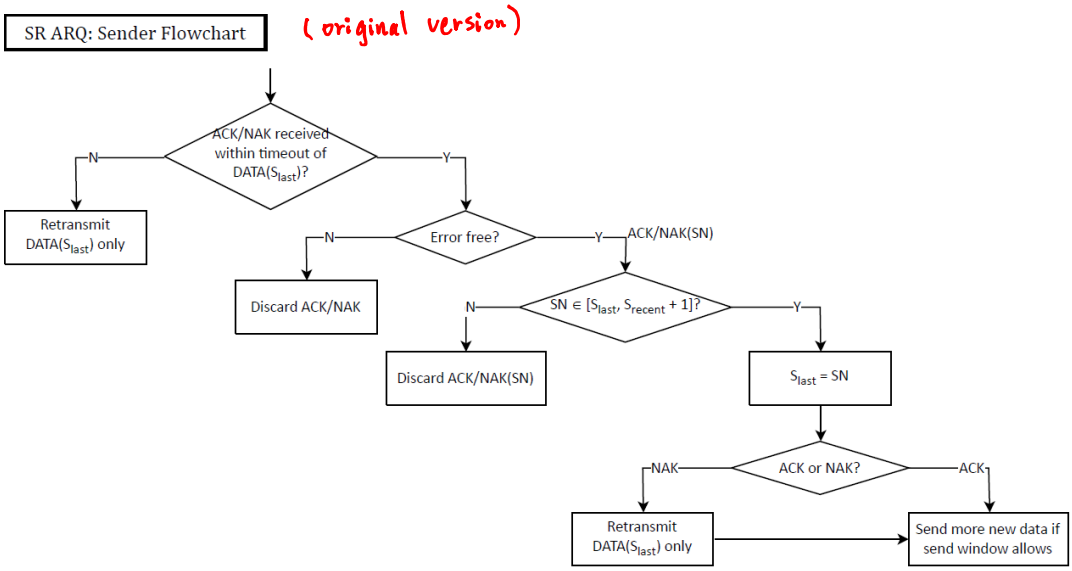
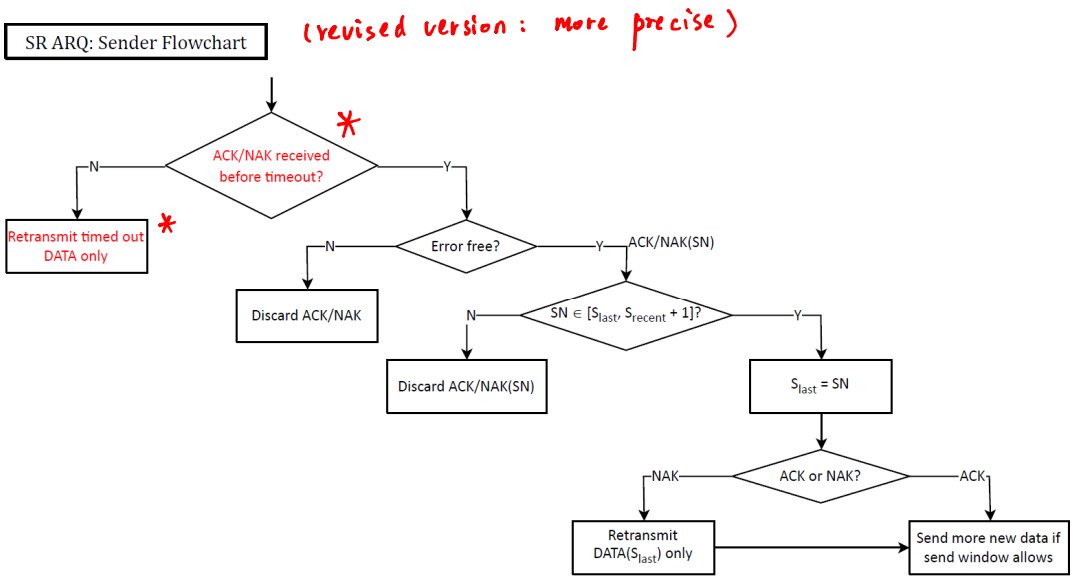
