EECS 3213 Fall 2014

L14: ARQ & Reliable Data Transfer

Sebastian Magierowski
York University

Outline

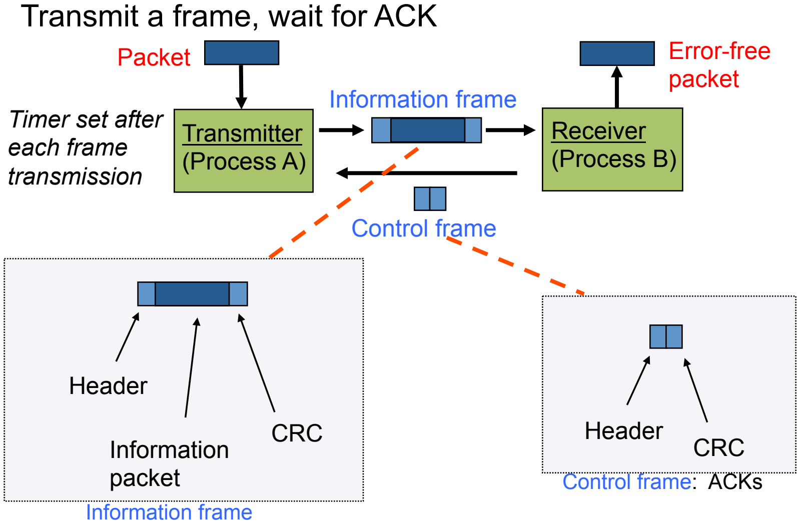
- A look at 3 ways of data link error control
- Progressively more efficient (and complex) means of transmission & re-transmission
- Can also be employed (and definitely is) by the higher layers (e.g. transport)

Automatic Repeat Request (ARQ)

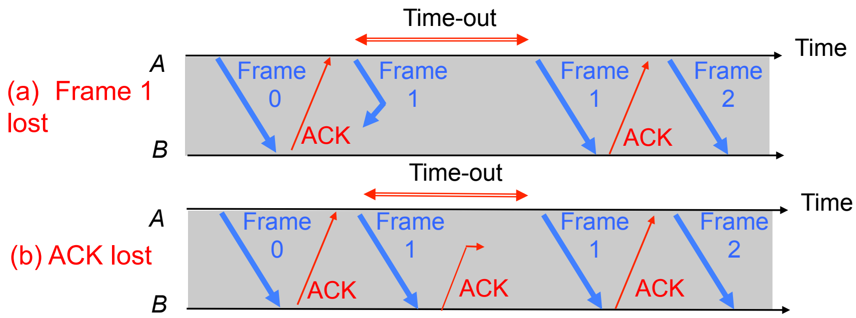
- **Purpose:** to ensure a sequence of information packets is delivered in order and without errors or duplications despite transmission errors & losses
- We will look at:
  1. Stop-and-Wait ARQ
  2. Go-Back N ARQ
  3. Selective Repeat ARQ
- Basic elements of ARQ:
  - *Error-detecting code* with high error coverage
  - *ACKs* (positive acknowledgments
  - *NAKs* (negative acknowledgments)
  - *Timeout mechanism*

Stop-and-Wait ARQ

Transmit a frame, wait for ACK



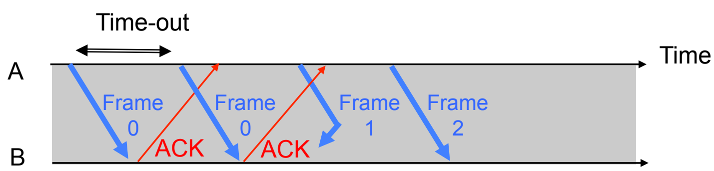
Need for Sequence Numbers



- In cases (a) & (b) the transmitting station A acts the same way
- But in case (b) the receiving station B accepts frame 1 twice
- Question: How is the receiver to know the second frame is also frame 1?
- Answer: ***Add frame sequence number in header***
- S_{last} :sequence number of most recent **transmitted** frame

