ARQ Error Control

- Two types of errors
  - Lost frames
  - Errors in frames
- Most Error Control techniques are based on
  1. Error detection scheme (e.g. parity checks, CRC), and
  2. Retransmission Scheme
- Error Control schemes that rely on error detection and retransmission of lost or corrupted frames are called **Automatic Repeat Request** (ARQ) error control

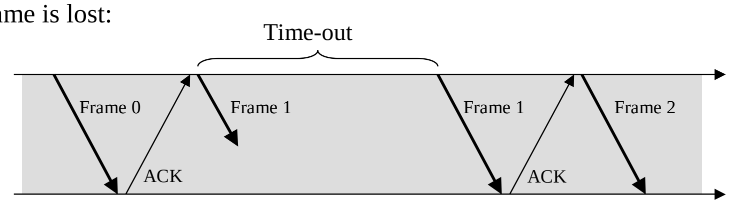
Basic Elements of ARQ

All ARQ retransmission schemes use a combination of the following components:

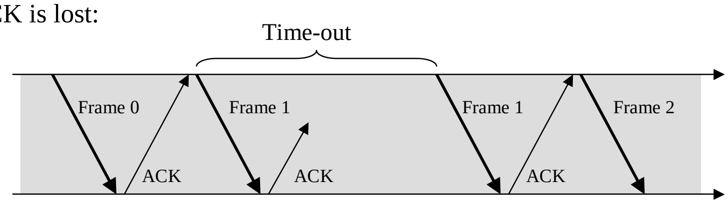
- Receiver sends an **acknowledgement (ACK)** to notify sender that frame is correctly received
- Receiver sends a **negative acknowledgement (NACK)** to notify sender that frame is incorrectly received
- Sender sends **enquiry** (ENQ) to trigger the receiver to transmit current state (in form of an ACK)
- Sender retransmits packet if ACK is not received within a **timeout** interval

Stop-and-wait ARQ

Frame is lost:

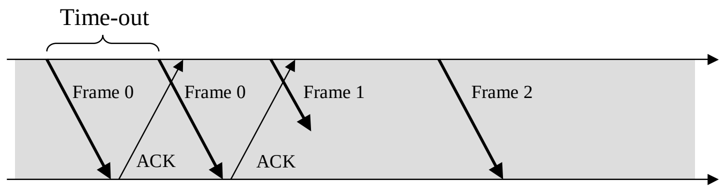


ACK is lost:



Other Ambiguities in ARQ

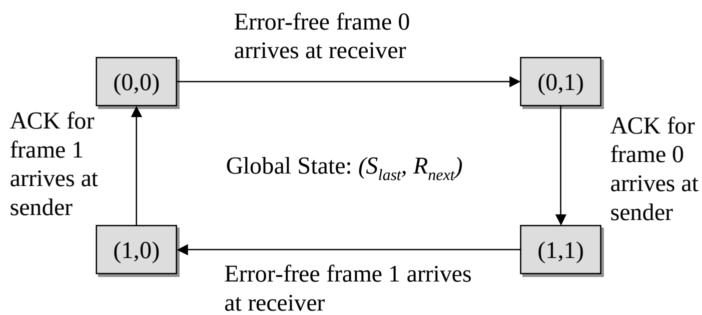
ACKs must be numbered, otherwise:



ACKs need sequence numbers in order for senders to match ACKs to frames that have been sent!