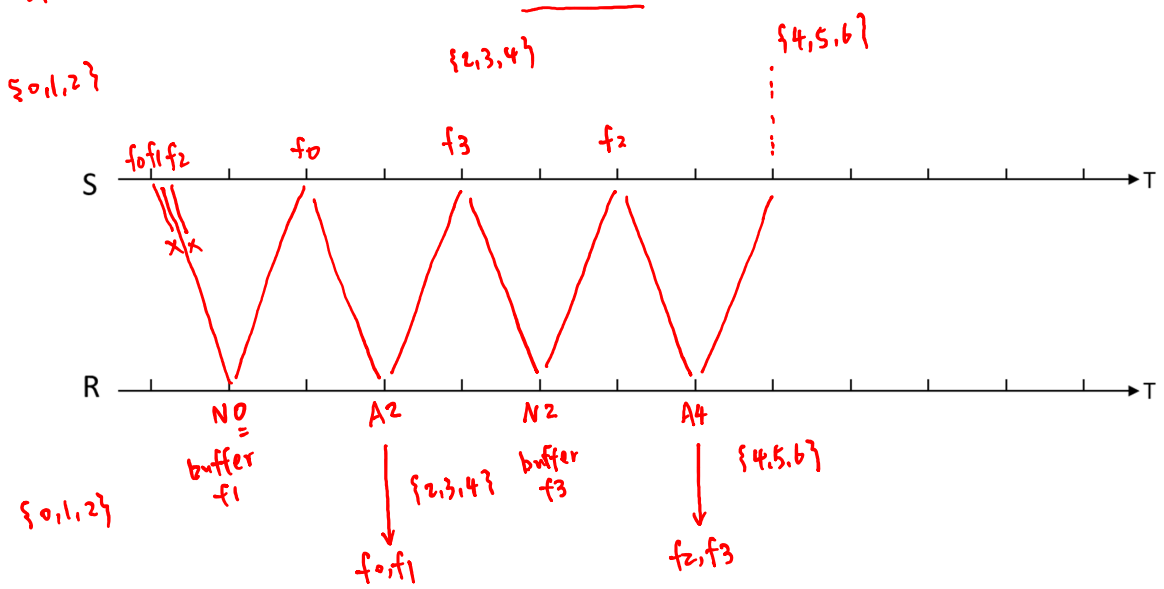

