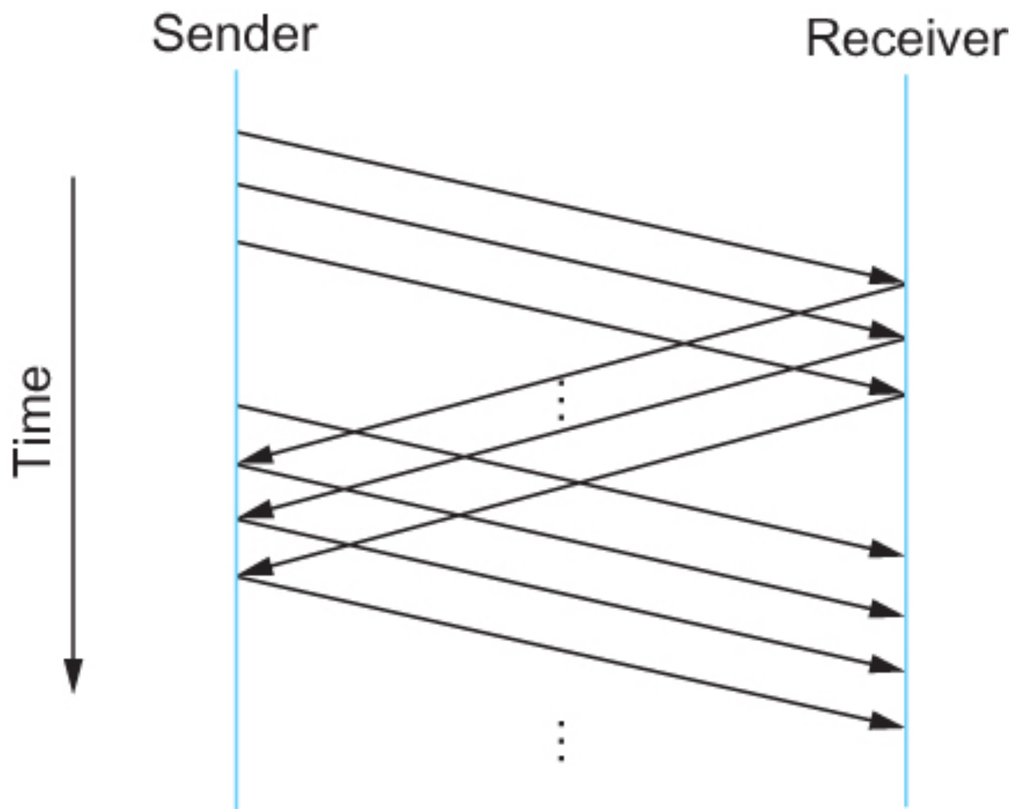
Networks and Distributed Systems

Lecture 5 – Sliding Window Protocol, Ethernet

Outline

- Perspectives on Connecting nodes
- Encoding
- Framing
- Error Detection
- Reliable Transmission
- Ethernet and Multiple Access Networks
- Wireless Networks

Sliding Window Protocol



Timeline for Sliding Window Protocol

Sliding Window Protocol

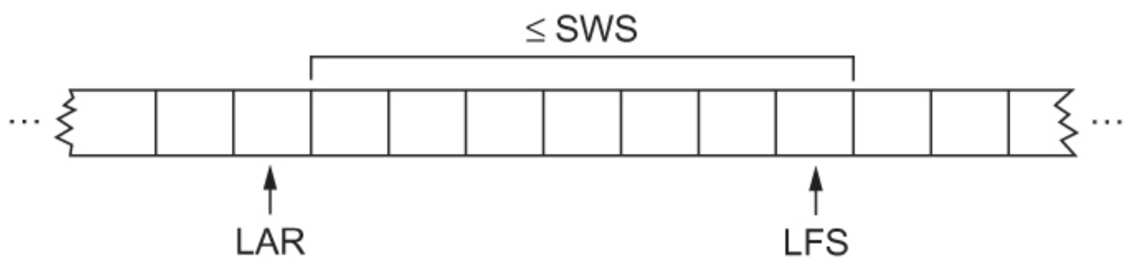
- Sender assigns a sequence number denoted as SeqNum to each frame.
  - Assume it can grow infinitely large

