# Reliable Data Delivery

Jonathan Windle

University of East Anglia

*J.Windle@uea.ac.uk*

June 6, 2017

# Overview I

# Intro

- Errors in underlying network (lower layers) introduce errors that ultimately lead to packet loss.
- Transport layer should hide these errors from upper layers.
- Transport layer solutions:
  - Stop and Wait (or Idle RQ) protocol to provide an inefficient error-free stream of messages.

# Terminology

- **Data rate:** Is a measure of the speed of data transmission. It is measured in bits per second.
- **Error rate:** Is an estimte of the reliability of a communication channel. It is the proportion of corrupted bits to uncorrupted bits. e.g. Domestic telephone lines have an error rate of 1 : 100 000.

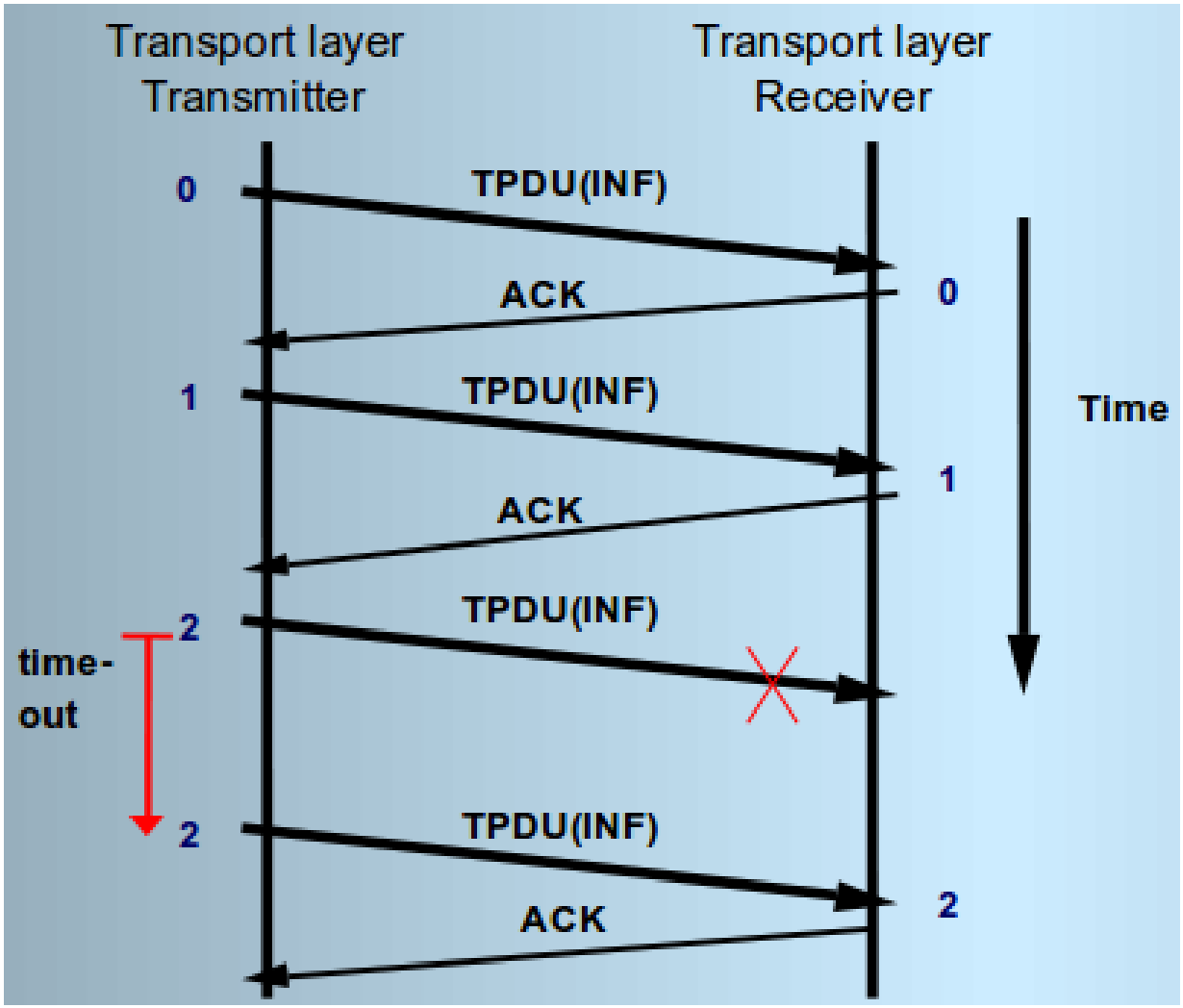
# Transport Protocol Data Unit (TPDU)

