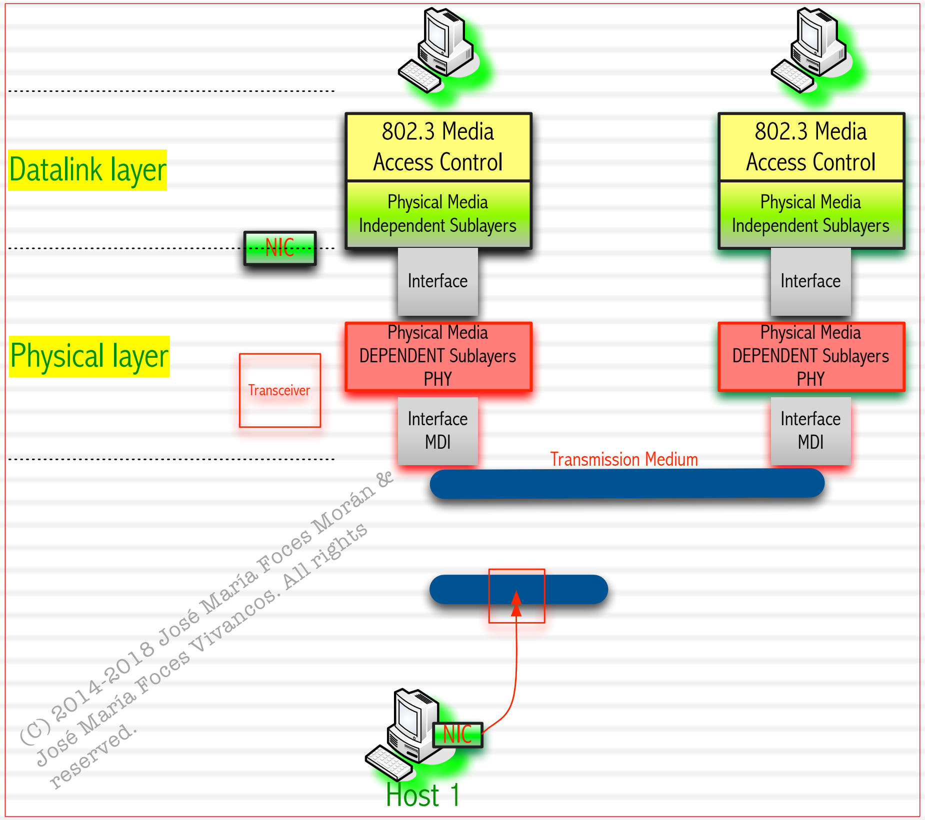
1

CSMA/CD Ethernet

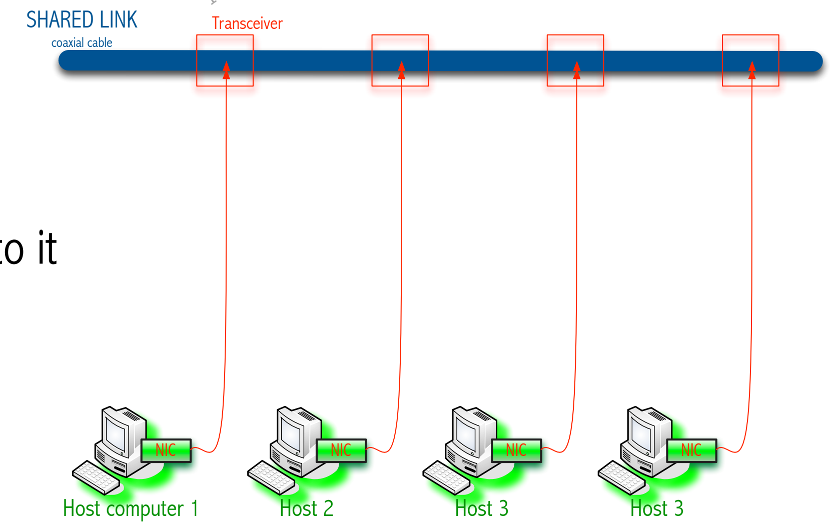


Ethernet

Uses **CSMA/CD** MAC (Media Access Control)

- CS: Carrier Sense

  A computer can distinguish when the link is being used and when it is not



- MA: Multiple Access

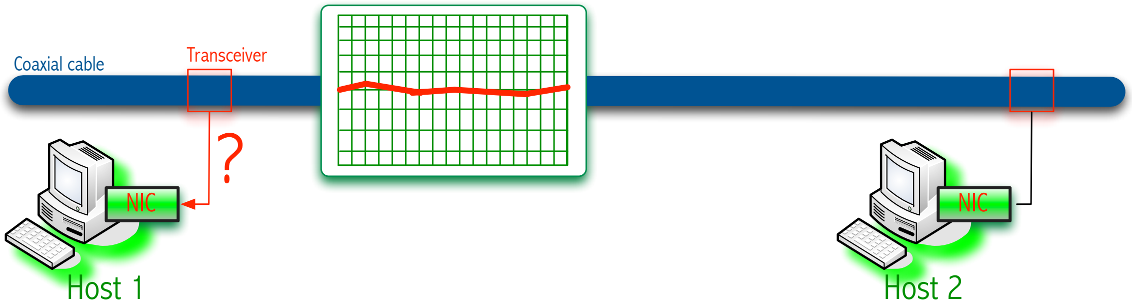
  The link is shared among all the computers connected to it