- Sender maintains three variables
  - Sending Window Size (SWS)
    - Upper bound on the number of outstanding (unacknowledged) frames that the sender can transmit
  - Last Acknowledgement Received (LAR)
    - Sequence number of the last acknowledgement received
  - Last Frame Sent (LFS)
    - Sequence number of the last frame sent

Sliding Window Protocol

- Sender also maintains the following invariant

  LFS – LAR ≤ SWS



Sliding Window on Sender

Sliding Window Protocol

- When an acknowledgement arrives
  - the sender moves LAR to right, thereby allowing the sender to transmit another frame
- Also the sender associates a timer with each frame it transmits
  - It retransmits the frame if the timer expires before the ACK is received
- Note that the sender has to be willing to buffer up to SWS frames
  - WHY?

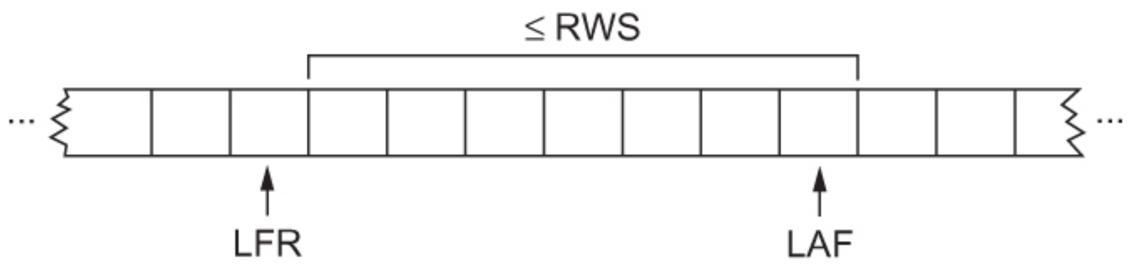
Sliding Window Protocol

- Receiver maintains three variables
  - Receiving Window Size (RWS)
    - Upper bound on the number of out-of-order frames that the receiver is willing to accept
  - Largest Acceptable Frame (LAF)
    - Sequence number of the largest acceptable frame
  - Last Frame Received (LFR)
    - Sequence number of the last frame received

Sliding Window Protocol

- Receiver also maintains the following invariant

$$LAF - LFR \leq RWS$$



Sliding Window on Receiver

Sliding Window Protocol

- When a frame with sequence number SeqNum arrives, what does the receiver do?
  - If SeqNum ≤ LFR or SeqNum > LAF
    - Discard it (the frame is outside the receiver window)
  - If LFR < SeqNum ≤ LAF
    - Accept it
    - Now the receiver needs to decide whether or not to send an ACK

Sliding Window Protocol

- Let SeqNumToAck
  - Denote the largest sequence number not yet acknowledged, such that all frames with sequence number less than or equal to SeqNumToAck have been received
- The receiver acknowledges the receipt of SeqNumToAck even if high-numbered packets have been received
  - This acknowledgement is said to be cumulative.
- The receiver then sets
  - LFR = SeqNumToAck and adjusts
  - LAF = LFR + RWS

Sliding Window Protocol

For example, suppose LFR = 5 and RWS = 4

(i.e. the last ACK that the receiver sent was for seq. no. 5)

⇒ LAF = 9

If frames 7 and 8 arrive, they will be buffered because they are within the receiver window

But no ACK will be sent since frame 6 is yet to arrive

Frames 7 and 8 are out of order

Frame 6 arrives (it is late because it was lost first time and had to be retransmitted)