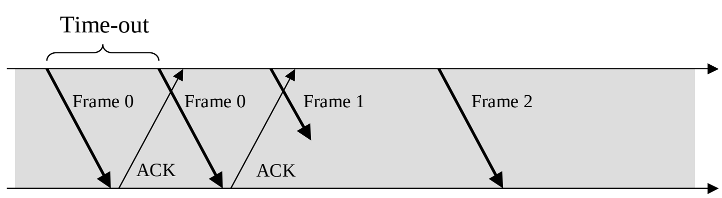
Sequence Numbers

- Sequence Numbers avoid ambiguities, but need to be limited in size (avoid overflows!)
- Example: One-bit sequence numbers in Stop-and-Wait ARQ
  - S_{last}: Sequence number of <u>last sent</u> frame at sender
  - R_{next}: Sequence number of <u>next expected</u> frame at receiver

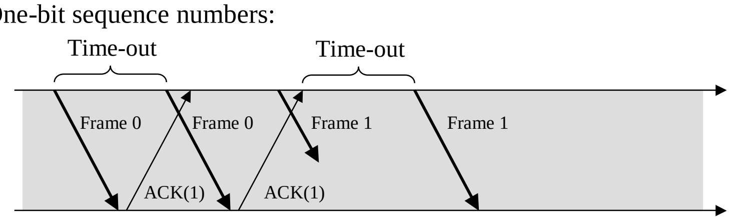


One-bit Sequence Numbers Solve Problem of Misinterpreted Duplicate ACKs

No sequence numbers:

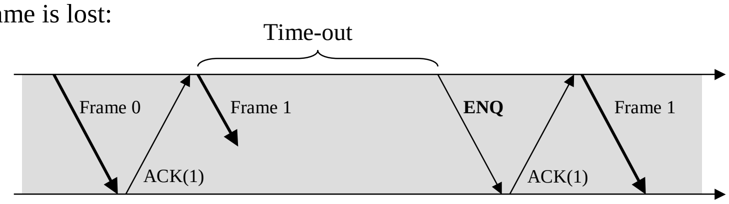


One-bit sequence numbers:

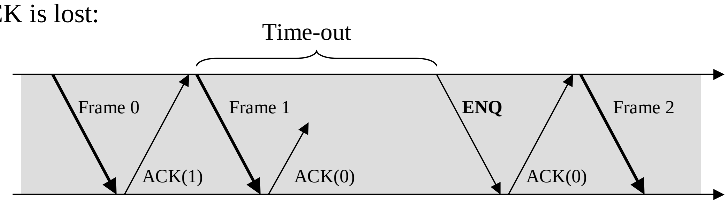


Variations of Stop-and-wait ARQ: ENQ Frames

Frame is lost:

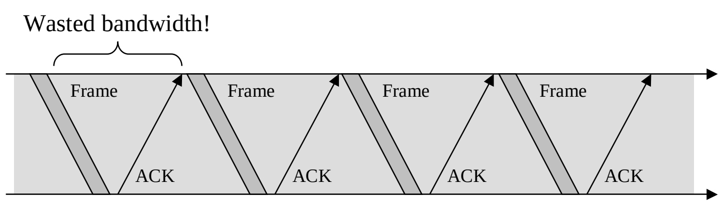


ACK is lost:



Inefficiency of Stop-and-wait ARQ

- **Efficiency** of a protocol is the maximum fraction of time when the protocol is transmitting data.

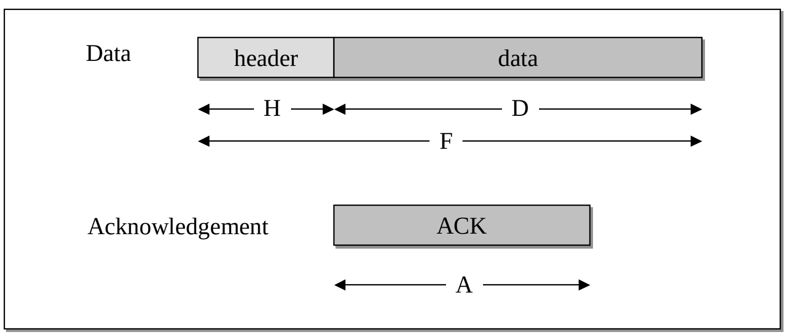


Analysis of Stop-and-wait

- Notation:

C	= Channel capacity (in bits/sec)
I	= Propagation delay
H	= Number of bits in a frame header
D	= Number of data bits in a frame
F	= Total length of a frame ($F = D+H$)
A	= Total length of an ACK frame
F/C	= Transmission delay for a frame

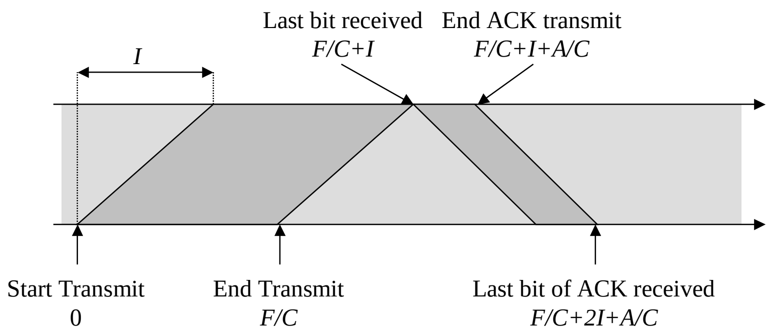
Analysis of Stop-and-wait (2)



$$a = \frac{propagation\_delay}{transmission\_delay} = \frac{I}{F/C}$$

Analysis of Stop-and-wait (3)

Transmission of a frame (in Stop-and-wait):



Analysis of Stop-and-wait (4)

- Efficiency of Stop-and-Wait (error-free case):

$$efficiency = \frac{D/C}{F/C + A/C + 2I} = \frac{D}{F + A + 2IC}$$

- Assuming that *H* and *A* are negligible, we obtain:

$$normalized\_efficiency = \frac{D}{D + 2IC} = \frac{1}{1 + 2a}$$

Example 1: Satellite Link

Satellite link:

- Roundtrip propagation delay: 270ms
- Data rate: 56kbps
- Frame length : 4000bit (including header)
- Length of ACK frames is negligible

- What is the value of *a*?
- What is the efficiency of the satellite link if Stop-and-Wait is used?

Example 2: LAN

Local Area Network:

- Data rate: 10 Mbps (100Mbps)
- Propagation rate: 200,000 km/sec
- Length of LAN: 10km
- Frame size: 500 bit (including header)
- Length of ACK frames is negligible

- What is the value for *a* for the LAN?
- How efficient is the LAN?
- What is the minimum frame size in the LAN to reach an efficiency of at least 80%?

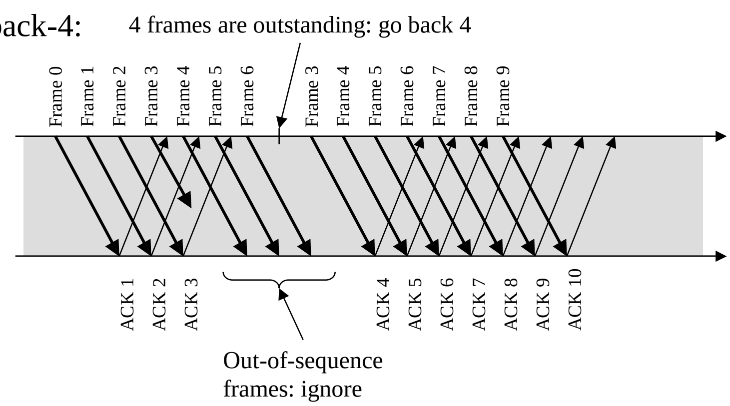
Go-back-*N* ARQ

- Problem with Stop-and-Wait ARQ:
  - Only one frame can be in transit at a time
  - This is inefficient when $a > 1$
- Solution:
  - Pipeline transmission by allowing multiple (W_s = window size) frames to be in transit, with ACKs outstanding
  - Choose W_s to be larger than bandwidth product, in order to keep channel busy

Basic Go-back-*N* ARQ

Idea of <u>basic</u> Go-back-*N* ARQ: Allow for $W_s = N$ outstanding packets at any given time. Whenever number of outstanding packets exceeds W_s, assume that something went wrong, and retransmit <u>all</u> of them.