- CD: Collision Detection

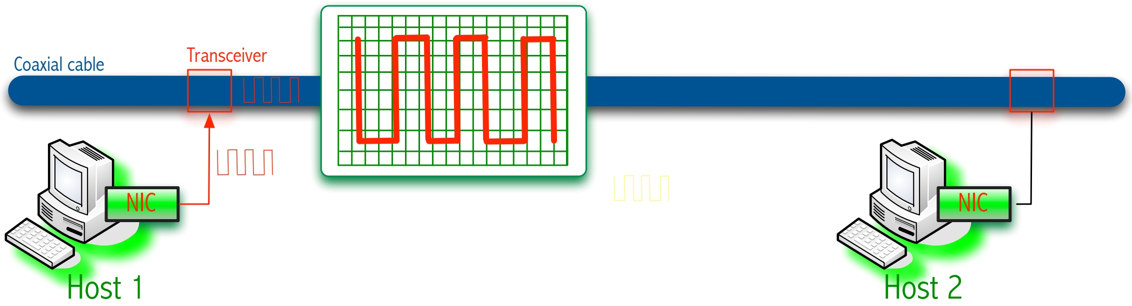
  A computer listens as it transmits, can detect when its transmission has collided with another frame being transmitted by another node

Ethernet Transmitter Algorithm: No carrier

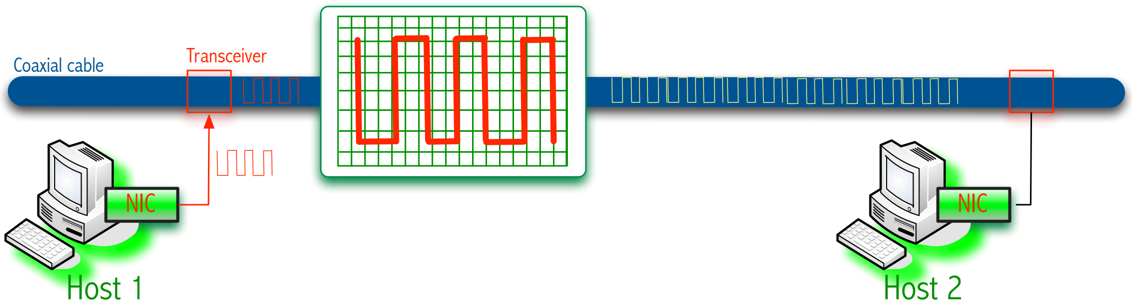
- NIC has a new frame to send
- Line is idle (No carrier)
  - Decides to transmit it *almost* immediately
  - Waits IFG seconds before starting transmission
  - IFG: Inter Frame Gap = 9,6 μs



1. Host 1 senses the medium to establish whether or not CARRIER is present
2. Concludes that there is not CARRIER



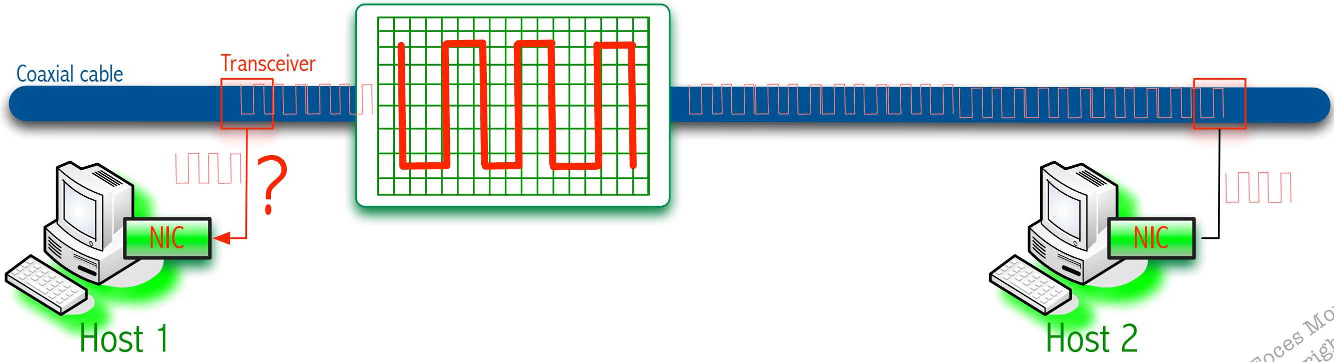
3. Host 1 begins to transmit a new Ethernet frame



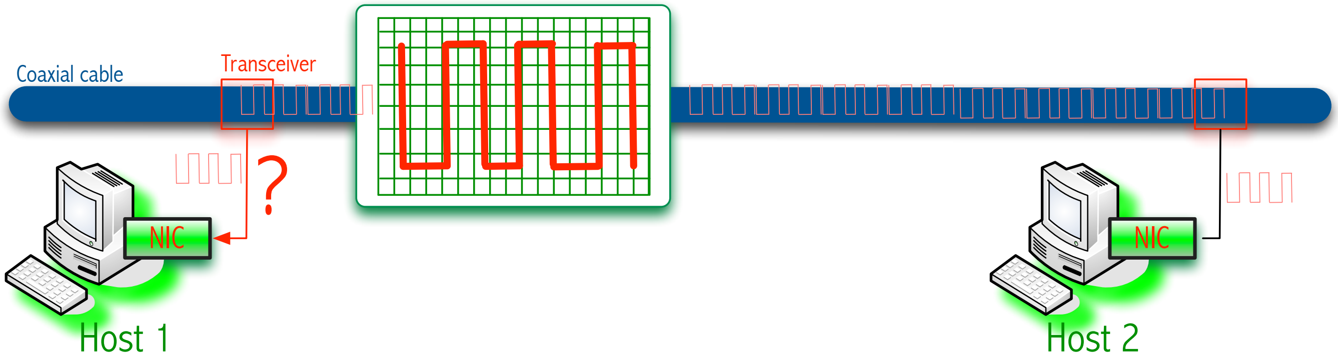
4. Host 1 continues transmiting the whole Ethernet frame, signal propagates