SR ARQ: Sender Flowchart
(original version)
ACK/NAK received within timeout of DATA(S_{last})?
N
Y
Retransmit DATA(S_{last}) only
Error free?
N
Y
ACK/NAK(SN)
Discard ACK/NAK
SN ∈ [S_{last}, S_{recent} + 1]?
N
Y
Discard ACK/NAK(SN)
S_{last} = SN
ACK or NAK?
NAK
ACK
Retransmit DATA(S_{last}) only
Send more new data if send window allows


SR ARQ: Sender Flowchart
(revised version: more precise)
*
ACK/NAK received before timeout?
N
Y
*
Retransmit timed out DATA only
Error free?
N
Y
ACK/NAK(SN)
Discard ACK/NAK
SN ∈ [S_{last}, S_{recent} + 1]?
N
Y
Discard ACK/NAK(SN)
S_{last} = SN
ACK or NAK?
NAK
ACK
Retransmit DATA(S_{last}) only
Send more new data if send window allows

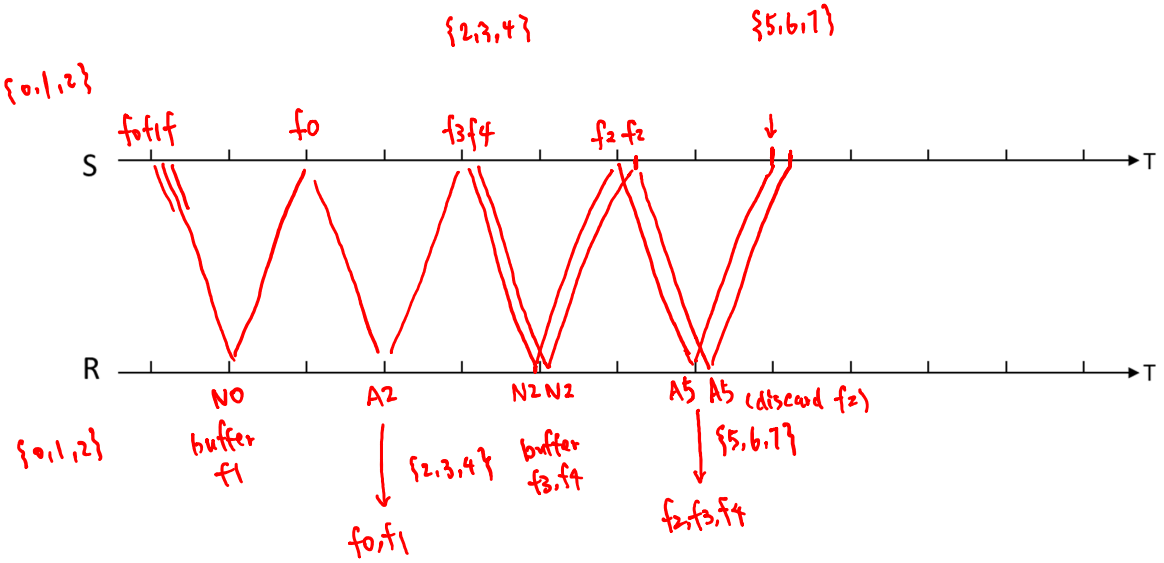
SR(3,3)

4 frames to send: f_0, f_1, f_2, f_3



SR(3,3)

5 frames to send: $f_0 \sim f_4$

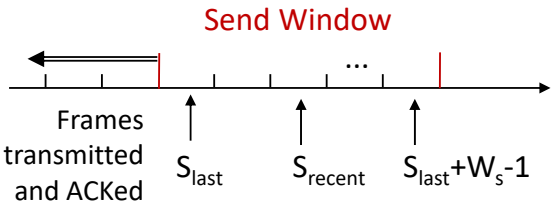


SR Protocol

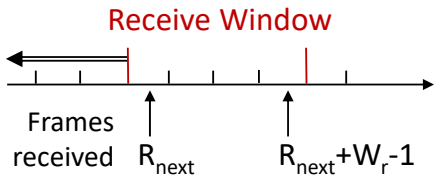
- *Essential Components:* ACK, NAK, timeout, sequence numbering
  - NAK is sent when an error-free out-of-sequence frame is received
  - ACK is sent for all other error-free frames
  - Both ACK and NAK acknowledge reception of all prior frames
- Frame in error is retransmitted upon
  - Timeout or reception of NAK
  - Only the frame in error is retransmitted
- Needs m-bit sequence numbering to remove ambiguities
  - What is the minimum value for m?
    - $\lceil \log2 (W_s + W_r) \rceil = m$