Now Receiver Acknowledges Frame 8

and bumps LFR to 8

and LAF to 12

Issues with Sliding Window Protocol

- When timeout occurs, the amount of data in transit decreases
  - Since the sender is unable to advance its window
- When the packet loss occurs, this scheme is no longer keeping the pipe full
  - The longer it takes to notice that a packet loss has occurred, the more severe the problem becomes
- How to improve this
  - Negative Acknowledgement (NAK)
  - Additional Acknowledgement
  - Selective Acknowledgement

Issues with Sliding Window Protocol

- Negative Acknowledgement (NAK)
  - Receiver sends NAK for frame 6 when frame 7 arrive (in the previous example)
    - However this is unnecessary since sender's timeout mechanism will be sufficient to catch the situation
- Additional Acknowledgement
  - Receiver sends additional ACK for frame 5 when frame 7 arrives
    - Sender uses duplicate ACK as a clue for frame loss
- Selective Acknowledgement
  - Receiver will acknowledge exactly those frames it has received, rather than the highest number frames
    - Receiver will acknowledge frames 7 and 8
    - Sender knows frame 6 is lost
    - Sender can keep the pipe full (additional complexity)

Issues with Sliding Window Protocol

How to select the window size

- SWS is easy to compute
  - Delay × Bandwidth
- RWS can be anything
  - Two common setting
    - RWS = 1

      No buffer at the receiver for frames that arrive out of order
    - RWS = SWS

      The receiver can buffer frames that the sender transmits

It does not make any sense to keep RWS > SWS

WHY?

Issues with Sliding Window Protocol

- Finite Sequence Number
  - Frame sequence number is specified in the header field
    - Finite size
      - 3 bit: eight possible sequence number: 0, 1, 2, 3, 4, 5, 6, 7
    - It is necessary to wrap around

Issues with Sliding Window Protocol

- How to distinguish between different incarnations of the same sequence number?
  - Number of possible sequence number must be larger than the number of outstanding frames allowed
    - Stop and Wait: One outstanding frame
      - 2 distinct sequence number (0 and 1)
    - Let MaxSeqNum be the number of available sequence numbers
    - SWS + 1 ≤ MaxSeqNum
      - Is this sufficient?

Issues with Sliding Window Protocol

SWS + 1 ≤ MaxSeqNum

- Is this sufficient?

- Depends on RWS
- If RWS = 1, then sufficient
- If RWS = SWS, then not good enough

- For example, we have eight sequence numbers

  0, 1, 2, 3, 4, 5, 6, 7

  RWS = SWS = 7

  Sender sends 0, 1, …, 6

  Receiver receives 0, 1, … ,6

  Receiver acknowledges 0, 1, …, 6

  ACK (0, 1, …, 6) are lost

  Sender retransmits 0, 1, …, 6

  Receiver is expecting 7, 0, …., 5

Issues with Sliding Window Protocol

To avoid this,

If RWS = SWS

SWS < (MaxSeqNum + 1)/2

Issues with Sliding Window Protocol

- Serves three different roles
  - Reliable
  - Preserve the order
    - Each frame has a sequence number
    - The receiver makes sure that it does not pass a frame up to the next higher-level protocol until it has already passed up all frames with a smaller sequence number
  - Frame control
    - Receiver is able to throttle the sender
      - Keeps the sender from overrunning the receiver
        - From transmitting more data than the receiver is able to process

Ethernet

- Most successful local area networking technology of last 20 years.
- Developed in the mid-1970s by researchers at the Xerox Palo Alto Research Centers (PARC).
- Uses CSMA/CD technology
    - Carrier Sense Multiple Access with Collision Detection.
    - A set of nodes send and receive frames over a shared link.
    - Carrier sense means that all nodes can distinguish between an idle and a busy link.
    - Collision detection means that a node listens as it transmits and can therefore detect when a frame it is transmitting has collided with a frame transmitted by another node.