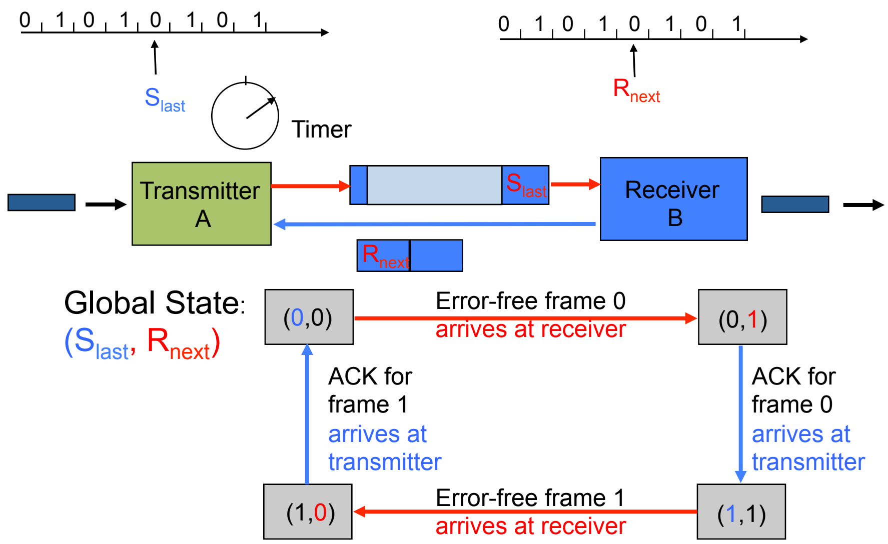
ACK Sequence Numbers

(c) Premature Time-out



- The transmitting station A misinterprets duplicate ACKs
- Incorrectly assumes second ACK acknowledges Frame 1
- Question: How is the receiver to know second ACK is for frame 0?
- Answer: ***Add frame sequence number in ACK header***
- R_{next} is sequence number of next frame expected by the receiver
- Implicitly acknowledges receipt of all prior frames

1-Bit Sequence Numbering Suffices



Stop-and-Wait ARQ Protocol Review

Transmitter

Ready state

- Await request from higher layer for packet transfer
- When request arrives, transmit frame with updated S_{last} and CRC
- Go to Wait State

Wait state

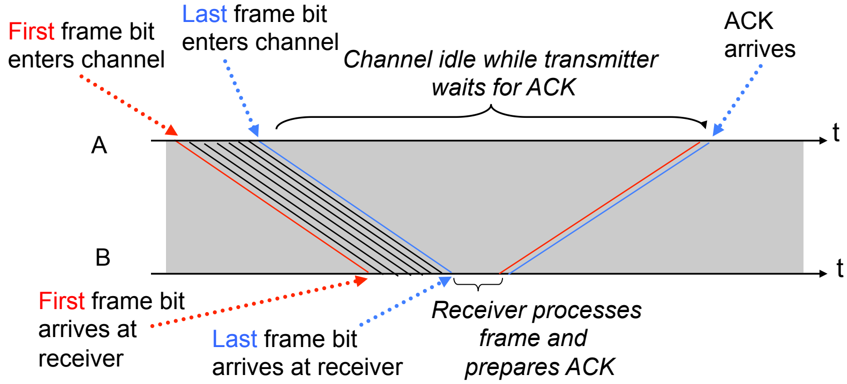
- Wait for ACK or timer to expire; block requests from higher layer
- If timeout expires
  - retransmit frame and reset timer
- If ACK received:
  - If sequence number is incorrect or if errors detected: ignore ACK
  - If sequence number is correct ($R_{next} = S_{last} + 1$): accept frame, $S_{last} = R_{next}$ go to Ready state