need at least $(W_s + W_r)$ unique seq. numbers

SR Transmitter & Receiver



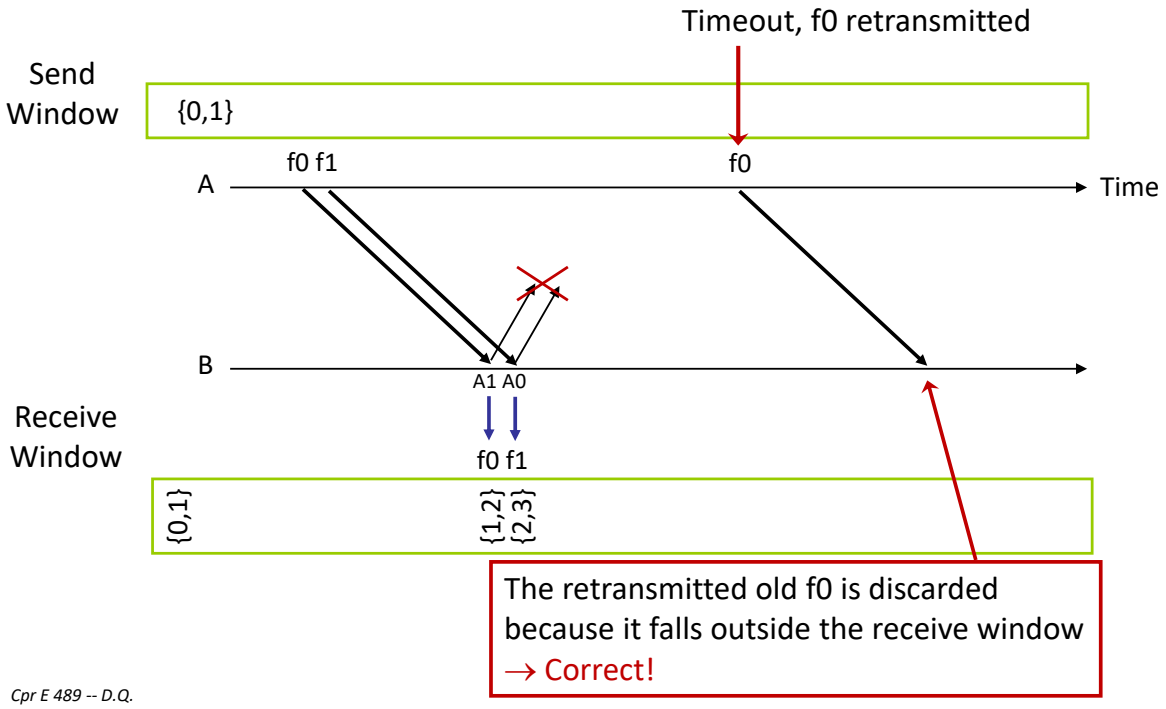
1. If an error-free ACK or NAK with $R_{next} \in [S_{last}, S_{recent}+1]$ arrives, send window slides forward: $S_{last} = R_{next}$
2. When timer for a frame expires or when a NAK arrives, transmitter retransmits the corresponding frame only



1. Receiver only accepts error-free frames with sequence number $\in [R_{next}, R_{next}+W_r-1]$
2. When frame with sequence number R_{next} arrives, R_{next} is incremented to a proper value $\Rightarrow$ receive window may slide forward by more than one
3. Erroneous frames and error-free frames with sequence number $\notin [R_{next}, R_{next}+W_r-1]$ are discarded
4. NAK is sent when an error-free out-of-sequence frame is received
5. ACK is sent for all other error-free frames received