Ethernet Transmitter Algorithm: Carrier found

- **Ethernet** is said to be 1-persistent protocol because an adaptor with a frame to send attempts to transmit it with probability 1after waiting an IFG time after it sees that the medium went idle



1. Host 2 is transmitting a frame and, so far, it propagated through host 1
2. Host 1 senses the medium to establish whether or not CARRIER is present
3. Concludes that there is CARRIER

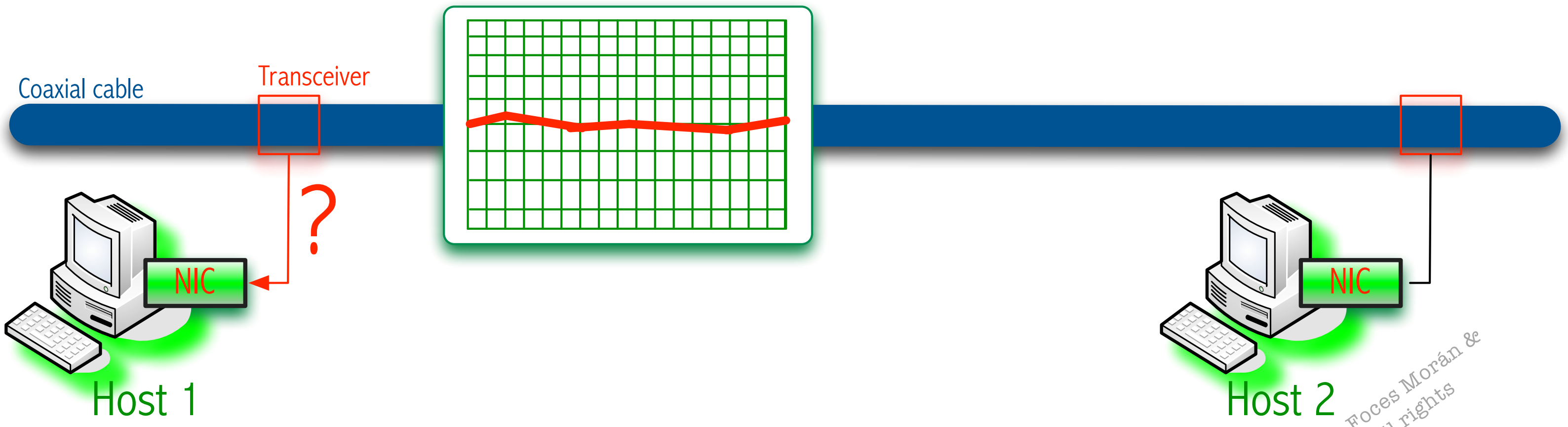


4. Host 1 defers transmission until the medium is idle (no carrier) + an IFG time (InterFrame Gap = 9,6 μs)

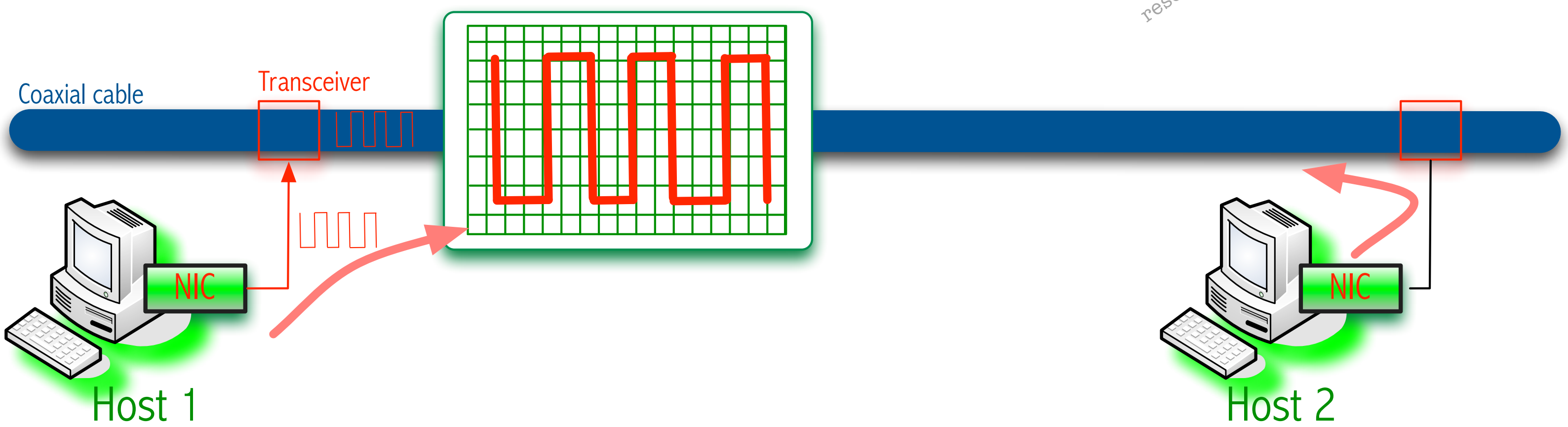
Ethernet Transmitter Algorithm: Collisions

- Since there is **no centralized control** it is possible for two adaptors to
  - Begin transmitting at the same time, is it possible? Yes, it is!
    - Either because both found the line to be idle at their physical positions along the cable
    - Or, both had been waiting for a busy line to become idle after an IFG time (9,6 µs)
- When this happens, the two frames are said to *collide* on the network
  - A collision has occurred
  - *Collisions may involve any number of nodes (>1, obviously)*