Receiver

Always in Ready State

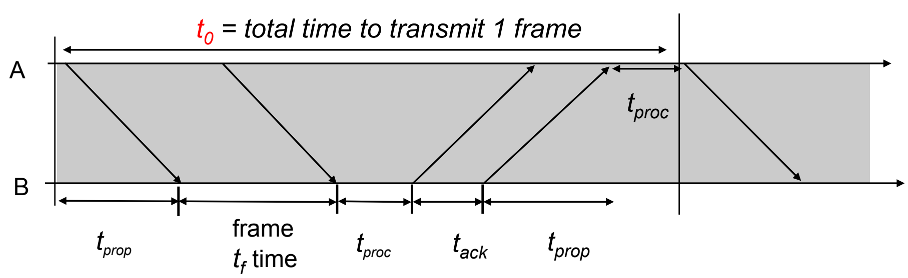
- Wait for arrival of new frame
- When frame arrives, check for errors
- If no errors detected and sequence number is correct ($S_{last} = R_{next}$), then
  - accept frame,
  - update R_{next},
  - send ACK frame with R_{next},
  - deliver packet to higher layer
- If no errors detected and wrong sequence number
  - discard frame
  - send ACK frame with R_{next}
- If errors detected
  - discard frame

Stop-and-Wait Efficiency



- 10000 bit frame @ 1 Mbps takes 10 ms to transmit
- If wait for ACK = 1 ms, then efficiency = 10/11= 91%
- If wait for ACK = 20 ms, then efficiency =10/30 = 33%

Stop-and-Wait Model



$$t_0 = 2t_{prop} + 2t_{proc} + t_f + t_{ack}$$

$$= 2t_{prop} + 2t_{proc} + \frac{n_f}{R} + \frac{n_a}{R}$$

bits/info frame

bits/ACK frame

channel transmission rate

S&W Efficiency on Error-Free Channel

Effective transmission rate:

bits for header & CRC

$$R_{eff} = \frac{\text{number of information bits delivered to destination}}{\text{total time required to deliver the information bits}} = \frac{n_f - n_o}{t_0},$$

Transmission efficiency:

$$\eta_0 = \frac{R_{eff}}{R} = \frac{\frac{n_f - n_o}{t_0}}{R} = \frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}}.$$

Effect of frame overhead

Effect of ACK frame

Effect of ***Delay-Bandwidth Product***

Example: Impact of Delay-Bandwidth Product

- n_f = 1250 bytes = 10000 bits (frame size)
- $n_a = n_o$= 25 bytes = 200 bits (overhead & ACK size)

2xDelayxBW Efficiency	1 ms 200 km	10 ms 2000 km	100 ms 20000 km	1 sec 200000 km
1 Mbps (10 ms)	10^3 88%	10^4 49%	10^5 9%	10^6 1%
1 Gbps (0.1 ms)	10^6 1%	10^7 0.1%	10^8 0.01%	10^9 0.001%

Stop-and-Wait does not work well for very high speeds or long propagation delays

S&W Efficiency in Channel with Errors

- Let 1 – P_f = probability frame arrives w/o errors
- Avg. # of transmissions to first correct arrival is then 1/ (1– P_f)
- *"If 1-in-10 get through without error, then avg. 10 tries to success"*
- Avg. Total Time per frame is then $t_0/(1 - P_f)$

$$\eta_{SW} = \frac{R_{eff}}{R} = \frac{\dfrac{n_f - n_o}{t_0 / (1 - P_f)}}{R} = \frac{1 - \dfrac{n_o}{n_f}}{1 + \dfrac{n_a}{n_f} + \dfrac{2(t_{prop} + t_{proc})R}{n_f}}(1 - P_f)$$

Effect of frame loss

Example: Impact Bit Error Rate

- n_f = 1250 bytes = 10000 bits
- $n_a = n_o$ = 25 bytes = 200 bits

Find efficiency for random bit errors with p=0, 10^{-6}, 10^{-5}, 10^{-4}

$$1 - P_f = (1 - p)^{n_f} \approx e^{-n_f p} \text{ for large } n_f \text{ and small } p$$

