets on one radio channel to all remotes whenever necessary
- All remote nodes share a second radio channel on which they transmit back to the central site
- When the remote node has a message, it just sends it.
  - If it overlaps or 'collides' with another transmission, both are garbled and neither is acknowledged.
  - Both remotes wait a random time and retry
- Analogous to entering the motorway with your eyes closed, if you crash, wait a bit, get a new car, and try again

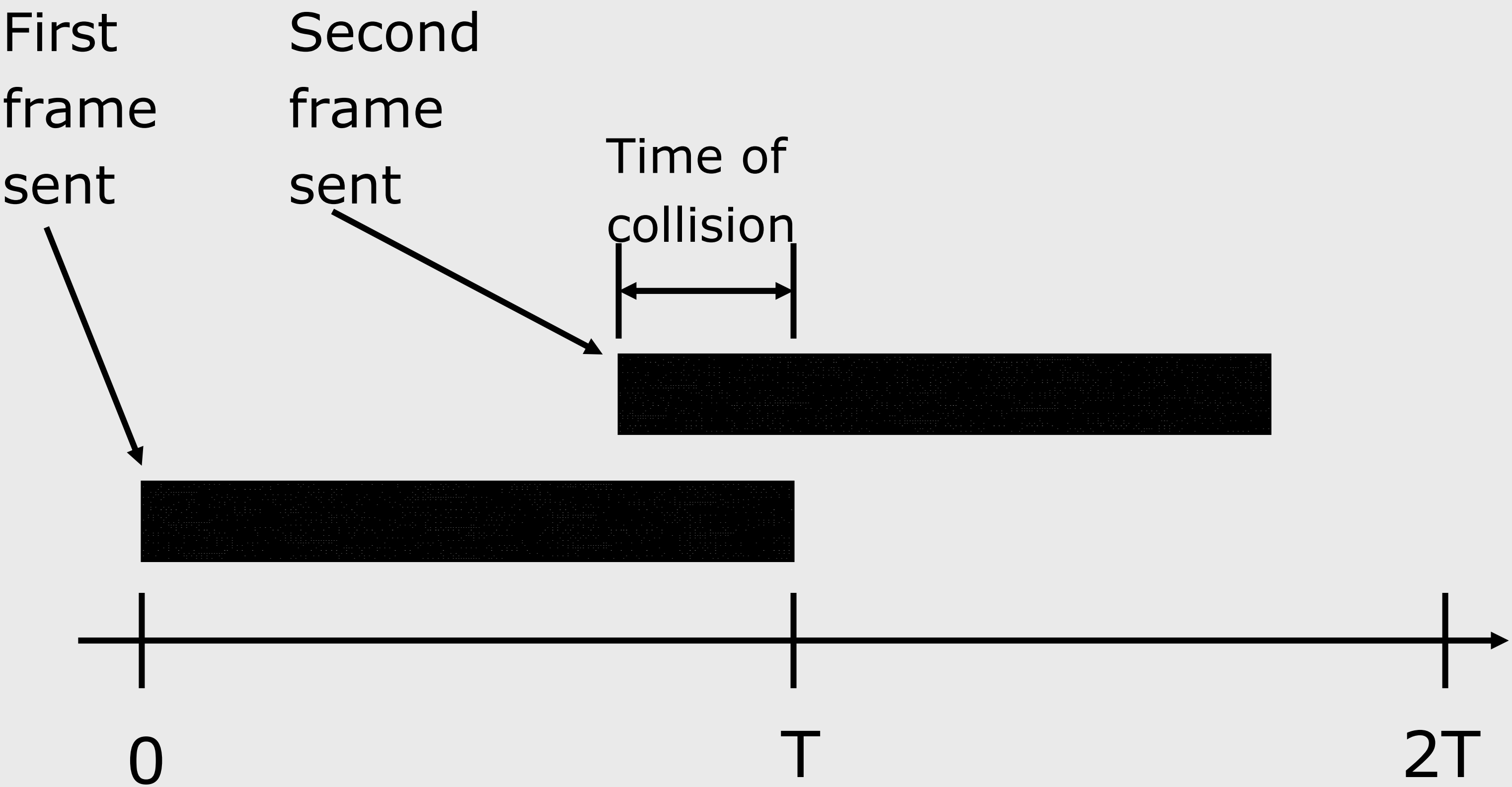
Aloha Protocol

- Collisions were detected easily
- When a frame was received an acknowledgement is sent
- Works well if there are not too many transmissions
  - Protocol is inefficient – doesn’t listen before transmitting