Ethernet Transmitter Algorithm: Collisions



1. Host 1 senses the medium to establish whether or not CARRIER is present
2. Concludes that there is not CARRIER

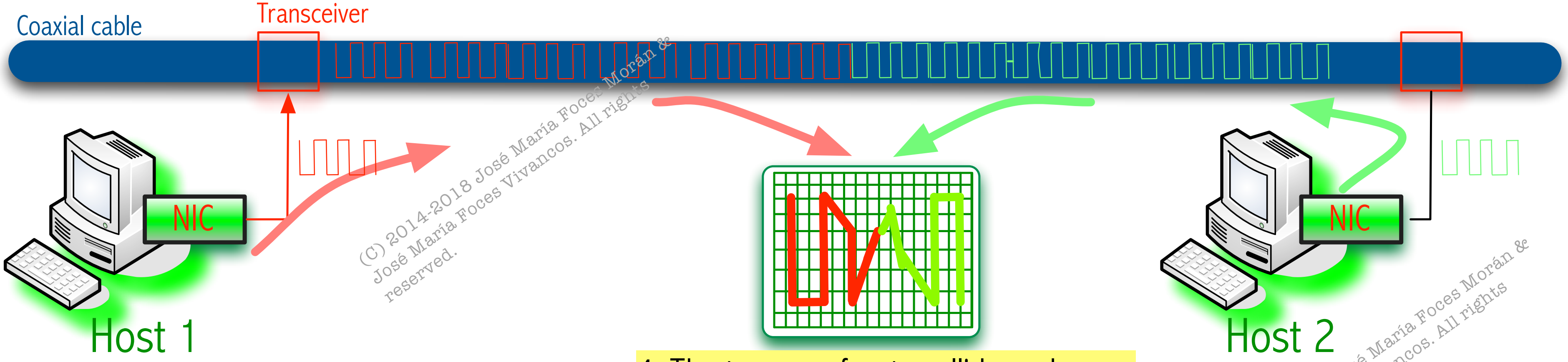


3. Host 1 begins to transmit a new Ethernet frame

4. Host 2 sees no CARRIER either and begins to transmit its new Ethernet frame

Ethernet Transmitter Algorithm: Collisions

7



4. The two wavefronts collide and the collided signal begins to propagate to the right and to the left

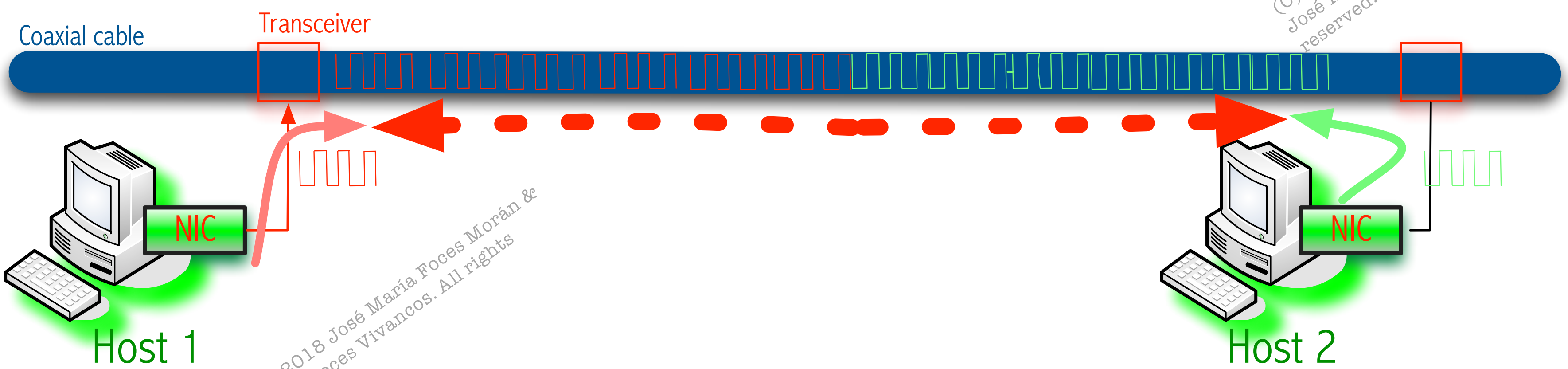
4'. Host 1 continues transmitting the whole Ethernet frame, bit by bit, each time it transmits a new bit, it reads it to make sure it is correct and to make sure that no collision has occured. From host 1 standpoint, the collision has not occurred yet

4'. Host 2: same as Host 1, keeps transmitting unaware yet that a collision has occurred

5. Collided data signal keeps propagating as the hosts keep transmitting

Ethernet Transmitter Algorithm: Collisions

- Since Ethernet supports collision detection (CD), each sender is able to determine that a collision is in progress.
- 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



6. When the collided signal arrives at host 2, for example, it interferes with the data transmitted and the host notices this because the data read is not coincident with the data written anymore.