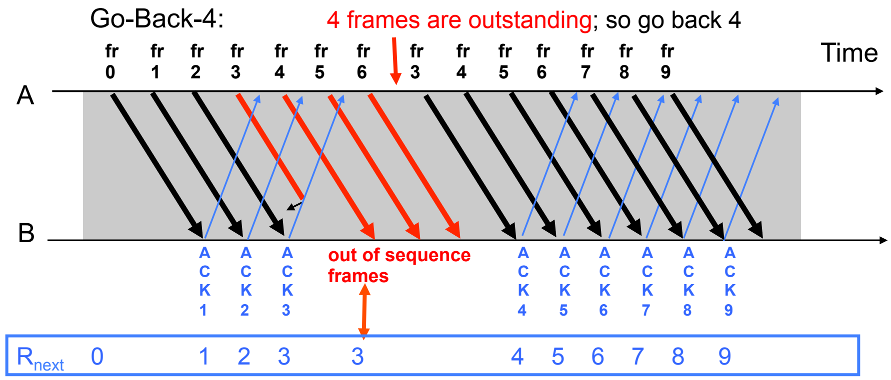
$1 - P_f$ Efficiency	0	10^{-6}	10^{-5}	10^{-4}
1 Mbps & 1 ms	1 88%	0.99 86.6%	0.905 79.2%	0.368 32.2%

Bit errors impact performance as $n_f p$ approaches 1

Go-Back-N ARQ

- Improve Stop-and-Wait by not waiting!
- Keep channel busy by continuing to send frames
- Allow a **window** of up to W_s outstanding frames
- Use *m*-bit sequence numbering
- If ACK for oldest frame arrives before window is exhausted, we can continue transmitting
- If window is exhausted, pull back and retransmit all outstanding frames
- Alternative: Use timeout

Go-Back-N ARQ

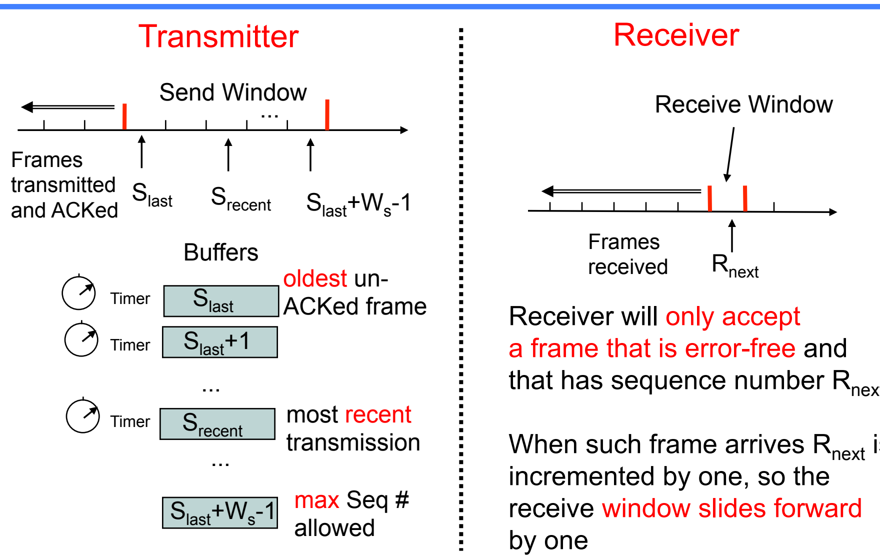


- Frame transmission are pipelined to keep the channel busy
- Frame with errors and subsequent out-of-sequence frames are ignored
- Transmitter is forced to go back when window of 4 is exhausted

Go-Back-N with Timeout

- Problem with Go-Back-N as presented:
  - What if I run out of frames to send before the end of a window?
    - Effectively A won't exhaust its window
  - But…
    - If earlier frame in A's window is lost we will not re-transmit it
      - Because our window is effectively not exhausted
    - So a frame is permanently lost!!!
- Solution: Use a timeout with each frame
  - When timeout expires, resend all outstanding frames

Go-Back-N Transmitter & Receiver

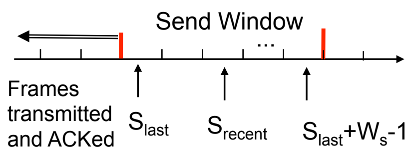


Receiver will only accept a frame that is error-free and that has sequence number R_{next}

When such frame arrives R_{next} is incremented by one, so the receive window slides forward by one