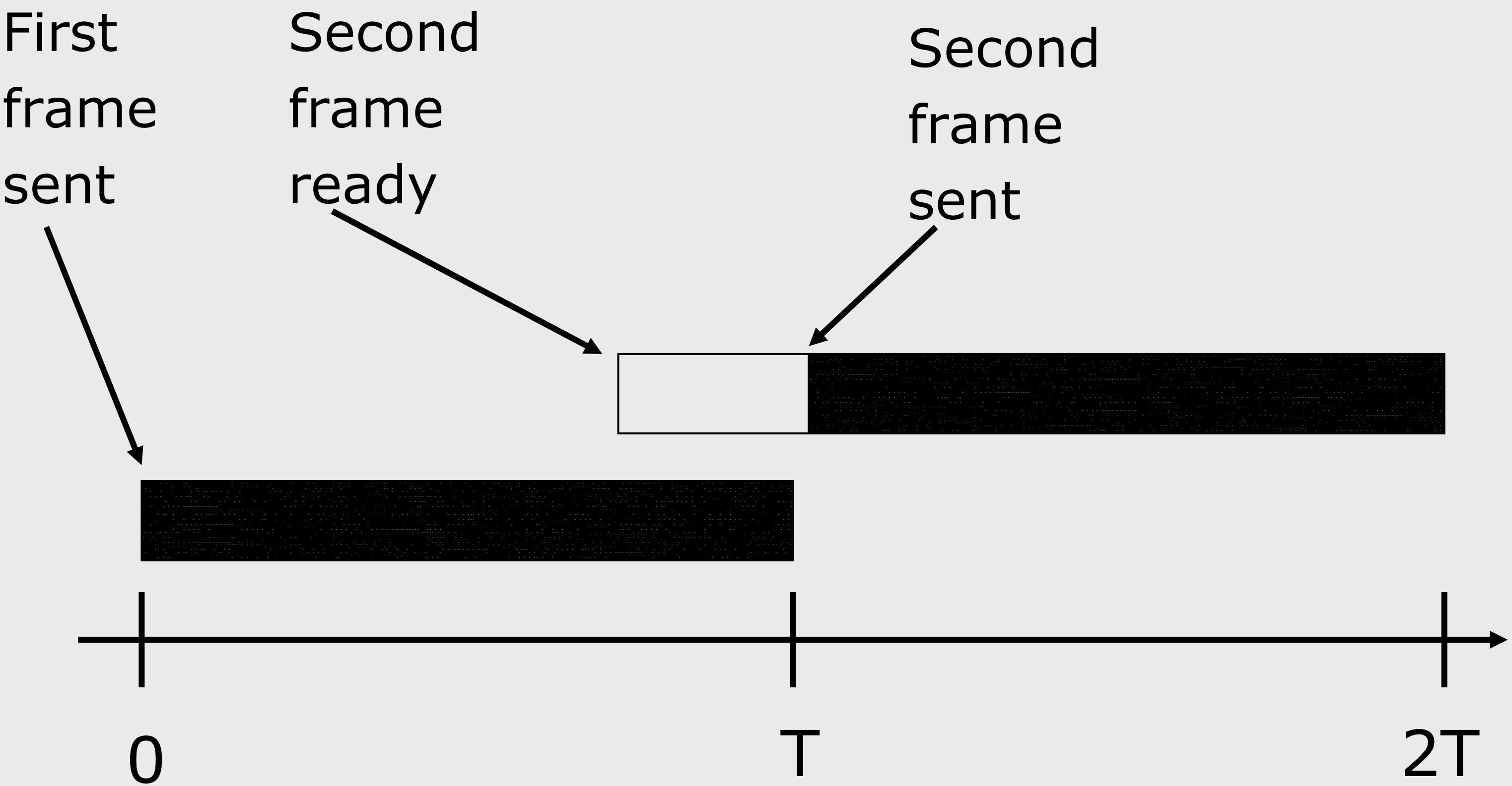
Slotted Aloha

- With higher traffic, we can divide the time for one transmission, into intervals (slots) of *T* units each
- Each station must begin transmission at the start of a slot.
  - Even if it is ready to transmit in the middle of a slot it must wait until the start of the next one
- Reduces collisions, as they now only occur when both stations are ready at the same slots
- Known as *slotted Aloha Protocol*