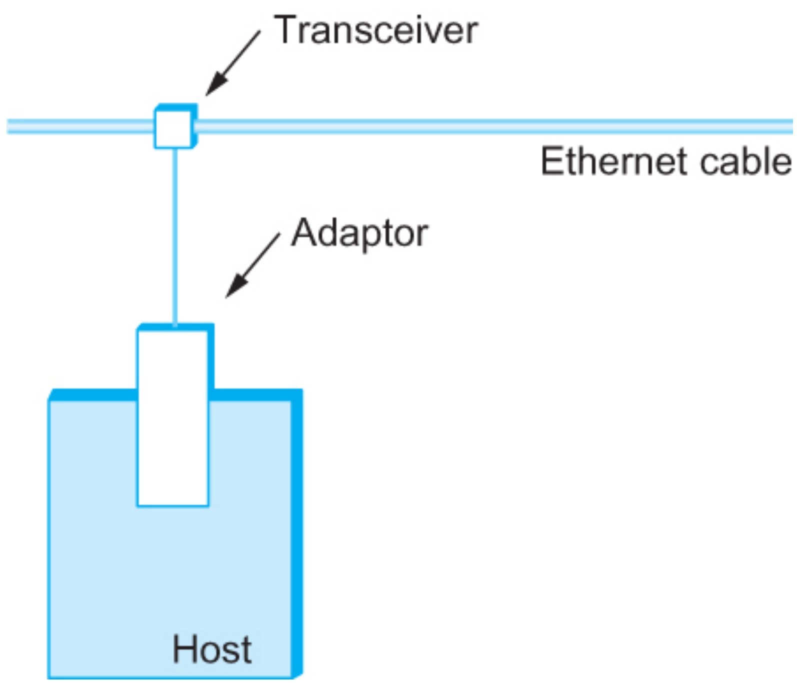
Ethernet

- Uses ALOHA (packet radio network) as the root protocol
  - Developed at the University of Hawaii to support communication across the Hawaiian Islands.
  - For ALOHA the medium was atmosphere, for Ethernet the medium is a coax cable.
- DEC and Intel joined Xerox to define a 10-Mbps Ethernet standard in 1978.
- This standard formed the basis for IEEE standard 802.3
- More recently 802.3 has been extended to include a 100-Mbps version called Fast Ethernet and a 1000-Mbps version called Gigabit Ethernet.

Ethernet

- An Ethernet segment is implemented on a coaxial cable of up to 500 m.
  - This cable is similar to the type used for cable TV except that it typically has an impedance of 50 ohms instead of cable TV’s 75 ohms.
- Hosts connect to an Ethernet segment by tapping into it.
- A transceiver (a small device directly attached to the tap) detects when the line is idle and drives signal when the host is transmitting.
- The transceiver also receives incoming signal.
- The transceiver is connected to an Ethernet adaptor which is plugged into the host.
- The protocol is implemented on the adaptor.