- Messages sent from transport entity to transport entity are sent in TPDU's (e.g. using TCP or UDP Layer 4 protocols).
- TPDU's comprise header and application message:
- Header:
  - Source and destination process IDs.
  - Identifier to indicate TPDU contains information (00).
  - Message length indicator.
- Application message:
  - Payload comprises packet from layer above.
- Errors can occur in either header or message at any section of the TPDU.

# Consequence of Errors

- **Destination process ID:** TPDU is not delivered to the receiver.
- **Source Process ID:** Reply is not returned to the original sender.
- **Transport layer payload:** Requires an error detectoin mechanism.
- **Length field/End of mesaage:** Only part of the TPDU is received.
- Acknowledgement Schemes:
  - Positive-Only Acknowledgement Scheme: ACK's. (Stop - Wait protocol).
  - Full Acknowledgement Scheme: ACK's and NAK's (Negative Acknowledgement).

# Stop and Wait protocol

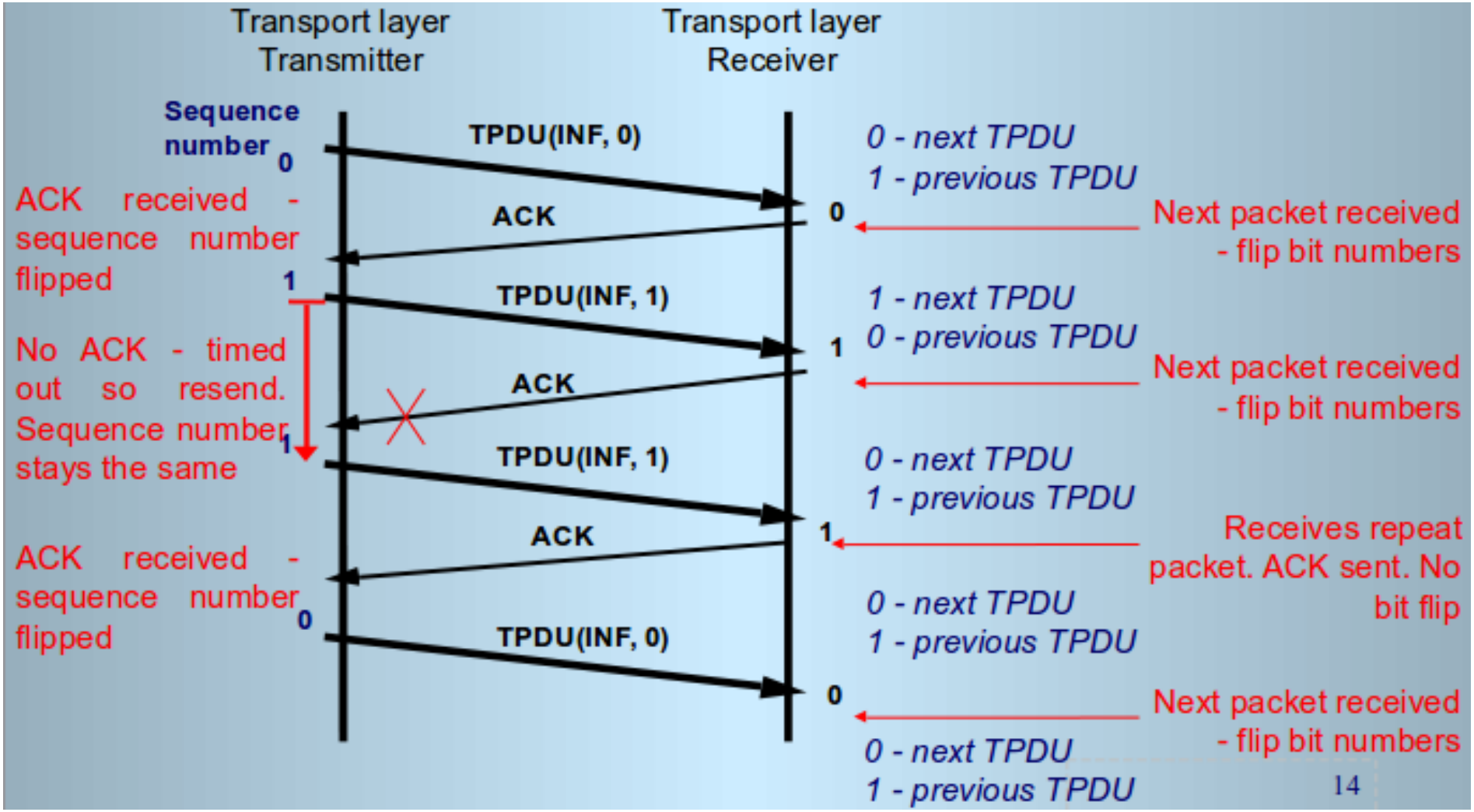



- Receiver sends an ACK in response to a correctly received TPDU.
- Transmitter starts a timer when it sends the TPDU.
- If this times out:
  1. Enquired receiver of the nature of the problem, or:
  2. Simply retransmits the TPDU.

# Some notes

- Positive-only scheme can handle destination address corruption (assuming an incorrect destination does not respond).
- Timers and time-outs are still needed tocope with errors in source and destination addresses- no ACK or NAK will be received at sender.
- On timing out, immediate retransmition of TPDU can be more efficient but requires a limit to number of re-tries - in case receiver has crashed.
- Still inadequate as ACKs and NAKs are themselves (short) TPDUs prone to corruption.
- If these are lost or corrupter the receiver cannot know if it is receiving a retransmitted last TPDU or the next TPDU.
- Solution is for TPDUs to contain a sequence number - receiver can detect multiple TPDUs.
  - Use one bit to reduce overhead and switch 0-1-0-1-0 etc. and if 0-0 is received then it is a duplicate.

# Example With Numbering



# Review

- Transport layer needs fields for:
  - Source and destination IDs
  - Length field or end of TPDU indicator.
  - Sequence number and the payload itself.
- Need for an acknowledgement scheme.
- Need for timer and timeouts and limited retries at the transmitter.
- Use of the TPDU sequence number to allow the receiver to detect duplicate TPDUs.
- Receiver sends packet and then stops and waits until acknowledgement of arrival comes through, or times-out.
- Issue is that the transmitter cannot proceed to transmit the next message until it receives an ACK / times-out.
- Solution is to use sliding window protocols