Pure Aloha



Slotted Aloha



CSMA/CD Protocol

- Ethernet is described as a CSMA/CD protocol
  - Carrier Sense Multiple Access, with Collision Detection
  - All stations can hear all transmissions (multi access)
  - Listen to the ‘medium’ for any activity (carrier sense)
  - If there is no activity, transmit; otherwise wait
- Reduces collisions

CSMA/CD

- With CSMA a collision can still occur if two, or more stations want to transmit at nearly the same time
- If both detect no activity, deduce it is OK to transmit
- Such collisions are infrequent

nonpersistant CSMA

- If station detects activity, it does not continue to monitor the medium
- Waits one time slot and checks again for activity
- At this point it transmits if idle, otherwise waits another time slot

p-persistent CSMA

- Station continues to monitor an active medium
- When it becomes quiet, station transmits with a probability, p. If $p = 1$, station always transmits. If $p = 0$ it always waits.
- Can still get collisions, if $p = 1$ for two or more stations wanting to transmit
- With more stations, collisions will occur more often

p-persistent CSMA

- Can reduce such frequency collisions by lowering *p*
  - Assume 0.5-persistent CSMA. Both stations are waiting on an idle medium, so each transmit with a *Pr*(0.5). One of 4 events happen with equal probability
    - Both transmit immediately
    - They both wait
    - The first sends and the second waits
    - The second sends and the first waits
  - There is a 0.5 probability that one will be able to send
  - There is 0.25 probability that neither sends

Collision Detection

- Reduce the time during which collisions occur
- Previous protocols, sent entire frame, and concludes there was a collision if it did not get a response
- During a collision, medium is also unusable to other stations
- Need a method for listening for collisions, and immediately stopping transmissions if a collision is detected

Collision Detection

- Both stations stop transmitting when a collision occurs, and send a jamming signal
  - The wasted time spans only part of the slot, and the time required to send a short jamming signal
- Usually used with CSMA, hence CSMA/CD
  - If the medium is busy, station waits per the persistence algorithm
  - If the medium is quiet, the station transmits the frame and continues to listen
  - If it detects a collision, it stops transmitting and sends a short jamming signal
  - After a collision it waits a random amount of time before trying to send again

Collision Detection

- The size of frame to send should be a function of the maximum time it takes to detect a collision
  - Need twice the distance to the other station (as the collision noise needs to travel back to the sending station)
  - In the worst case the time to detect is twice the time it takes a signal to travel the longest distance covered by the network
- A problem is then collision detection over long distance, such as satellite networks

Collision Detection

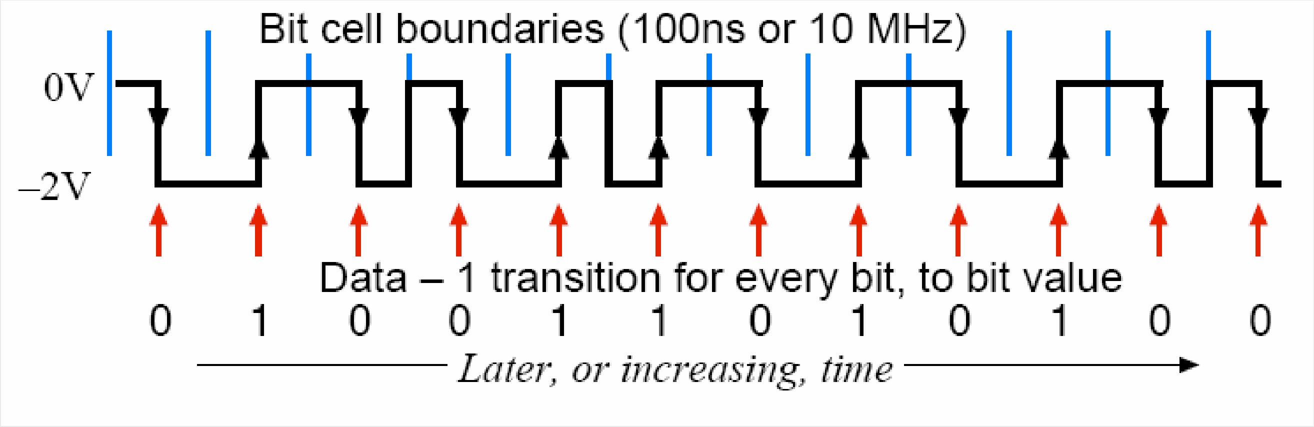
- Need to reduce transmission delays
- The number of slot times a station waits needs to be limited
- One technique 'binary exponential backoff' varies this limit
  - If a station's fame collides the first time, wait 0 or 1 time slots and try again
  - After a second collision wait 0, 1, 2, or 3 slots
  - After a third collision wait 0 – 7 slots
  - After *n* collisions wait 0 – 2^{n-1} slots if $n \leq 10$, else if $n > 10$, wait between 0 – 2^{10} slots
  - After 16 collisions give up (assume there is an error somewhere)