Sliding Window Operation

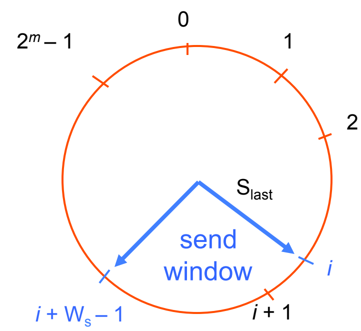
Transmitter



Transmitter waits for error-free ACK frame with sequence number S_{last}

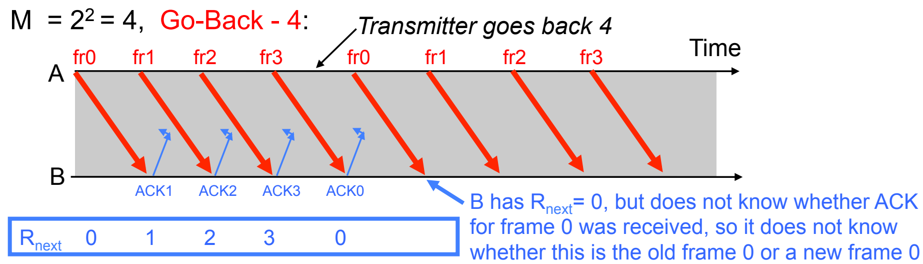
When such ACK frame arrives, S_{last} is incremented by one, and the send window slides forward by one

m-bit Sequence Numbering

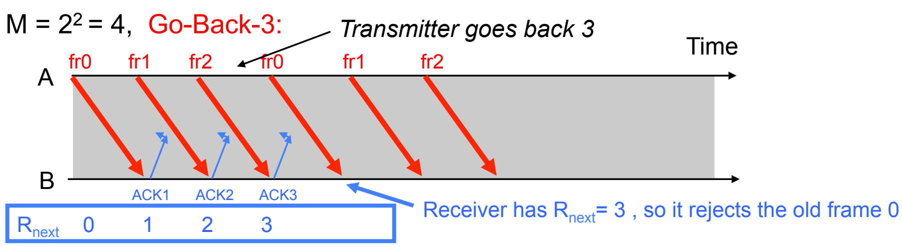


Maximum Allowable Window Size is $W_s = 2^m - 1$

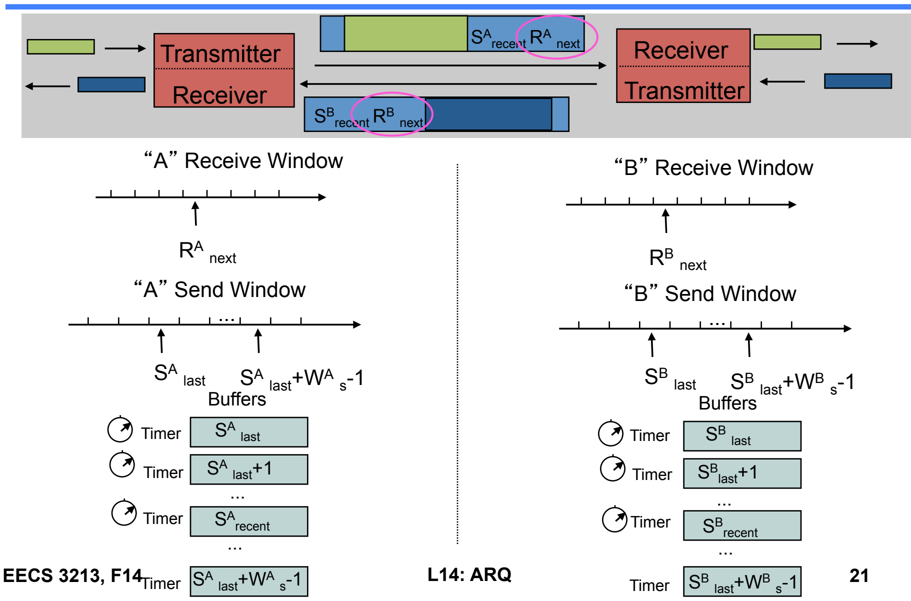
M $= 2^2 = 4$, Go-Back - 4:



M $= 2^2 = 4$, Go-Back-3:



ACK Piggybacking in Bidirectional GBN



Required Window Size for Delay-Bandwidth Product

Frame = 1250 bytes =10,000 bits, *R* = 1 Mbps		
$2(t_{prop} + t_{proc})$	**2 x Delay x BW**	**Window**
1 ms	1000 bits	2
10 ms	10,000 bits	2
100 ms	100,000 bits	11
1 second	1,000,000 bits	101