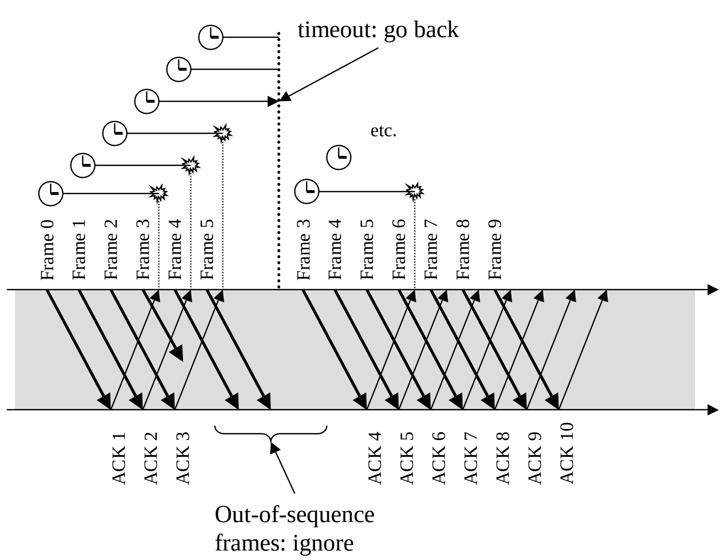
Go-back-4:



Go-back-*N* with Timers

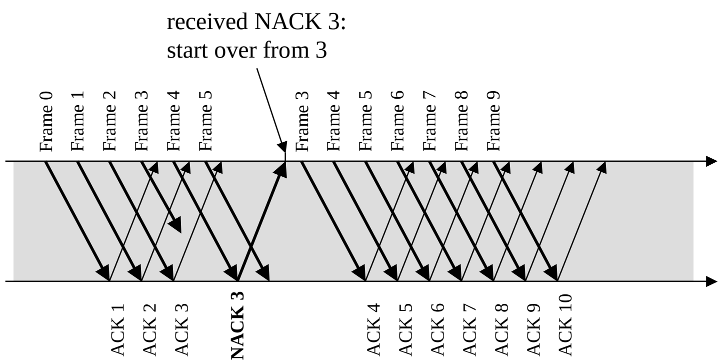
- Problem with basic Go-back-*N*:
  - Sender needs to keep sending in order to have outstanding frames, and so detect missing ACK.
- Alternative:
  - Associate timer with every outstanding frame.
  - Timeout triggers re-transmission from affected frame on.

Timers in Go-back-*N* ARQ



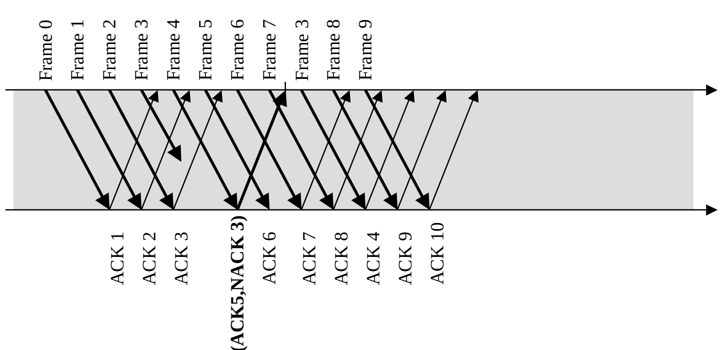
Go-back-*N* ARQ with NACKs

- Receiver sends NACK *i* if frame *i* is in error. After that, the receiver discards all incoming frames until it correctly receives frame *i*.
- If sender receives a NACK *i* it will retransmit frame *i* and all frames *i*+1, *i*+2, … that have not been acknowledged.



