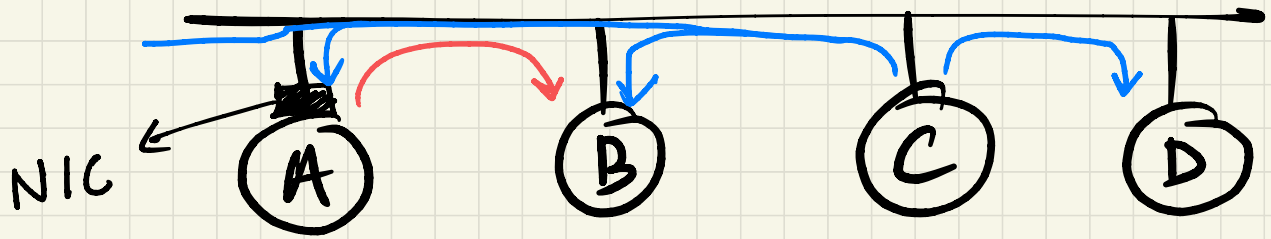
Carrier Sense Multiple Access (CSMA)

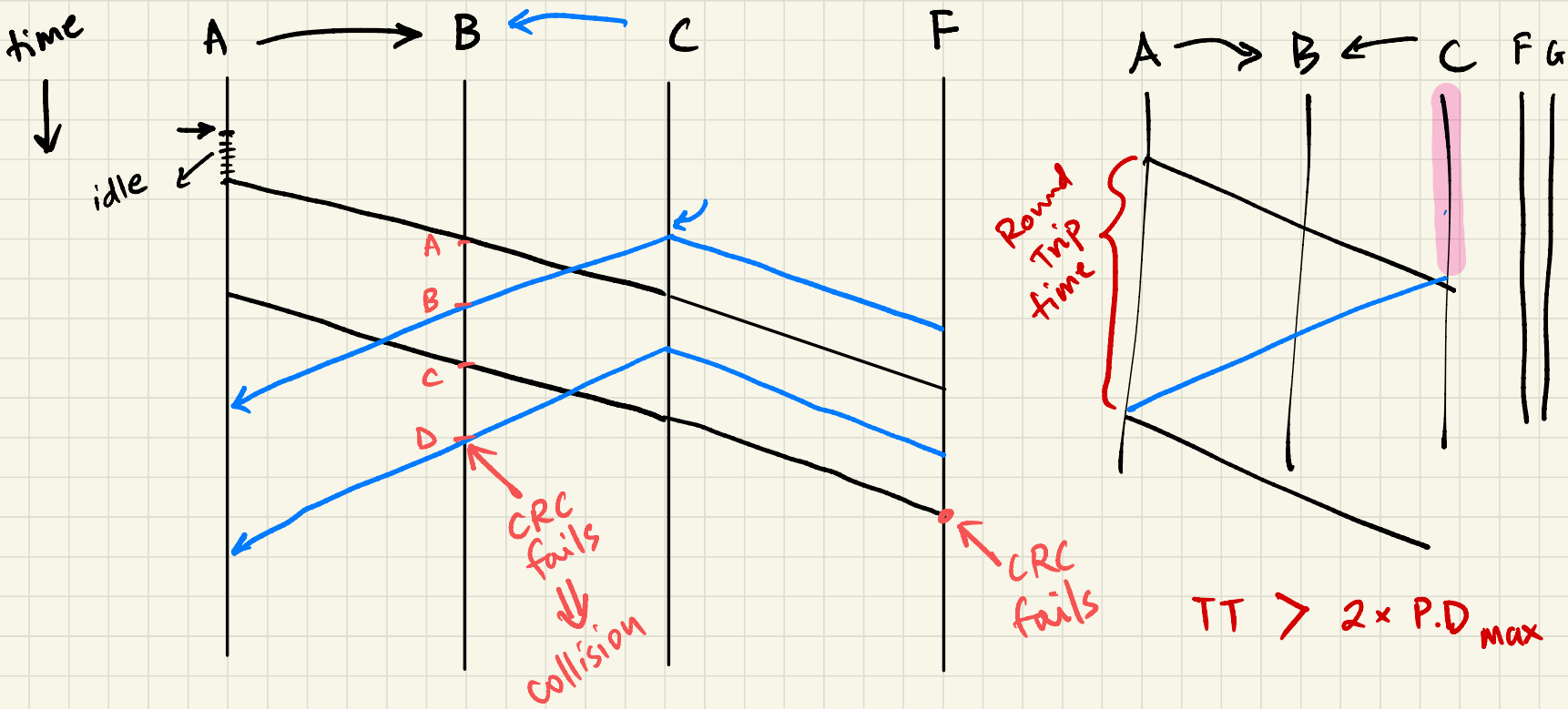


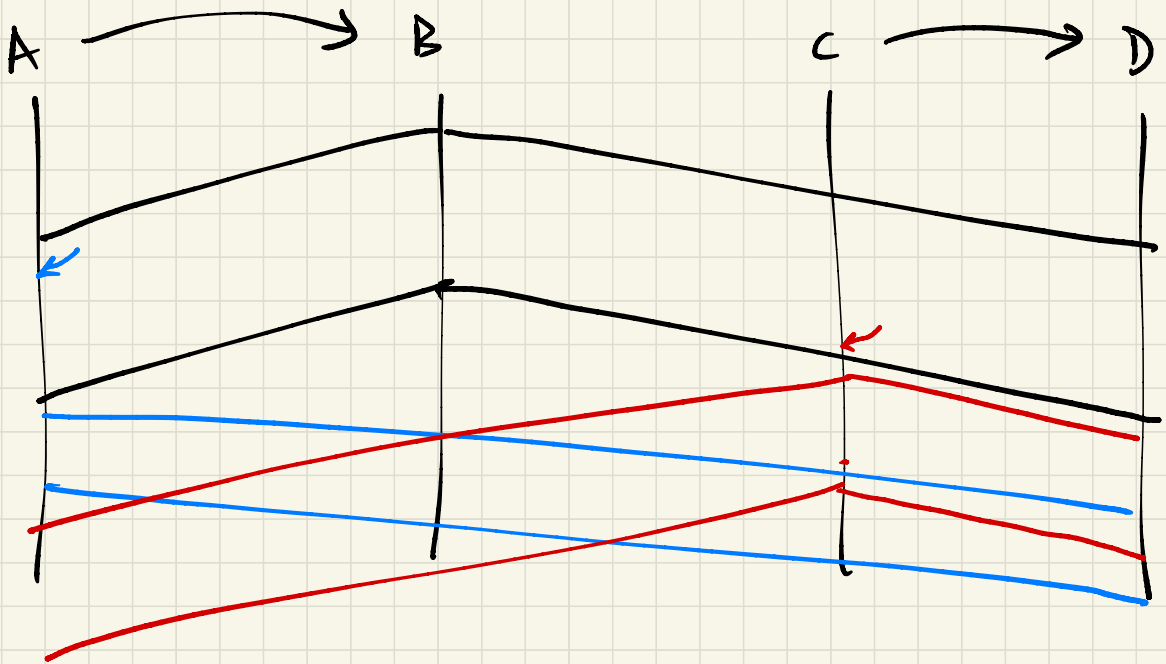