Ethernet

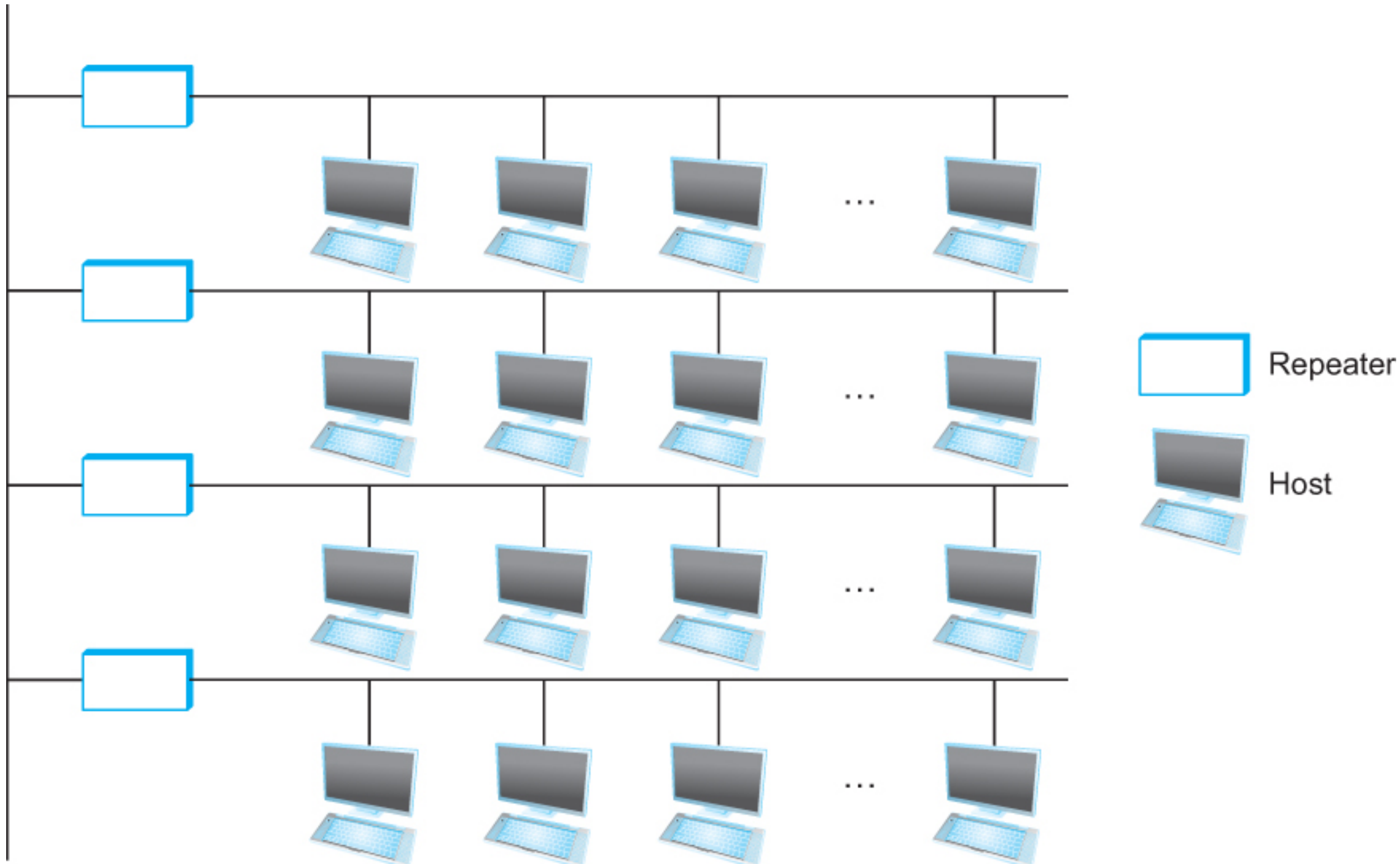


Ethernet transceiver and adaptor

Ethernet

- Multiple Ethernet segments can be joined together by *repeaters.*
- A *repeater* is a device that forwards digital signals.
- No more than four repeaters may be positioned between any pair of hosts.
  - An Ethernet has a total reach of only 2500 m.

Ethernet



Ethernet repeater

Ethernet

- Any signal placed on the Ethernet by a host is broadcast over the entire network
  - Signal is propagated in both directions.
  - Repeaters forward the signal on all outgoing segments.
  - Terminators attached to the end of each segment absorb the signal.
- Ethernet uses Manchester encoding scheme.