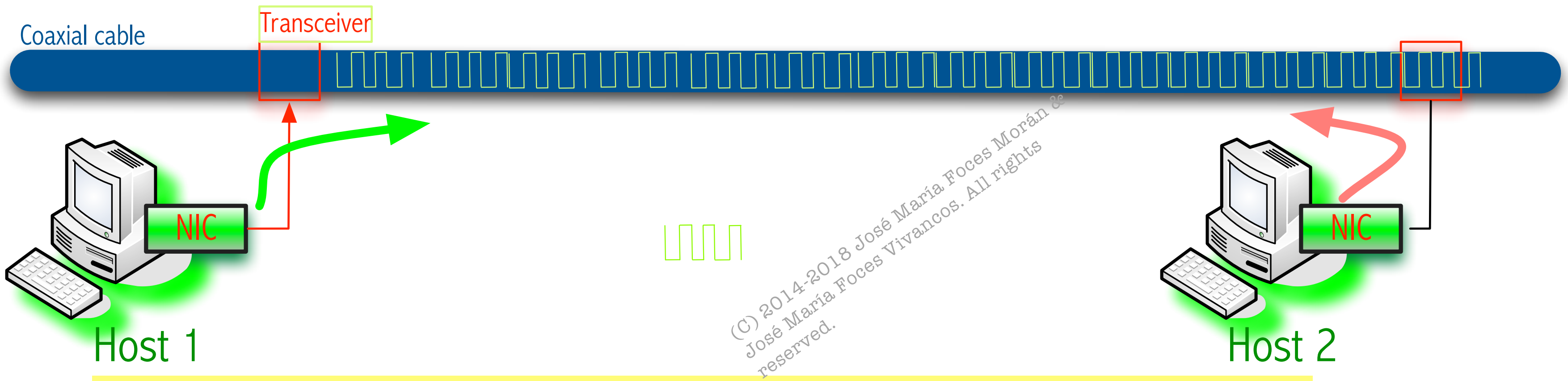
7. Host 2 sends a 32-bit jamming sequence so that the rest of nodes are informed that a collision occurred.

Ethernet Transmitter Algorithm: Collisions

- 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: Collisions

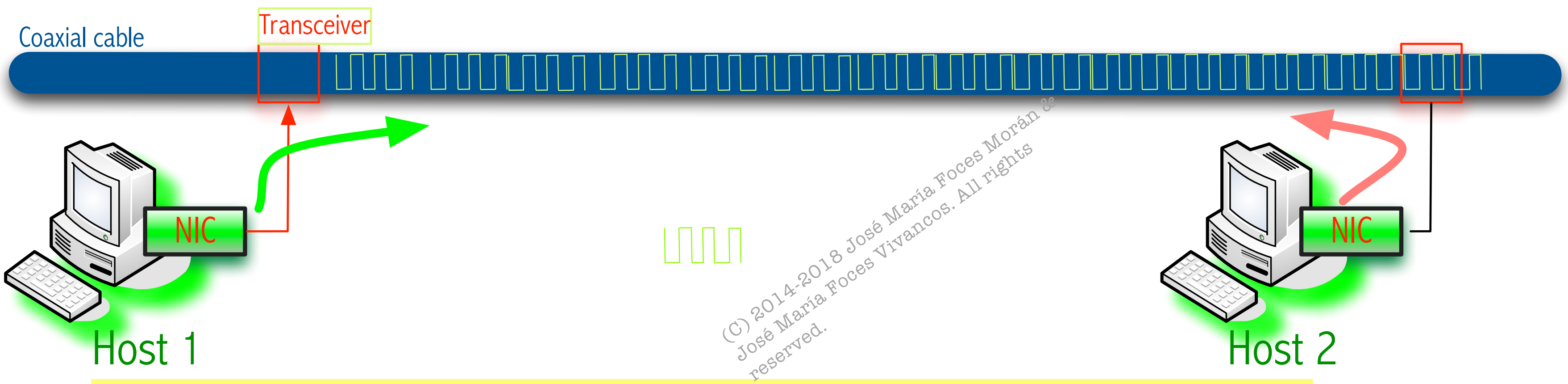
- The worst case scenario happens when the two hosts are at opposite ends of the Ethernet
- To know for sure that the frame it has just sent did not collide with another frame, the transmitter has to send at least 512 bits
  - CONCLUSION: Ethernet frames must be at least 512 bits (64 bytes) long
    - 14 bytes of header + 46 bytes of data + 4 bytes of CRC
    - If the application is sending less than 46 bytes of data, the transmitter will include padding 0's



Host 1 must transmit at least 512 bits to make sure that it itself detects a collision even with the host located at the farthest end of the network, otherwise it will miss that a collision occurred

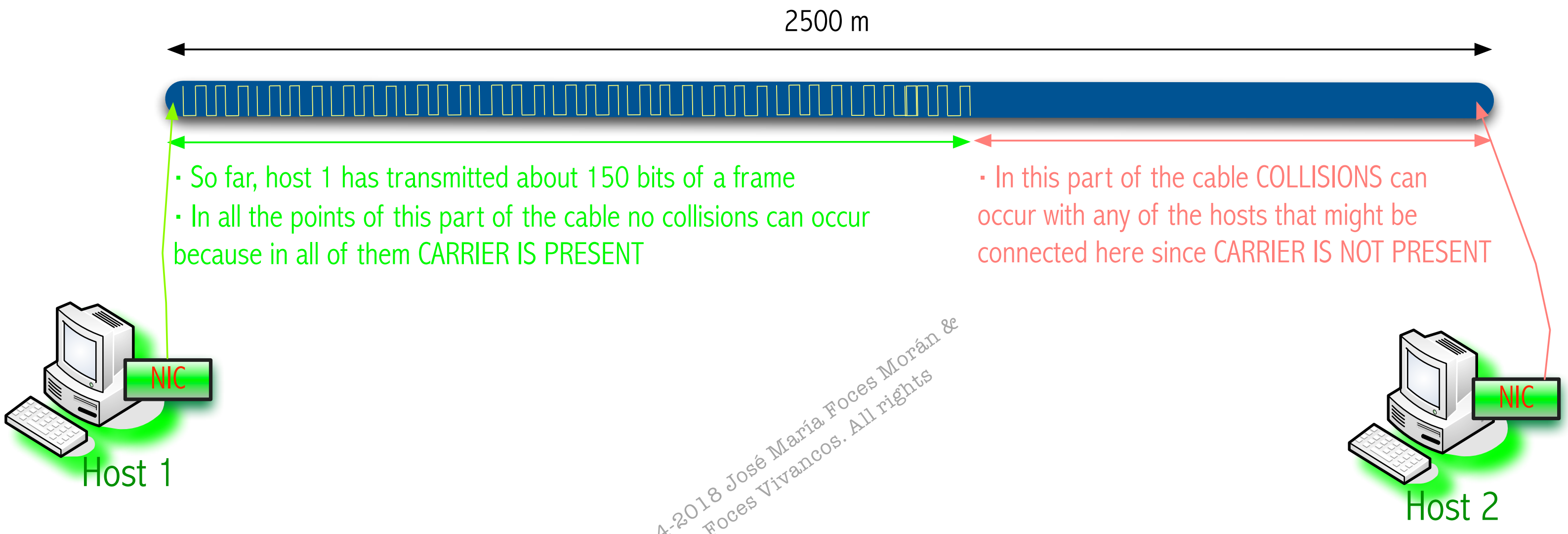
Ethernet Transmitter Algorithm: Collisions