Efficiency of Go-Back-N

- GBN is completely efficient, if W_s large enough to keep channel busy, and if channel is error-free
- Assume P_f frame loss probability, then time to deliver a frame is:
  - t_f if first frame transmission succeeds $(1 – P_f)$
  - $t_f + W_s t_f/(1-P_f)$ if first transmission does not succeed (P_f)

$$t_{GBN} = t_f(1-P_f) + P_f\{t_f + \frac{W_s t_f}{1-P_f}\} = t_f + P_f \frac{W_s t_f}{1-P_f} \quad \text{and}$$

$$\eta_{GBN} = \frac{\dfrac{n_f - n_o}{t_{GBN}}}{R} = \frac{1-\dfrac{n_o}{n_f}}{1+(W_s-1)P_f}(1-P_f)$$

Delay-bandwidth product determines W_s

Example: Impact of BER on GBN

- n_f = 1250 bytes = 10000 bits
- n_a=n_o=25 bytes = 200 bits
- Compare S&W with GBN efficiency for random bit errors with p = 0, 10^{-6}, 10^{-5}, 10^{-4} and R = 1 Mbps & 2*delay = 100 ms
- 1 Mbps x 100 ms = 100000 bits = 10 frames → Use W_s = 11

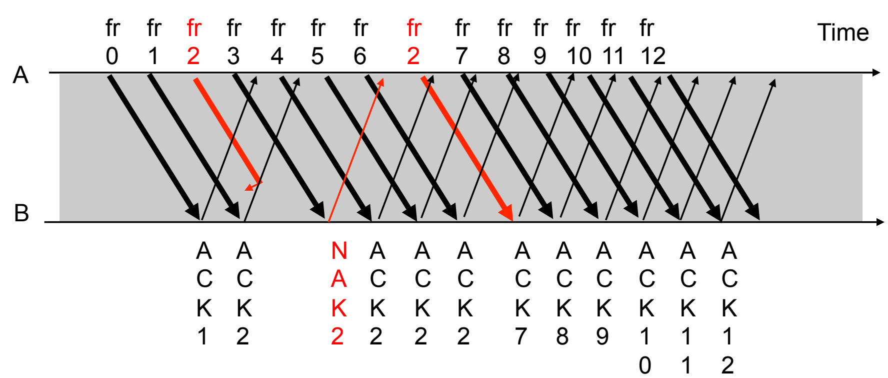
Efficiency	0	10^{-6}	10^{-5}	10^{-4}
S&W	8.9%	8.8%	8.0%	3.3%
GBN	98%	88.2%	45.4%	4.9%

- Go-Back-N significant improvement over Stop-and-Wait for large delay-bandwidth product
- Go-Back-N becomes inefficient as error rate increases