From Aloha to Ethernet

- Ethernet makes several changes on Aloha
  - Transmission is on a multi-dropped cable (giving *multiple access* for many stations) – all stations can hear every transmission
  - When a station wants to send, it first monitors the cable and waits if any other station is sending *(carrier sense*)
  - When a station does transmit, it monitors the cable (ether) for another station which sends at about the same time (*collision detection*)
  - A station which detects collision stops transmission immediately and 'defers' a random time before retrying
  - If it collides again, the deferral time is doubled for each collision to reduce the network traffic and ease congestion (*binary exponential back off*)

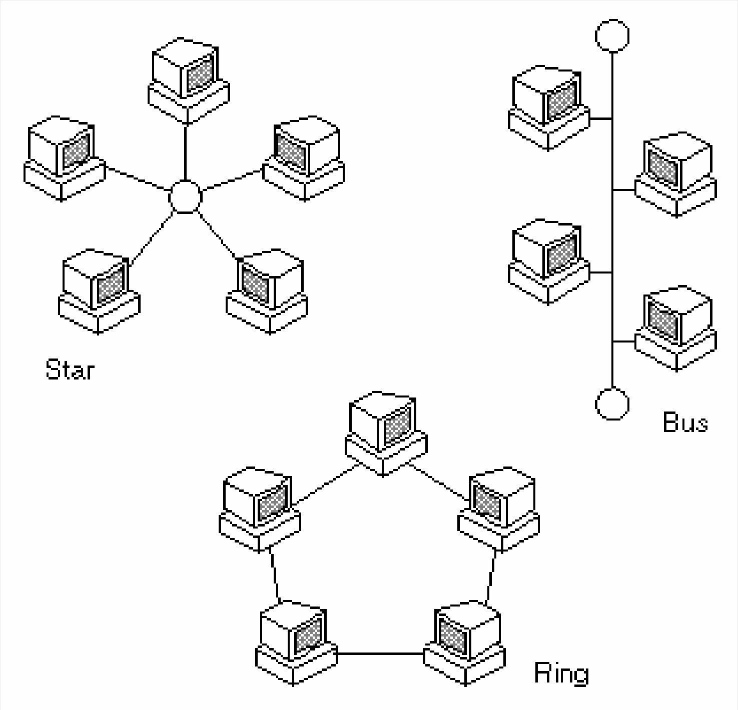
Original Ethernet – technical detail

- Signals are encoded with *Manchester coding* at 10Mbit/s with voltages of 0 - -2V on a 50Ω cable
- The 'bit cell' is divided into 2 halves, with a possible data transition in the middle of each cell
  - Transition –ve → +ve = 1
  - transition +ve → -ve = 0
- Data transitions are optionally provided at cell boundaries as needed



Network Topologies

- Traditional Ethernet used a bus topology



Token Ring Topologies

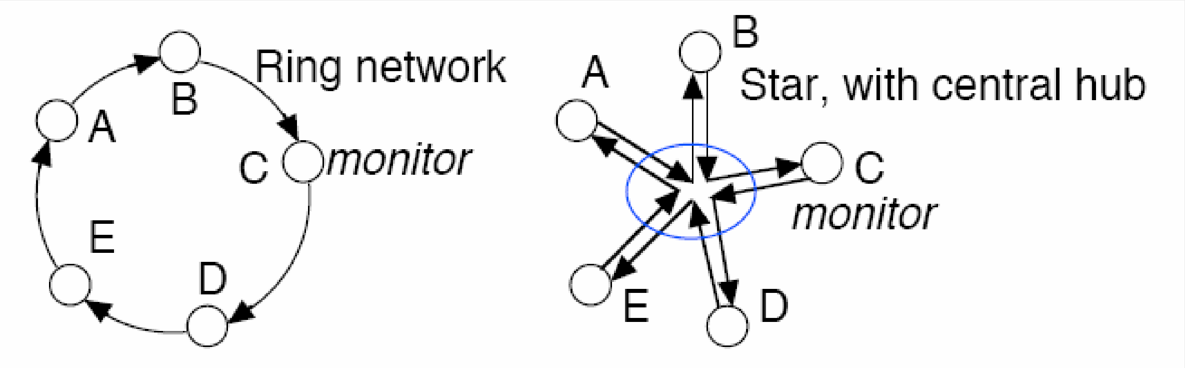
- Were popular during the 80's
- All stations are arranged in a ring
  - Each station is only connected to two of its neighbors
  - To send a message, it must pass through all of the stations in-between (clockwise, or anti-clockwise)
- Had better performance to early Ethernet implementations
- Supports speeds of 4Mbps, 16Mbps, 100Mbps, 1Gbps
- Declined in use, after the introduction of switched Ethernet