Selective-Repeat ARQ

- The sender only retransmits frames for which NACK is received.
- Receiver buffers frames that come out of sequence. Receiver maintains a **receiver window**.



Analysis of ARQ

- Earlier: Analysis of Stop-and-Wait ARQ for no-error case.

- Assumptions:
  - Channel can experience frame errors.
  - Control frames (ACKs, NACKs) have no errors.
  - Sizes of control frames are negligible.

Review: Analysis of Stop-and-Wait ARQ

- Notation:

C	= Channel capacity (in bits/sec)
I	= Propagation delay
F	= Total length of a frame ($F = D+H$)
$T = F/C$	= Transmission delay for a frame
P	= Probability that frame is erroneous
U	= Efficiency

- Stop-and-Wait with no error:

$$U = \frac{T}{T + 2I}$$

Stop-and-Wait ARQ: Error Case

- Probability that k transmission attempts are needed to successfully transmit a frame:

$$P[\text{k attempts needed}] = P^{k-1} \cdot (1 - P)$$

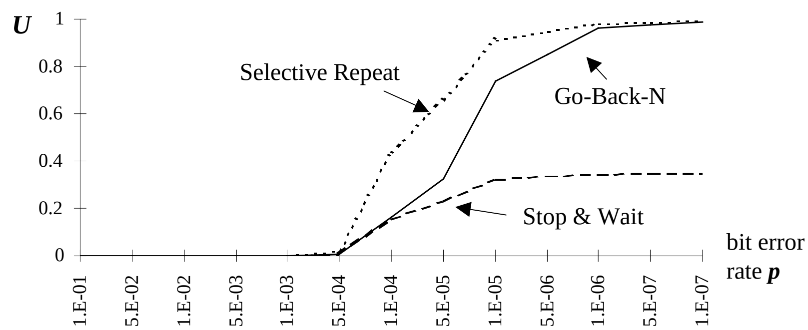
- Expected number of attempts (= *E[A]*)

$$E[A] = \sum_{i=1}^{\infty} i \cdot P^{i-1} \cdot (1 - P) = \frac{1}{1 - P}$$

- Expected efficiency with errors:

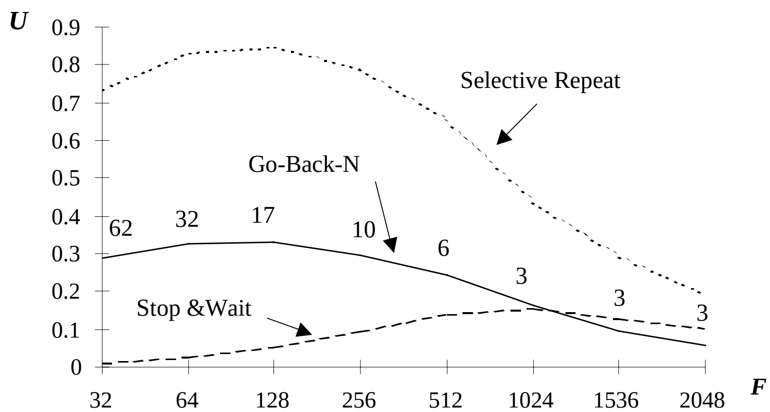
$$U = \frac{T}{E[A] \cdot (T + 2I)} = \frac{(1 - P) \cdot T}{T + 2I} = \frac{1 - P}{1 + 2a}$$

Efficiency of ARQ Protocols



1024 bit frames, 1.5 Mbps, 5msec prop delay, window size 4.