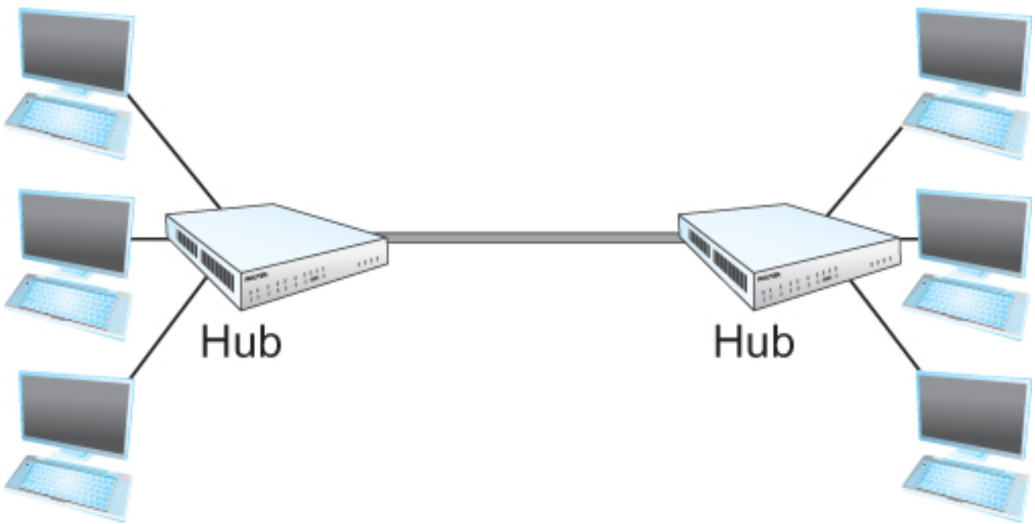
Ethernet

- New Technologies in Ethernet
  - Instead of using coax cable, an Ethernet can be constructed from a thinner cable known as 10Base2 (the original was 10Base5)
    - 10 means the network operates at 10 Mbps
    - Base means the cable is used in a baseband system
    - 2 means that a given segment can be no longer than 200 m

Ethernet

- New Technologies in Ethernet
  - Another cable technology is 10BaseT
    - T stands for twisted pair
    - Limited to 100 m in length
  - With 10BaseT, the common configuration is to have several point to point segments coming out of a multiway repeater, called *Hub*

Ethernet

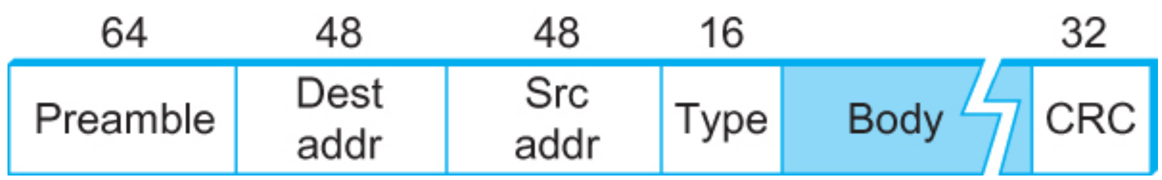


Ethernet Hub

Access Protocol for Ethernet

- The algorithm is commonly called Ethernet's Media Access Control (MAC).
  - It is implemented in Hardware on the network adaptor.
- Frame format
  - Preamble (64bit): allows the receiver to synchronize with the signal (sequence of alternating 0s and 1s).
  - Host and Destination Address (48bit each).
  - Packet type (16bit): acts as demux key to identify the higher level protocol.
  - Data (up to 1500 bytes)
    - Minimally a frame must contain at least 46 bytes of data.
    - Frame must be long enough to detect collision.
  - CRC (32bit)

Ethernet Frame



Ethernet Frame Format

Ethernet Addresses

- Each host on an Ethernet (in fact, every Ethernet host in the world) has a unique Ethernet Address.
- The address belongs to the adaptor, not the host.
  - It is usually burnt into ROM.
- Ethernet addresses are typically printed in a human readable format
  - As a sequence of six numbers separated by colons.
  - Each number corresponds to 1 byte of the 6 byte address and is given by a pair of hexadecimal digits, one for each of the 4-bit nibbles in the byte
  - Leading 0s are dropped.
  - For example, 8:0:2b:e4:b1:2 is
    - 00001000 00000000 00101011 11100100 10110001 00000010