Token Networks

- Passes a *token* or right to transmit around the network in an orderly fashion. Only the station with the token can actually send data. Others must wait until they receive the token
- There are two approaches
  - A *token ring* has all the stations or nodes in a *physical ring*, so that each node is connected to only two neighbors. All communication requires intervening nodes to relay or forward traffic
  - A *token bus* physically resembles an Ethernet. Nodes form a *logical ring* based on their physical addresses. Most receive the token from the immediately higher address and send it to the immediately lower

Token Ring (IEEE 802.5)

- There are two physical configurations, the obvious physical ring, and a star with a central hub, which makes for easier control and maintenance

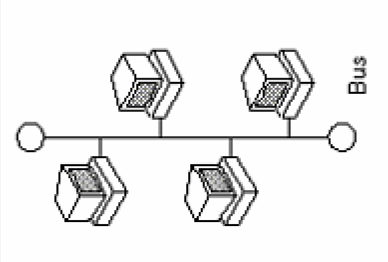


  - One node becomes the *active monitor,* which apart from its normal communication, checks that the ring is operating correctly and supplies the bit-clock for the ring. All other nodes are standby monitors
  - As each node receives from only its up-stream neighbor there is no need for a preamble to synchronize clocks (as with Ethernet

Star Topology

- Has one central hub, or computer, to which all others connect
- Easy to implement and extend, even in large networks
- Well suited for temporary networks (quick setup)
- Performance degrades as additional computers added
- Failure of central node can disable entire network
- Limited cable length and number of stations

Bus Topology



- Has a single communication line, typically a twisted pair, coaxial cable or optical fiber (medium is known as a *segment*)
- Is simple and is easy to add or remove nodes
- Only one station can send at a time, and under heavy traffic bottlenecks may occur

Ethernet

- Has a bus topology
- Uses a form of CSMA/CD contention protocol
- Concepts have been proposed as a standard
  - IEEE standard 802.3
- There a several variations on cable specifications depending on maximum number of stations that can connect and data rates

Cable Types for IEEE 802.3

- 10Base5 Ethernet
  - 50 ohm, 10mm coaxial cable
  - Maximum length of 500m
  - Known as ‘thick wire’
  - Relatively expensive and large diameter does not allow it to bend much
- 10Base2 Ethernet (thin wire)
  - 50 ohm, 5mm cable
  - Maximum length 200m
  - More flexible
  - Thinner wire allows it be connected directly to a computer using a T-connector
  - Thinner cable has more electronic resistance and cannot span as far
- **Note: In the notation XBaseY, X refers to the data rate, not cable width**

10Base-T Ethernet (twisted pair)

- All stations are connected to a central hub
- Essentially a star topology
  - But behaves like a bus topology
  - A station transmits to the hub, which regenerates the signal and broadcasts it to all other stations, until it reaches its destination
  - Known as ‘shared Ethernet'
- Collisions occur when the hub receives two transitions simultaneously
  - Then it broadcasts a special signal alerting the stations that a collision has occurred

10Base-T Ethernet (twisted pair)

- Configuration useful in office buildings
  - Allows networks to be implemented using fixed wiring

100Base-T Ethernet (‘fast Ethernet')

- 10Base-T is only 10Mbps
- In 1995 IEEE added 100BaseT to the 802.3 protocols
  - Similar to 10BaseT but uses higher quality cables
    - Uses a central hub
    - Uses twisted pair cable
  - Supports 100Mbps

Gigabit Ethernet (1000Base-x Ethernet)

- Uses 4x twisted pair cable
  - 1000Base-T
    - Maximum length is 100m
  - 1000Base-F
    - Offers the noise immunity of optical fiber
    - Maximum length is 2000m
- Identical protocols to 10Base-*x* Ethernets
- Can send a burst of packets, up to 8192 bytes

Recommended Reading

- Understanding Data Communications and Networks
  - Section 3.5 – Contention Protocols
  - Sections 6.1, 6.2 - LANs