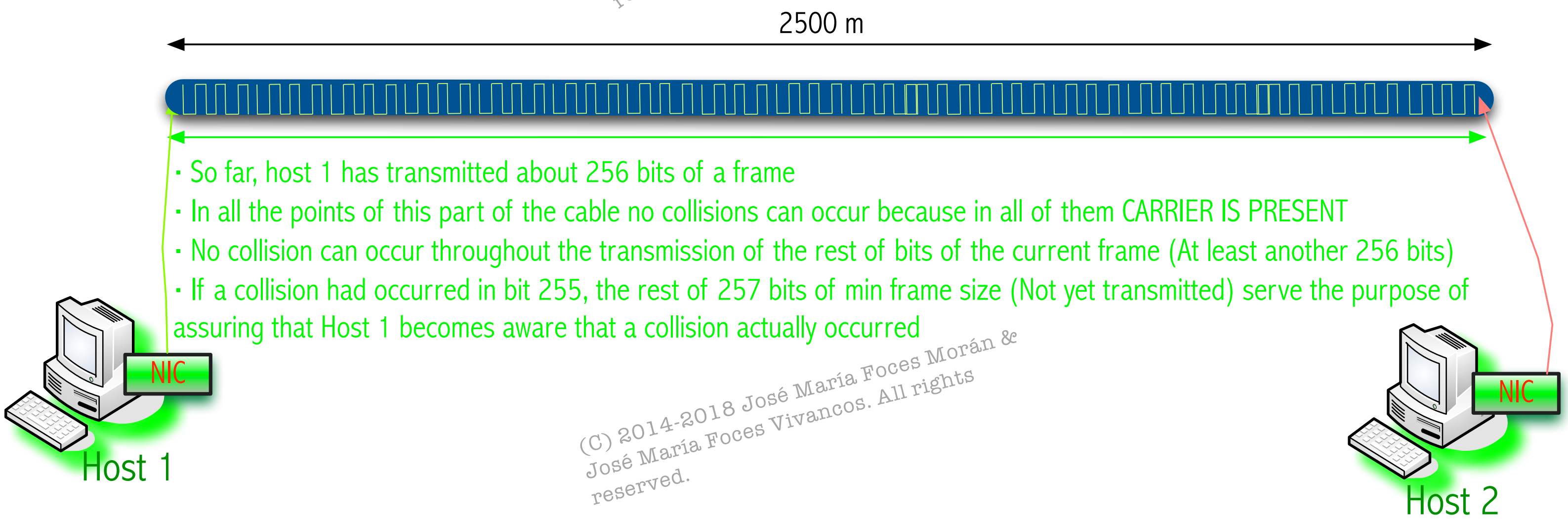
- Why is 512 bits Ethernet's minimum frame size?
  - Why is its maximum length limited to 2500 m?
  - Recall speed is 10Mbps and RTT=51,2 µs, therefore $V_{prop} \cong 2/3 \cdot c$
- Only the first 256 bits (maximum) of the 512 are vulnerable to collisions with any other ethernet host connected to the cable
- In the worst case, the whole 512 bits are necessary for the transmitting host to become aware that a collision took place –with any of the first 256 bits