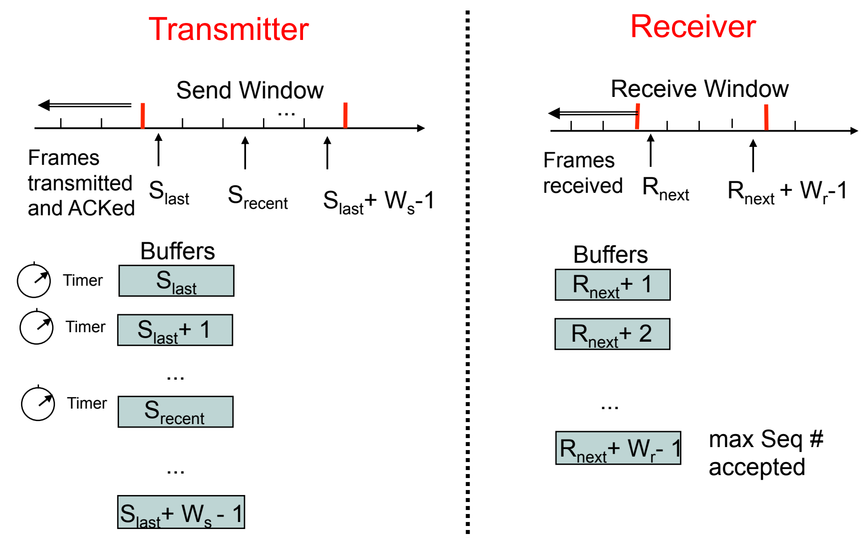
Selective Repeat ARQ

- Go-Back-N ARQ inefficient because multiple frames are resent when errors or losses occur
- Selective Repeat retransmits only an individual frame
  - Timeout causes individual corresponding frame to be resent
  - NAK causes retransmission of oldest un-acked frame
- Receiver maintains a receive window of sequence numbers that can be accepted
  - Error-free, but out-of-sequence frames with sequence numbers within the receive window are buffered
  - Arrival of frame with R_{next} causes window to slide forward by 1 or more

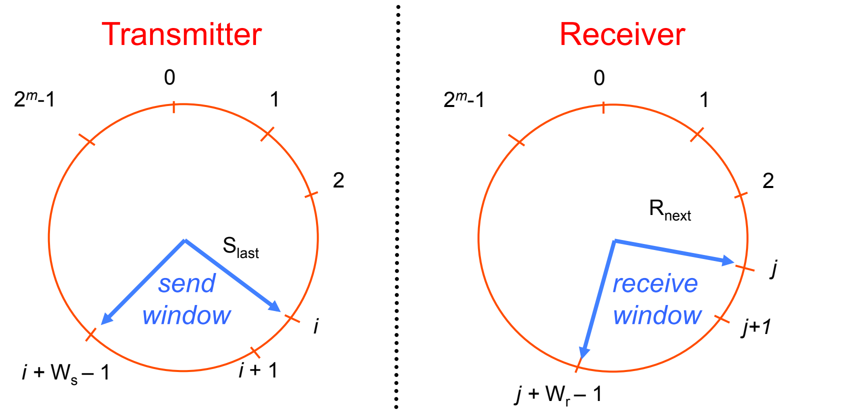
Selective Repeat ARQ



Selective Repeat ARQ



Send & Receive Windows

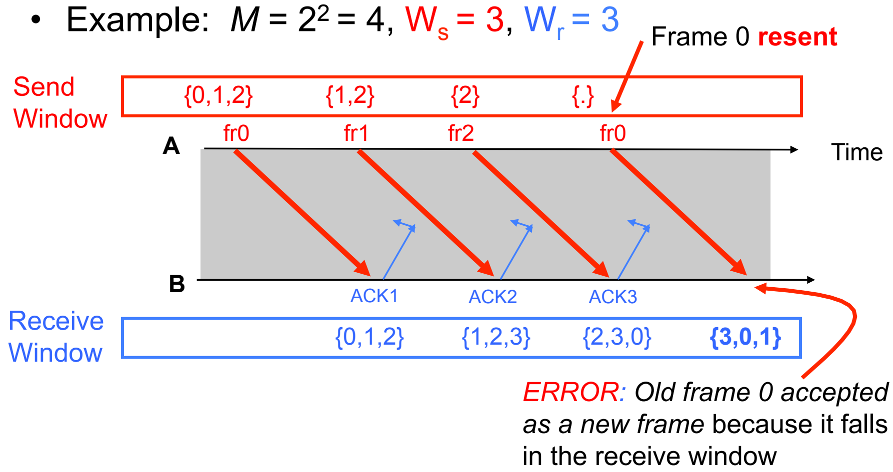


Moves k forward when ACK arrives with $R_{next} = S_{last} + k$
$k = 1, \ldots, W_s-1$