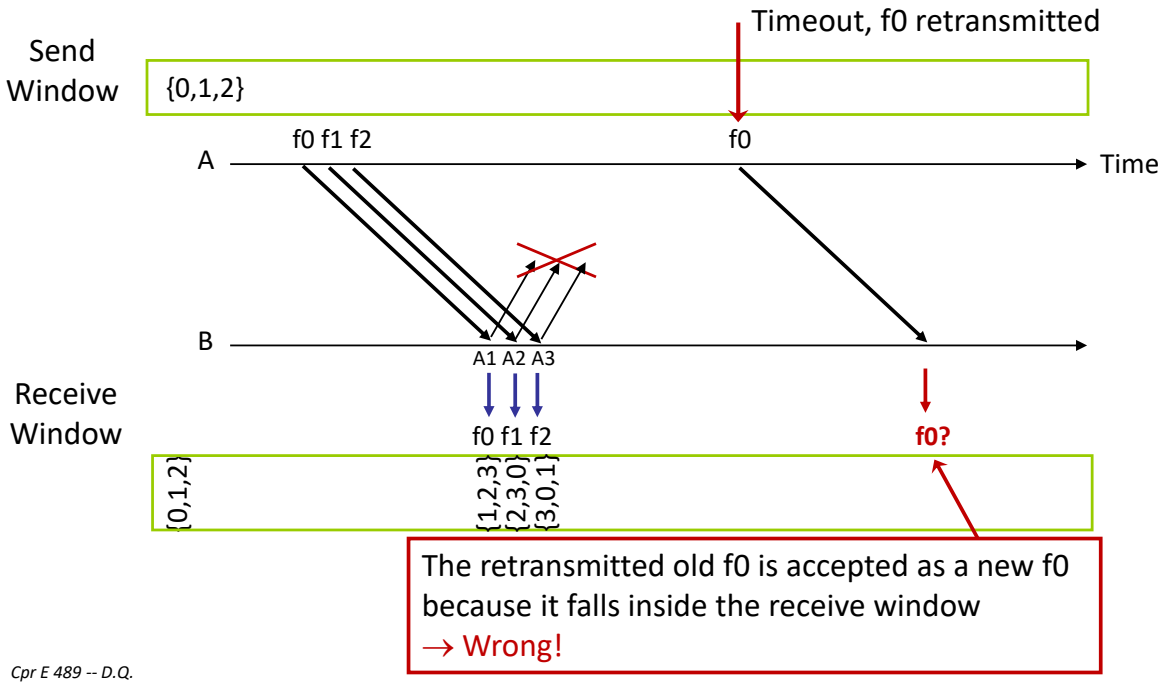
$W_s + W_r \leq 2^m$

- Example: 2-bit (m = 2) sequence numbering, $W_s = W_r = 2$



$W_s + W_r \leq 2^m$

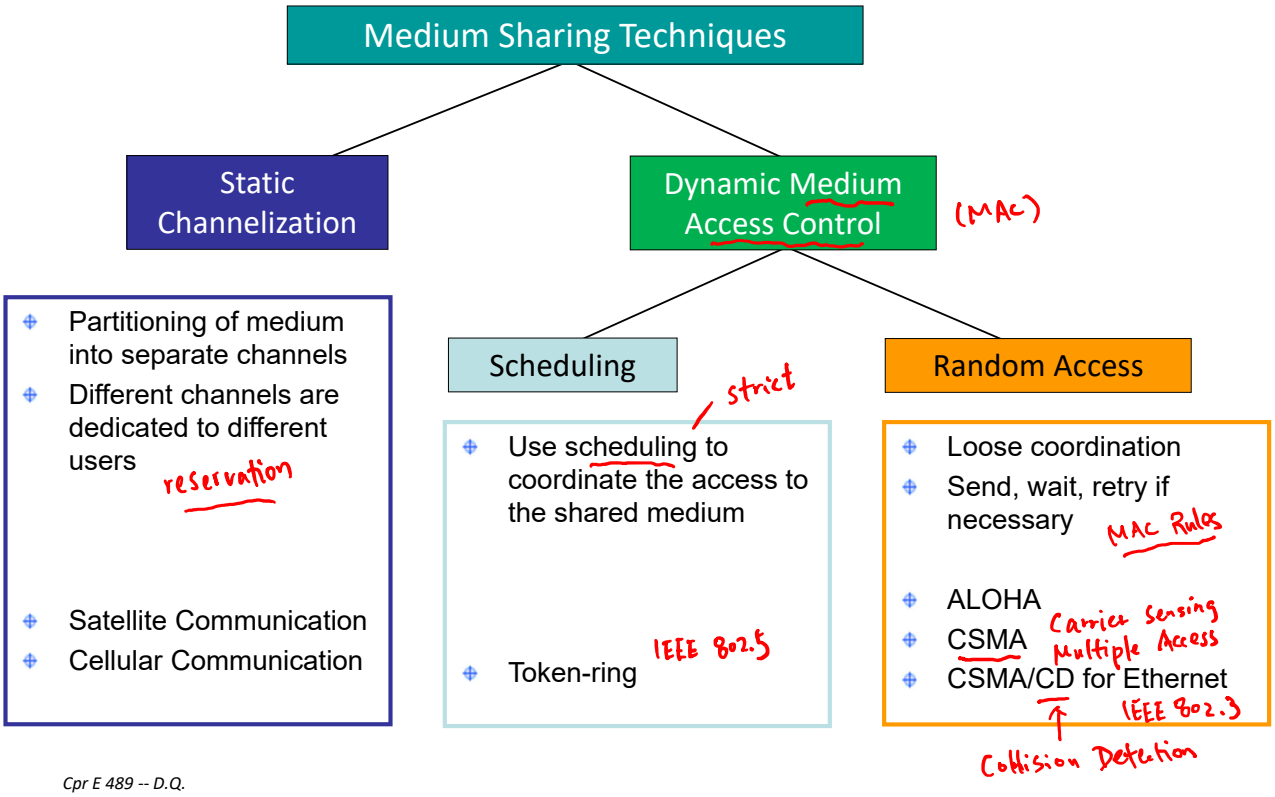
- Example: Problem when $W_s+W_r > 2^m$
  2-bit (m = 2) sequence numbering, $W_s = W_r = 3$