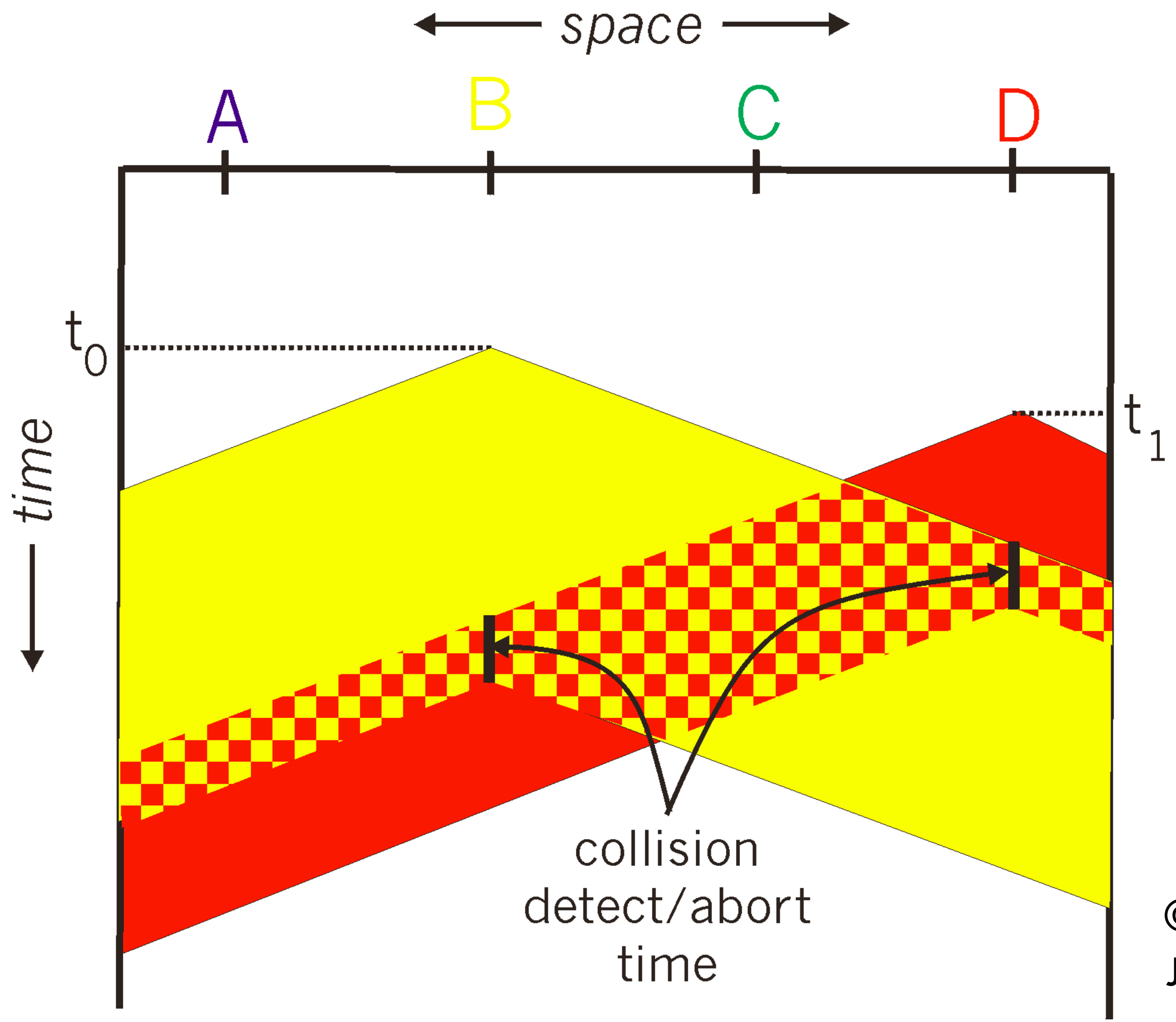
Host 1 must transmit at least 512 bits to make sure that it itself detects a collision even with the host located at the farthest end of the network, otherwise it will miss that a collision occurred

Ethernet Transmitter Algorithm: Collisions



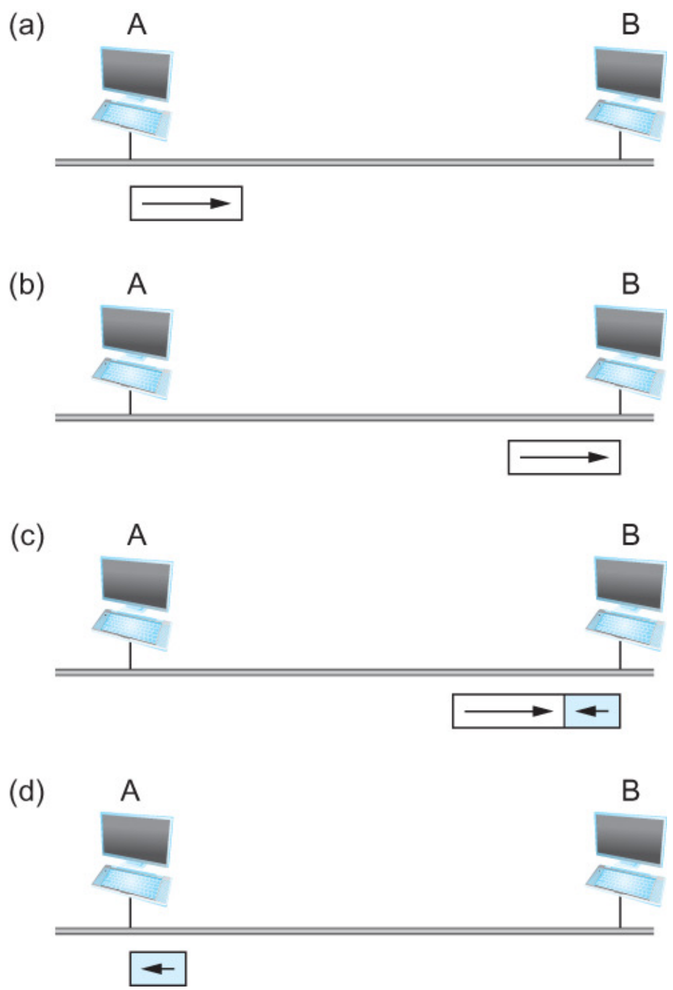


Ethernet Transmitter Algorithm: Collisions



Ethernet Transmitter Algorithm: Collisions

- A begins transmitting a frame at time t
- d denotes the one link latency. RTT = 51,2 μs, d = RTT/2
- 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 $2 * d = RTT$ to be sure that it detects all possible collisions



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 Rtt = 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 RTT to a fairly small value (51.2 μs) for the access algorithm to work fine
    - 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***