Ethernet Addresses

- To ensure that every adaptor gets a unique address, each manufacturer of Ethernet devices is allocated a different prefix that must be prepended to the address on every adaptor they build
  - AMD has been assigned the 24bit prefix 8:0:20

Ethernet Addresses

- Each frame transmitted on an Ethernet is received by every adaptor connected to that Ethernet.
- Each adaptor recognizes those frames addressed to its address and passes only those frames on to the host.
- In addition, to *unicast* address, an Ethernet address consisting of all 1s is treated as a *broadcast* address.
    - All adaptors pass frames addressed to the *broadcast* address up to the host.
- Similarly, an address that has the first bit set to 1 but is not the *broadcast* address is called a *multicast* address.
    - A given host can program its adaptor to accept some set of *multicast* addresses.

Ethernet Addresses

- To summarize, an Ethernet adaptor receives all frames and accepts
  - Frames addressed to its own address
  - Frames addressed to the broadcast address
  - Frames addressed to a multicast addressed if it has been instructed

Ethernet Transmitter Algorithm

- When the adaptor has a frame to send and the line is idle, it transmits the frame immediately.
  - The upper bound of 1500 bytes in the message means that the adaptor can occupy the line for a fixed length of time.
- When the adaptor has a frame to send and the line is busy, it waits for the line to go idle and then transmits immediately.
- The Ethernet is said to be 1-persistent protocol because an adaptor with a frame to send transmits with probability 1 whenever a busy line goes idle.

Ethernet Transmitter Algorithm

- Since there is no centralized control it is possible for two (or more) adaptors to begin transmitting at the same time,
  - Either because both found the line to be idle,
  - Or, both had been waiting for a busy line to become idle.
- When this happens, the two (or more) frames are said to be *collide* on the network.

Ethernet Transmitter Algorithm

- Since Ethernet supports collision detection, each sender is able to determine that a collision is in progress.
- At the moment an adaptor detects that its frame is colliding with another, it first makes sure to transmit a 32-bit jamming sequence and then stops transmission.
  - Thus, a transmitter will minimally send 96 bits in the case of collision
    - 64-bit preamble + 32-bit jamming sequence

Ethernet Transmitter Algorithm

- One way that an adaptor will send only 96 bit (called a *runt frame*) is if the two hosts are close to each other.
- Had they been farther apart,
  - They would have had to transmit longer, and thus send more bits, before detecting the collision.

Ethernet Transmitter Algorithm

- The worst case scenario happens when the two hosts are at opposite ends of the Ethernet.
- To know for sure that the frame its just sent did not collide with another frame, the transmitter may need to send as many as 512 bits.
  - Every Ethernet frame must be at least 512 bits (64 bytes) long.
    - 14 bytes of header + 46 bytes of data + 4 bytes of CRC

Ethernet Transmitter Algorithm

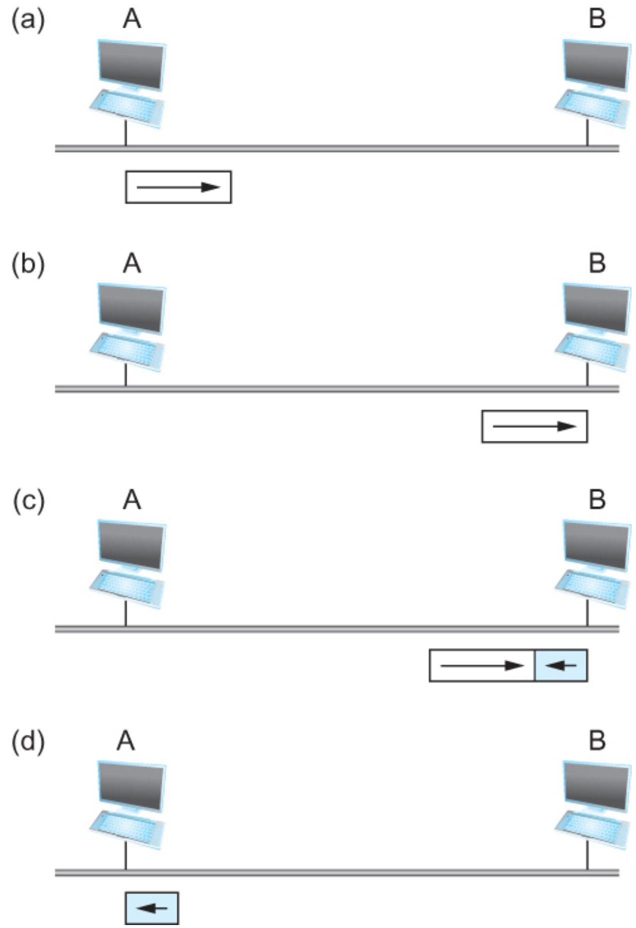
- Why 512 bits?
  - Why is its length limited to 2500 m?