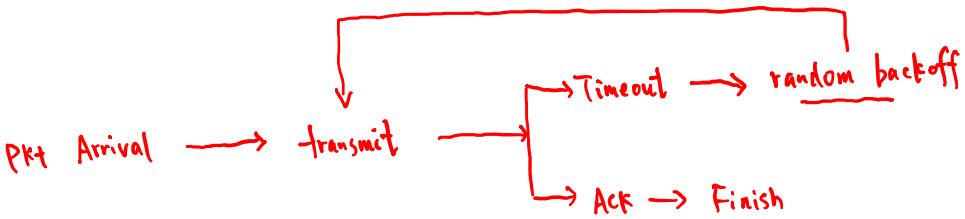
	S&W = GBN	GBN	SR	
ACK	x	x	x	
NAK			x	
Timeout mechanism	x	x	x	
Send window size	1	N	W_s	
Receive window size	1	1	W_r	
How many bits are needed for sequence numbering?	1	$\lceil \log_2(N+1) \rceil$ *	$\lceil \log_2(W_s + W_r) \rceil$ *	
Maximum # outstanding frames	1	N	W_s	
Maximum # out-of-order frames buffered at receiver	0	0	$W_r - 1$	
Transmitter: upon reception of an ACK/NAK frame with SN ∈ [S_{last}, S_{recent} + 1]	S_{last} = SN	S_{last} = SN	S_{last} = SN	
	Send window may slide forward by 1	Send window may slide forward by more than 1	Send window may slide forward by more than 1	
Transmitter: re-transmit _____ upon _____	S_{last}, timeout	from S_{last} to S_{recent}, timeout	S_{last}, timeout or NAK *	
Receiver: upon reception of an error-free in-order frame with SN = R_{next}	ACK	ACK	ACK	
	Receive window slides forward by 1	Receive window slides forward by 1	Receive window may slide forward by more than 1	
Receiver: upon reception of an error-free out-of-order frame with SN ≠ R_{next}	ACK	ACK	NAK * if it is within the receive window	ACK * if it is outside the receive window
	Receive window unchanged	Receive window unchanged	Receive window unchanged	Receive window unchanged
	Discard the frame	Discard the frame	Buffer the frame if it is within the receive window: SN ∈ [R_{next} + 1, R_{next} + W_r − 1]	Discard the frame if it is outside the receive window

Topic 4: Data Link Layer

Schemes for Medium Sharing

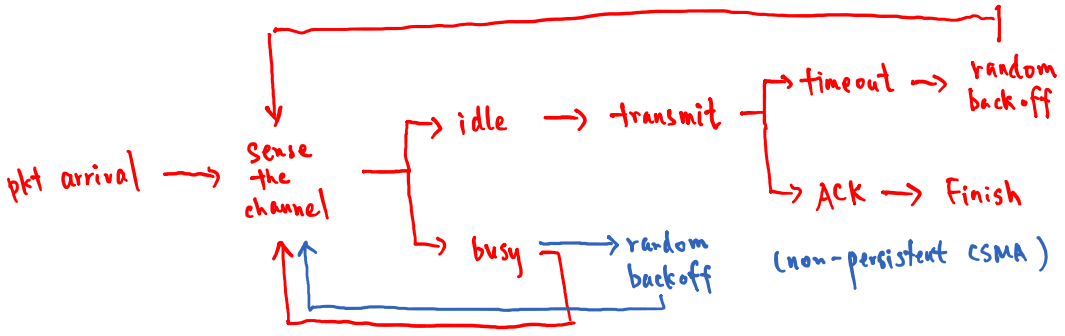


① ALOHA



2 CSMA (Carrier Sensing Multiple Access)

- A station senses (listen) the channel before it starts transmission
  - If idle, start transmission
  - If busy, either wait or schedule backoff (different options)



(1- persistent CSMA): * more greedy
* more collisions
* lower delay
→ less efficient
more efficient
less greedy
fewer collisions
higher delay