Exponential Backoff: Defer next transmission attempt after a collision

- When two hosts have collided, how can we avoid further collisions to occur which involve those two hosts?
- We'll have both hosts choose a random number to determine how much time they will defer the new transmission attempt, but, what algorithm can be applied? Ethernet uses *Exponential Backoff Algorithm*
- Assume k is the number of collisions undergone by a host adapter when attempting to transmit a specific frame
  - k = 1 The *adaptor flips 1 coin* and gets heads (0) or tails (1) = {0, 1}. Name the result r, then:
    - Time to defer transmission= r * 51,2 µs
    - 0 -> 0 µs
    - 1 -> 51,2 µs
  - k = 2 The *adaptor flips 2 coins* and gets one of {00, 01, 10, 11} = {0, 1, 2, 3}
    - Time to defer transmission = r * 51,2 µs
    - 0 -> 0 µs
    - 1 -> 1 x 51,2 µs
    - 2 -> 2 x 51,2 µs
    - 3 -> 3 x 51,2 µs
  - In general, the transmitter will flip k coins, thereby obtaining one of {0, 1, 2, … 2^{k-1}} and calculating the time to defer next transmission attempt as r x 51,2 µs
- Flipping the coin, in a computer system, consists of generating random numbers

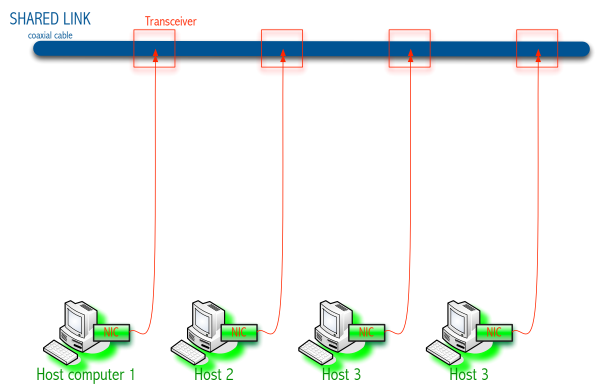
Experience with Ethernet

- 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.

- 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.

Relevant discrete probability distributions

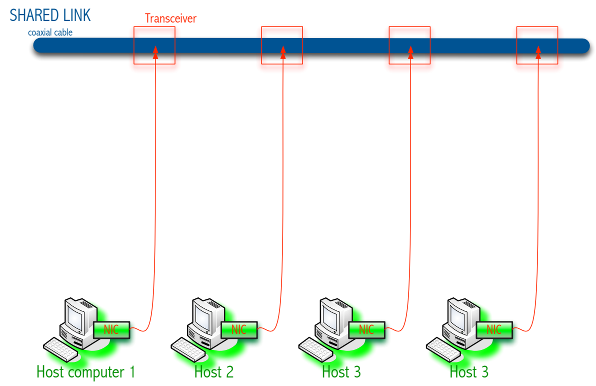
- Bernouilli's, outcomes {0, 1}
  - p(1) = a
  - p(0) = 1 – a
    - What's the average probability that host 1 is transmitting at a given instant of time?
    - In other words, what's the network utilization achieved by any individual host?



- Binomial
  - Assume that the average p(1) to all hosts on the bus is known, then what's the average probability that any number of hosts attempt transmission at the same time?
  - Otherwise, the experiment about "transmit/not-transmit" is repeated n times, then, what's the probability that I of them want to transmit?
  - What's the probability of a collision involving i hosts out of n

$$p(i) = \binom{n}{i} a^i (1-a)^{n-i}$$

Example on probability of collision

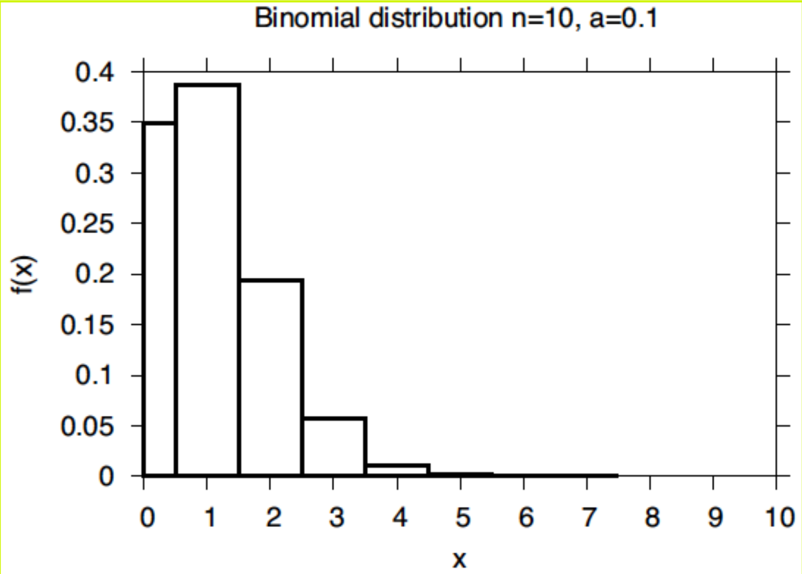


- A shared bus Ethernet is comprised of ten stations (Hosts). The utilization achieved by the stations is about p(1) = 10% or 0,1.
  - What's the probability that one station is transmitting?
  - What's the probability that five stations are transmitting?
  - What's the probability that all of the stations are transmitting?

$$p(1) = \binom{10}{1} 0.1^1 0.9^9 = 0.38$$

$$p(5) = \binom{10}{5} 0.1^5 0.9^5 = 1.49 \times 10^{-3}$$

$$p(10) = \binom{10}{10} 0.1^{10} 0.9^0 = 1 \times 10^{-10}$$



$$p(i) = \binom{n}{i} a^i (1-a)^{n-i}$$