Analysis of ARQ Protocols: Optimal Frame Size

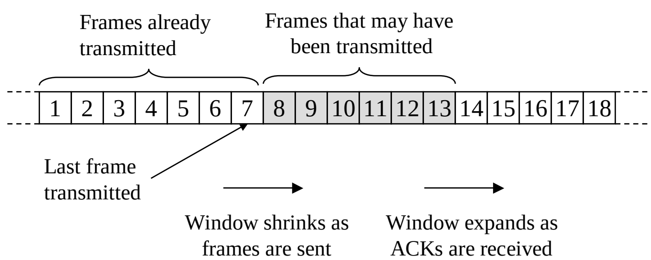


Flow Control

- Flow control is needed in order to match the speed of the sender to that of the receiver. This makes sure that data does not overflow at the receiving node.
- We assume (for simplicity) that there are no errors between sender and receiver.
- **Stop-and-Wait** Flow Control: receiver indicates readiness to receiver data for each frame.
- **Sliding Window** Flow Control allows for transmission of multiple frames.

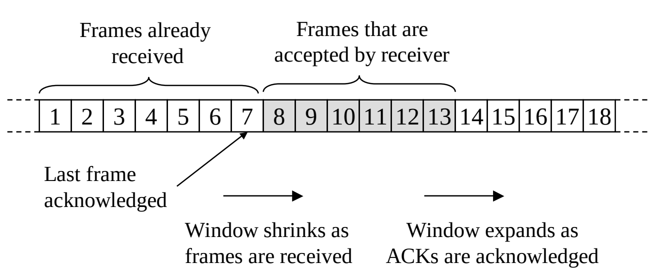
Operation of Sliding Window: Sender Window

- At any time, sender is permitted to transmit frames with sequence number is a certain range.
- This range of numbers is called the **sender window**.

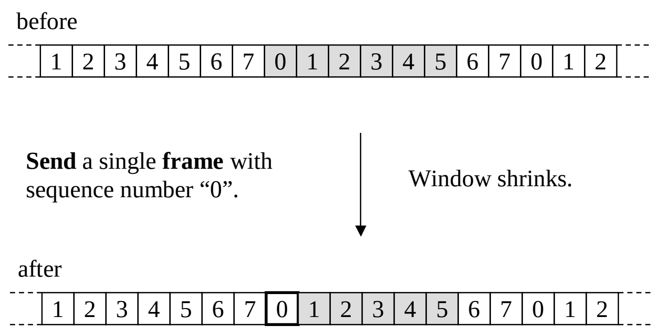


Operation of Sliding Window: Receiver Window

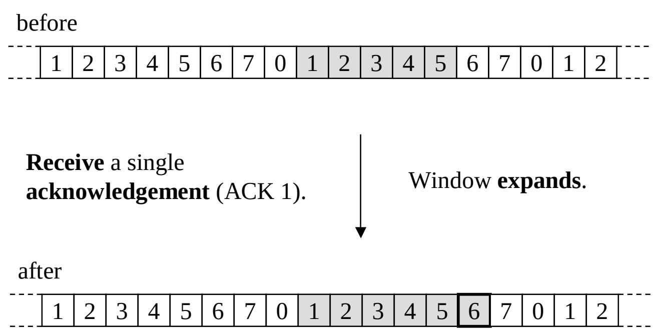
- The receiver maintains a **receiving window** of sequence numbers of frames that are accepted.