Listen before you talk

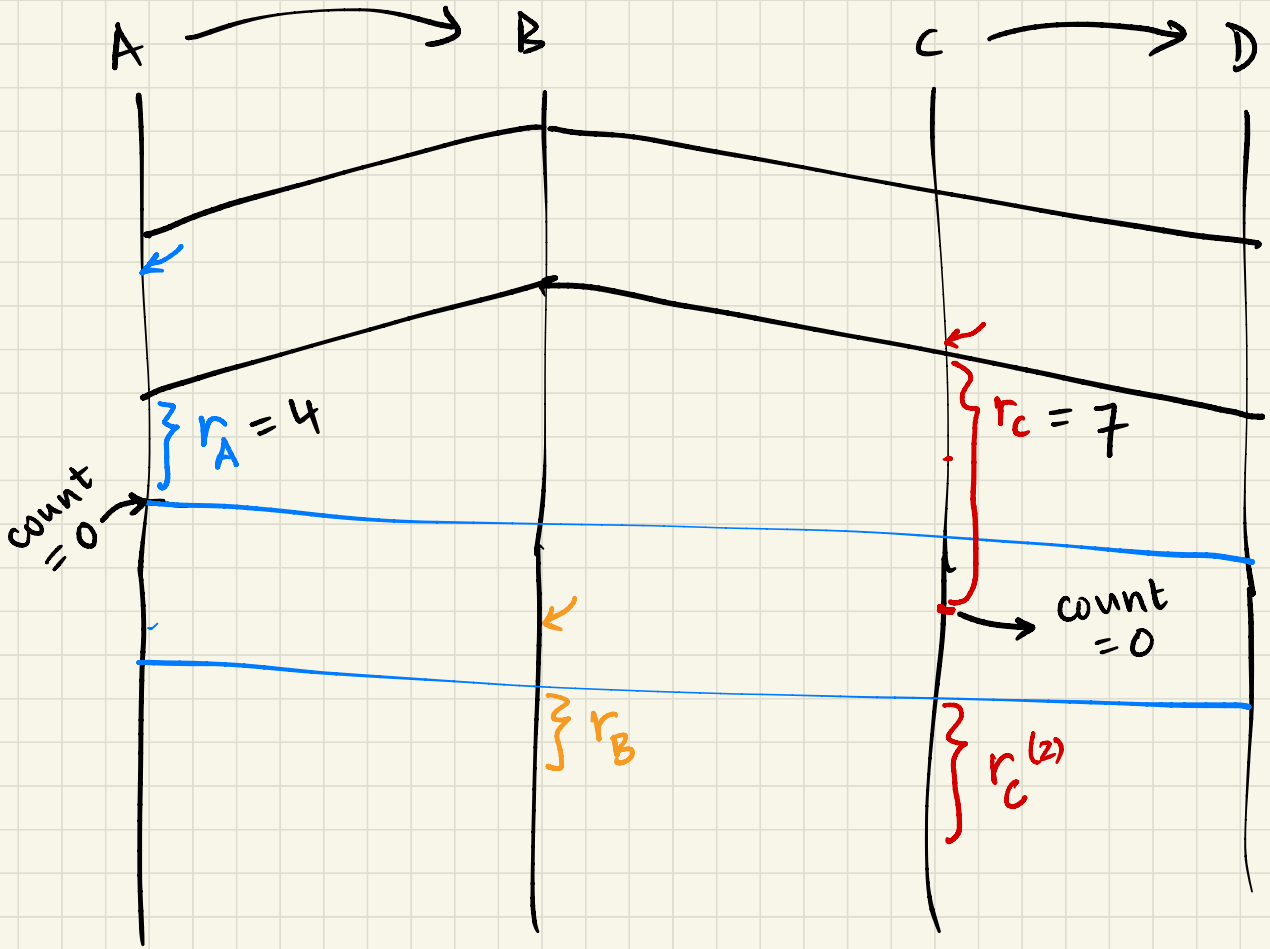
CSMA : Before A transmits, A checks if (transmitted sig == sensed signal)