- The farther apart two nodes are, the longer it takes for a frame sent by one to reach the other, and the network is vulnerable to collision during this time

Ethernet Transmitter Algorithm

- A begins transmitting a frame at time *t*
- *d* denotes the one link latency
- The first bit of A's frame arrives at B at time *t* + *d*
- Suppose an instant before host A's frame arrives, host B begins to transmit its own frame
- B's frame will immediately collide with A's frame and this collision will be detected by host B
- Host B will send the 32-bit jamming sequence
- Host A will not know that the collision occurred until B's frame reaches it, which will happen at *t* + 2 * *d*
- Host A must continue to transmit until this time in order to detect the collision
  - Host A must transmit for 2 * *d* to be sure that it detects all possible collisions

Ethernet Transmitter Algorithm



Worst-case scenario: (a) A sends a frame at time *t; (b) A's frame arrives* at B at time *t + d; (c) B begins transmitting at time t + d and collides with A's frame;* (d) B's runt (32-bit) frame arrives at A at time *t + 2d.*

Ethernet Transmitter Algorithm

- Consider that a maximally configured Ethernet is 2500 m long, and there may be up to four repeaters between any two hosts, the round trip delay has been determined to be 51.2 μs
  - Which on 10 Mbps Ethernet corresponds to 512 bits
- The other way to look at this situation,
  - We need to limit the Ethernet's maximum latency to a fairly small value (51.2 μs) for the access algorithm to work
    - Hence the maximum length for the Ethernet is on the order of 2500 m.

Ethernet Transmitter Algorithm

- Once an adaptor has detected a collision, and stopped its transmission, it waits a certain amount of time and tries again.
- Each time the adaptor tries to transmit but fails, it doubles the amount of time it waits before trying again.
- This strategy of doubling the delay interval between each retransmission attempt is known as *Exponential Backoff*.

Ethernet Transmitter Algorithm

- The adaptor first delays either 0 or 51.2 μs, selected at random.
- If this effort fails, it then waits 0, 51.2, 102.4, 153.6 μs (selected randomly) before trying again;
  - This is k * 51.2 for k = 0, 1, 2, 3
- After the third collision, it waits k * 51.2 for $k = 0\ldots2^3 - 1$ (again selected at random).
- In general, the algorithm randomly selects a k between 0 and $2^n - 1$ and waits for k * 51.2 μs, where n is the number of collisions experienced so far.

Experience with Ethernet

- Ethernets work best under lightly loaded conditions.
  - Under heavy loads, too much of the network's capacity is wasted by collisions.
- Most Ethernets are used in a conservative way.
  - Have fewer than 200 hosts connected to them which is far fewer than the maximum of 1024.
- Most Ethernets are far shorter than 2500m with a round-trip delay of closer to 5 μs than 51.2 μs.
- Ethernets are easy to administer and maintain.
  - There are no switches that can fail and no routing and configuration tables that have to be kept up-to-date.
  - It is easy to add a new host to the network.
  - It is inexpensive.
    - Cable is cheap, and only other cost is the network adaptor on each host.