Automatic Repeat-Query (ARQ)

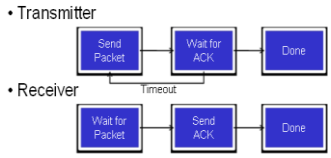
- *Purpose*: to ensure a sequence of information packets is delivered in order and without errors or duplications despite transmission errors & losses
- We will look at:
  - Stop-and-Wait ARQ
  - Go-Back N ARQ
  - Selective Repeat ARQ
- Basic elements of ARQ:
  - *Error-detecting code* with high error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

Stop and Wait

- source transmits frame
- destination receives frame and replies with acknowledgement (ACK)
- source waits for ACK before sending next
- destination can stop flow by not send ACK
- works well for a few large frames
- Stop and wait becomes inadequate if large block of data is split into small frames

Early approaches

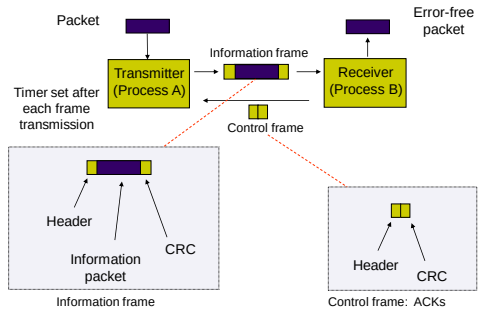
- Automatic Repeat-Query (ARQ)
- "Stop and Wait" protocol



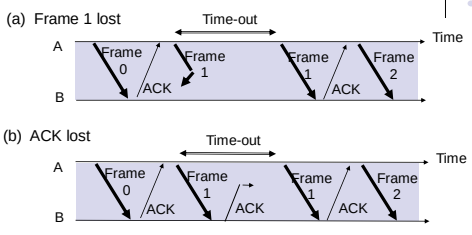
- Transmitter ensures delivery of packet N before sending packet N+1

Stop-and-Wait ARQ

Transmit a frame, wait for ACK



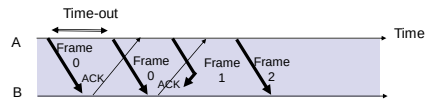
Need for Sequence Numbers



- 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***

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***
- Implicitly acknowledges receipt of all prior frames

Stop-and-Wait ARQ

Transmitter

Ready state
- Await request from higher layer for packet transfer
- When request arrives, transmit frame with CRC
- Go to Wait State

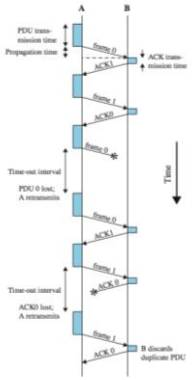
Wait state
- Wait for ACK or timer to expire; block requests from higher layer
- If timeout expires
  - retransmit frame and reset timer
- If ACK received:
  - If sequence number is incorrect or if errors detected: ignore ACK
  - If sequence number is correct
  - accept frame, go to Ready state

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.
  - accept frame,
  - send ACK frame
  - deliver packet to higher layer
- If no errors detected and wrong sequence number
  - send ACK frame
- If errors detected
  - discard frame

Stop and Wait

- see example with both types of errors
- pros and cons
  - simple
  - inefficient



Applications of Stop-and-Wait ARQ

- IBM *Binary Synchronous Communications protocol* (Bisync): character-oriented data link control
- *Xmodem*: modem file transfer protocol
- *Trivial File Transfer Protocol*: simple protocol for file transfer

Go-Back-N

- Alternative: Use timeout
- Improve Stop-and-Wait by not waiting!
- Keep channel busy by continuing to send frames
- Based on sliding window
- If no error, ACK as usual
- Use window to control number of outstanding frames
- If error, reply with rejection
  - discard that frame and all future frames until error frame received correctly.
  - transmitter must go back and retransmit that frame and all subsequent frames.

Sliding Window

- In a reliable connection-oriented data transfer, the sequence of data segments must be delivered to the receiver in the same sequence that they were transmitted.
- TCP Uses sliding windows to buffer data for transmission between two hosts.
- Each TCP/IP host maintains two sliding window: One for receiving data, and the other for sending data.
- Size of the window indicates the amount of data that can be buffered on a computer.

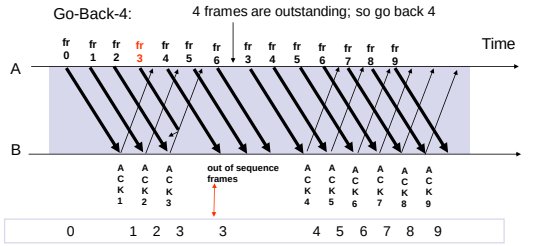
Improvement: Sliding Window Protocol

- Allow multiple unacknowledged packets in transit
- As packets are acknowledged, transmit additional ones

"Window Size" defines how many unacknowledged packets can be in transit

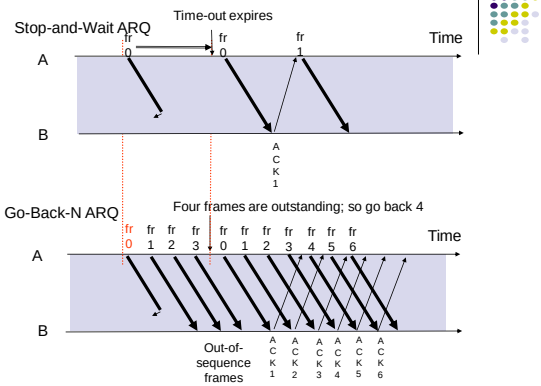
- Sender of GBN keeps a sliding window of size N on sequence numbers.
- Sender can send N packets without acknowledgement.
- Receiver of GBN expects packets to be delivered in order.
- Out-of-order packets are discarded.
- Because of this, acknowledgement in GBN is cumulative.

Go-Back-N ARQ



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

Window size should be long enough to cover a round trip time



Applications of Go-Back-N ARQ

- *HDLC* (High-Level Data Link Control): is a simple protocol used to connect point to point serial devices.

V.42 modem: error control over telephone modem links

Selective Repeat

- GBN leads to unnecessary retransmission since out of order packets are discarded.
- Selective repeat improves of GBN by allowing receiver to buffer packets that are received out of order.
- Both receiver and sender maintains a window.
- Each packet must now be individually acknowledged and we need a timer per packet.

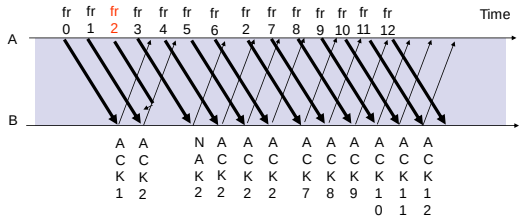
Selective Reject

- also called selective retransmission
- only rejected frames are retransmitted
- subsequent frames are accepted by the receiver and buffered
- minimizes retransmission
- receiver must maintain large enough buffer
- more complex logic in transmitter
- hence less widely used
- useful for satellite links with long propagation delays

Selective Repeat ARQ

- Go-Back-N ARQ is 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

Selective Repeat ARQ



Applications of Selective Repeat ARQ

- **TCP** (Transmission Control Protocol): transport layer protocol uses variation of selective repeat to provide reliable stream service