red + blue

CSMA/CD → Collision Detection (red)

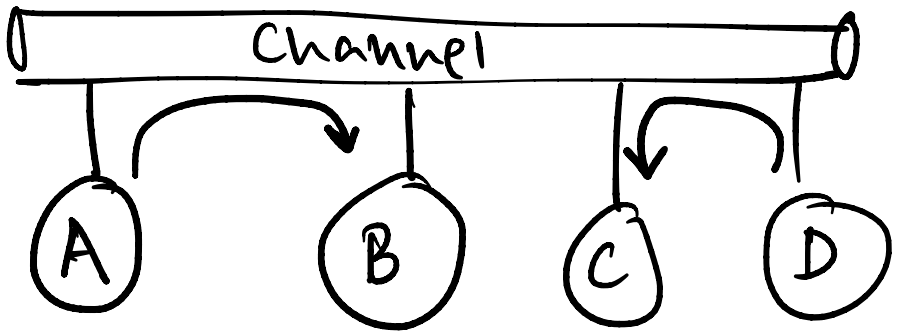

time
A
B
C
F
idle
A
B
C
D
CRC fails
collision
CRC fails
A
B
C
F
G
Round Trip time
TT > 2 × P.D max


A
B
C
D


A
B
C
D
$r_A = 4$
$r_C = 7$
count = 0
count = 0
r_B
$r_C^{(2)}$

Ethernet uses CSMA/CD

- No slots
- adapter doesn't transmit if it senses that some other adapter is transmitting, that is, carrier sense
- transmitting adapter aborts when it senses that another adapter is transmitting, that is, collision detection
- Before attempting a retransmission, adapter waits a random time, that is, random access



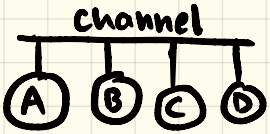
Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the mth collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits K·512 bit times and returns to Step 2

So current protocol:

Contention Window:

$CW_{min} = 15$ slots



I have a pkt to send

Listen to channel

Is channel idle

Y

N

Pick Random No. R

$[0, CW_{min}]$

Count down R to zero

Keep listening while counting down

Is Channel Busy?

N

Y

Is countdown = zero?

N

Y

Freeze Countdown

Transmit and keep listening

Is Tx done?

N

Y

Is Foreign Signal?

CSMA

Y

collision

Abort Tx

send jam

Prepare retransmission

Wait for ACK

In Ethernet, no Ack, so go back to "I have a pkt".

Reset Random # range to min. value

$= 15$

CW_{min}

Pick new Random # R+

$R \in [0, 31]$

Increase Random # range

$CW = 31$

(→) Some key points.

(1) Collision happens always at the Receiver. Transmitter may detect collision by observing a foreign signal, but that doesn't always mean collision is at T_x.

(2) Channel is wasted because of random count down ⇒ called BACKOFF. This is the price to be paid for distributed coordination.

(3) The above protocol assumes that a T_x can transmit and listen at the same time. Possible in wired networks like Ethernet. Harder in wireless networks.

(4) T_x detects foreign signal and can tell for sure that collision is happening at R_x. This assumes channel is identical at T_x and R_x. True for wired networks, not for wireless.