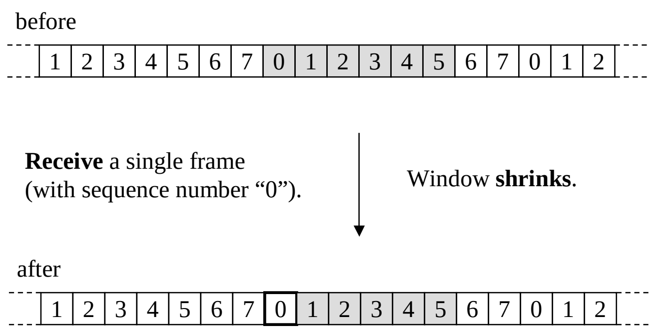
Operations at the Sender: Send Frame



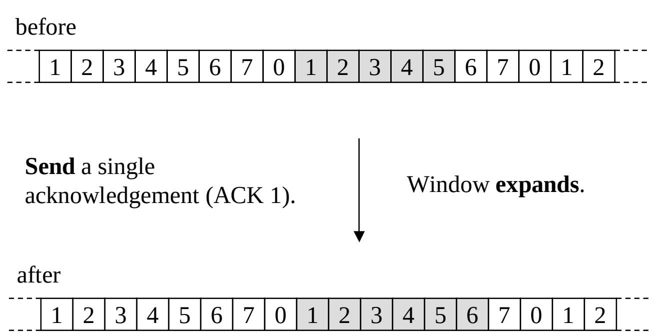
Operations at the Sender: Receive ACK



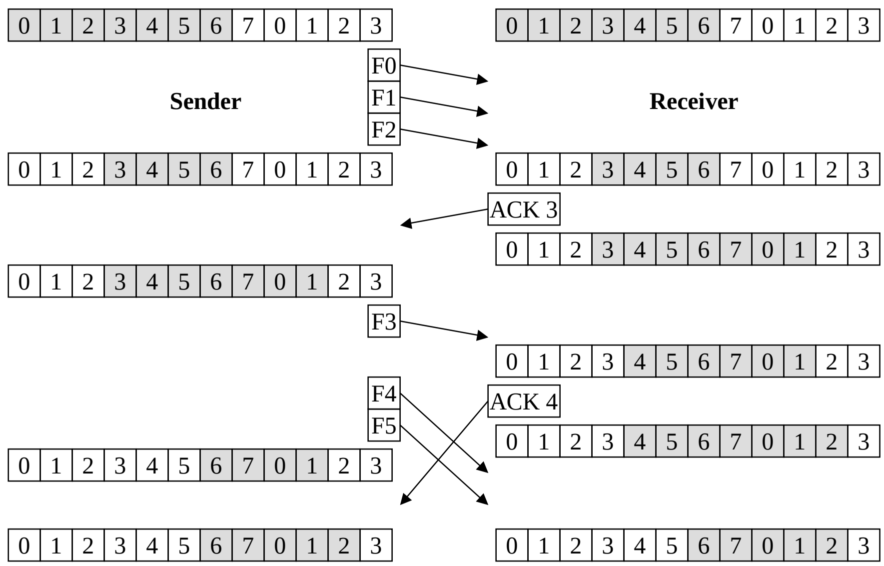
Operation at Receiver: Receive Frame



Operation at Receiver: Send ACK



Sliding Window: Example



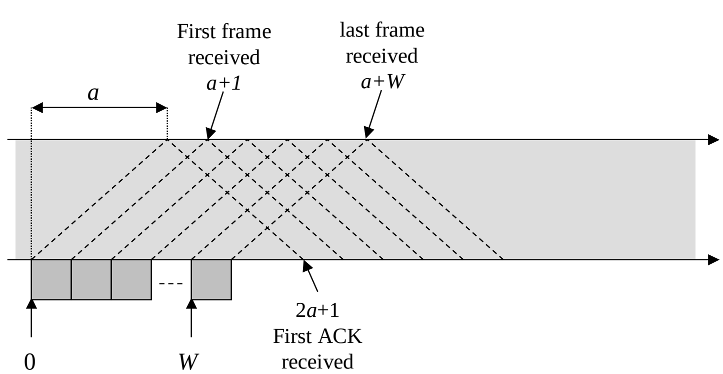
Performance of Sliding Window

- Simplify notation:

  $F/C = 1$

  $I = a$ (Normalization)

  W = Maximum window size (identical for sender and receiver)



Performance of Sliding Window

- Window must be sufficiently large in order for sender to send continuously.

- ***W >= 2a+1*** : Sender can send continuously:

$$U = 1$$

- ***W < 2a+1***: Sender can transmit *W* frames every *2a+1* time units:

$$U = \frac{W}{1+2a}$$