Moves forward by 1 or more when frame arrives with Seq. # = R_{next}

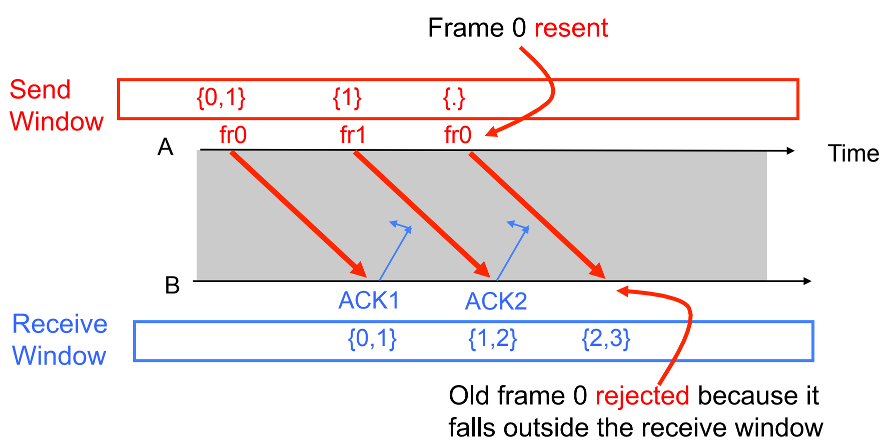
Allowable W_s and W_r Sizes

- Example: $M = 2^2 = 4$, $W_s = 3$, $W_r = 3$



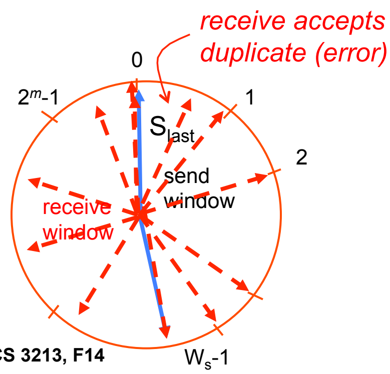
$W_s + W_r = 2^m$ is Maximum Allowed

- Example: $M = 2^2 = 4$, $W_s = 2$, $W_r = 2$

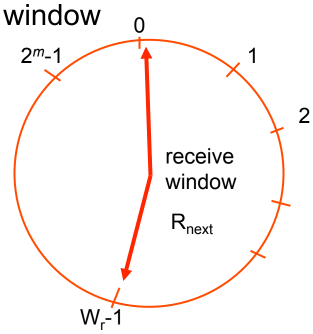


Why $W_s + W_r = 2^m$ Works

- Transmitter sends frames 0 to W_s-1; send window empty
- All arrive at receiver
- All ACKs lost
- Transmitter resends frame 0



- Receiver window starts at {0, …, W_r}
- Window slides forward to {W_s, …,W_s+W_r-1}
- Receiver rejects frame 0 because it is outside receive window



Efficiency of Selective Repeat

- Assume P_f frame loss probability, then number of transmissions required to deliver a frame is:

$t_f/(1-P_f)$

$$\eta_{SR} = \frac{\dfrac{n_f - n_o}{t_f/(1-P_f)}}{R} = (1 - \frac{n_o}{n_f})(1-P_f)$$

Example: Impact of BER on Selective Repeat

- n_f = 1250 bytes = 10000 bits
- $n_a = n_o$ = 25 bytes = 200 bits
- Compare S&W, GBN & SR efficiency for random bit errors with p=0, 10^{-6}, 10^{-5}, 10^{-4} and R = 1 Mbps & reaction time = 100 ms

Efficiency	0	10^{-6}	10^{-5}	10^{-4}
S&W	8.9%	8.8%	8.0%	3.3%
GBN	98%	88.2%	45.4%	4.9%
SR	98%	97%	89%	36%

- Selective Repeat outperforms GBN and S&W, but efficiency drops as error rate increases

Comparison of ARQ Efficiencies

- Assume n_a and n_o are negligible relative to n_f, and $L = 2(t_{prop}+t_{proc})R/n_f=(W_s-1)$, then

Selective-Repeat:

$$\eta_{SR} = (1-P_f)(1-\frac{n_o}{n_f}) \approx (1-P_f)$$

For $P_f \approx 0$, SR & GBN same

Go-Back-N:

$$\eta_{GBN} = \frac{1-P_f}{1+(W_S-1)P_f} = \frac{1-P_f}{1+LP_f}$$

For $P_f \to 1$, GBN & SW same

Stop-and-Wait:

$$\eta_{SW} = \frac{(1-P_f)}{1+\frac{n_a}{n_f}+\frac{2(t_{prop}+t_{proc})R}{n_f}} \approx \frac{1-P_f}{1+L}$$

ARQ Efficiencies