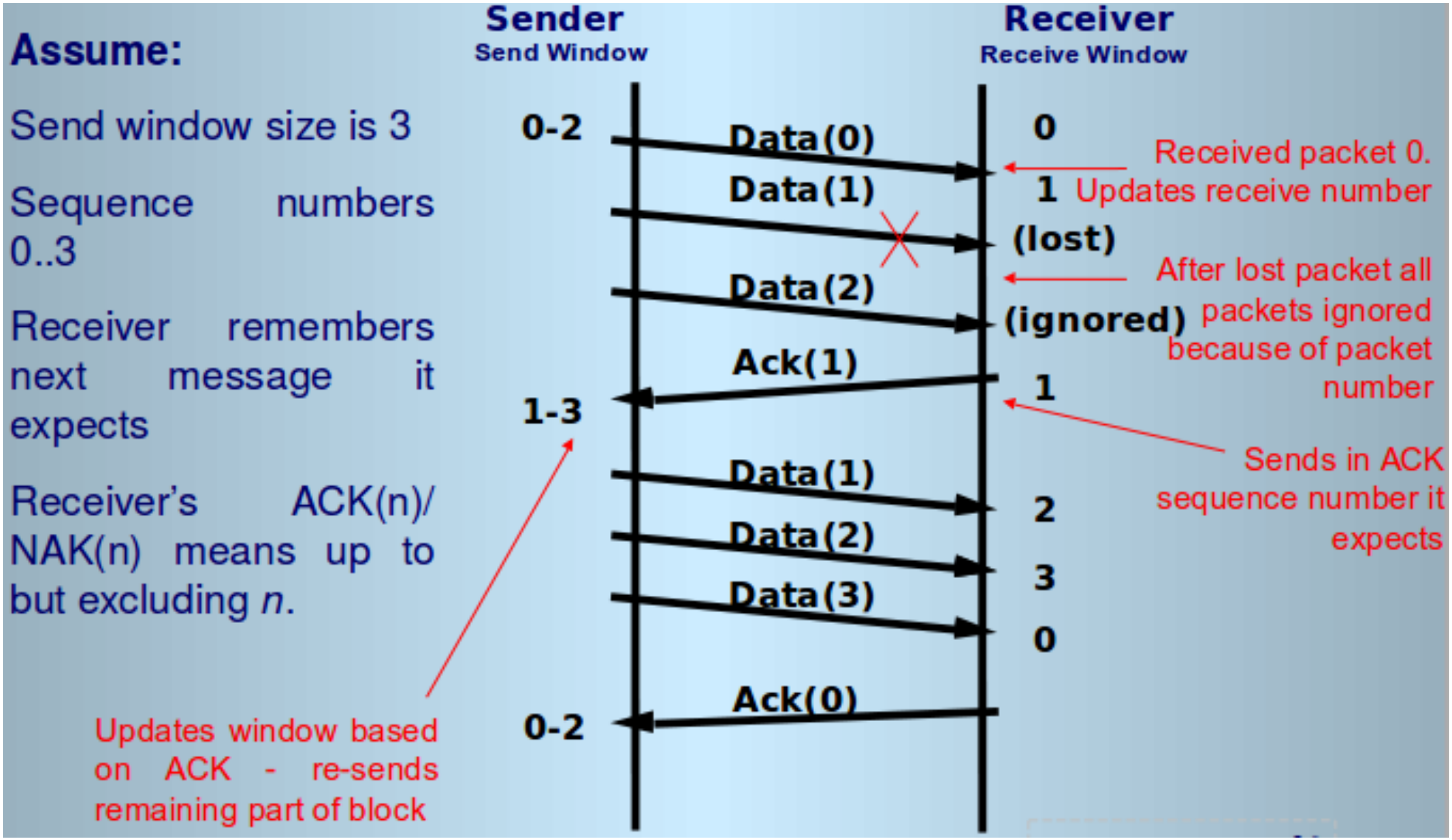
# Sliding Window Protocols

- Need for more efficient Connection Orientated Protocols.
- Problem is the transmitter waits for an ACK after each message
- Solution is to remove the restriction and allow the transmitter too have N messages in transit whose ACK have not been recived yet.
- This is the basis of sliding window protocols.
- Stop wait is a special case of sliding window where window size is 1.
- Also known as pipeline protocols.
- Two types:
  - Go Back N
  - Selective Repeat

# Go back N

- Transmitter is said to have a Send Window of size N.
- Requirements are:
  - Buffering at Transmitter:
    - Buffer N messages.
  - Sequence Numbers:
    - Sequence numbers can be 0 to Num. SeqNos - 1.
  - The Send Window:
    - Range of consecutive Num SeqNs - 1 values.
  - The Receive Window:
    - Tracks sequence number of the next message it expects and discards all other messages.
  - Acknowledgements:
    - On receipt of ACK, the sender can advance its send window to be the set of values ranging from ACK to (ACK+SeqNo-2)%SeqNo.
  - On Error:
    - The receiver either sends a NAK, or the transmitter times out.
    - The transmitter re-transmits the message and all subsequent messages. Hence Go-Back-N.

# Example


Assume:
Send window size is 3
Sequence numbers 0..3
Receiver remembers next message it expects
Receiver's ACK(n)/ NAK(n) means up to but excluding n.
Updates window based on ACK - re-sends remaining part of block
Sender
Send Window
Receiver
Receive Window
0-2
Data(0)
Data(1)
(lost)
Data(2)
(ignored)
Ack(1)
1-3
Data(1)
Data(2)
Data(3)
Ack(0)
0-2
0
1
Received packet 0.
Updates receive number
After lost packet all packets ignored because of packet number
1
Sends in ACK sequence number it expects
2
3
0

# Selective Repeat

- Rather than transmitting block of frames following first lost packet (as with Go Back N), better method is to simply retransmit only those packets which were lost (or in error).
- This is selective Repeat protocol.
- Must be able to receive packets out of sequence and be able to buffer these until the sequence can be restored - following repeated re-transmission.
- Receiver must have a Receive Window of X messages.

# Example

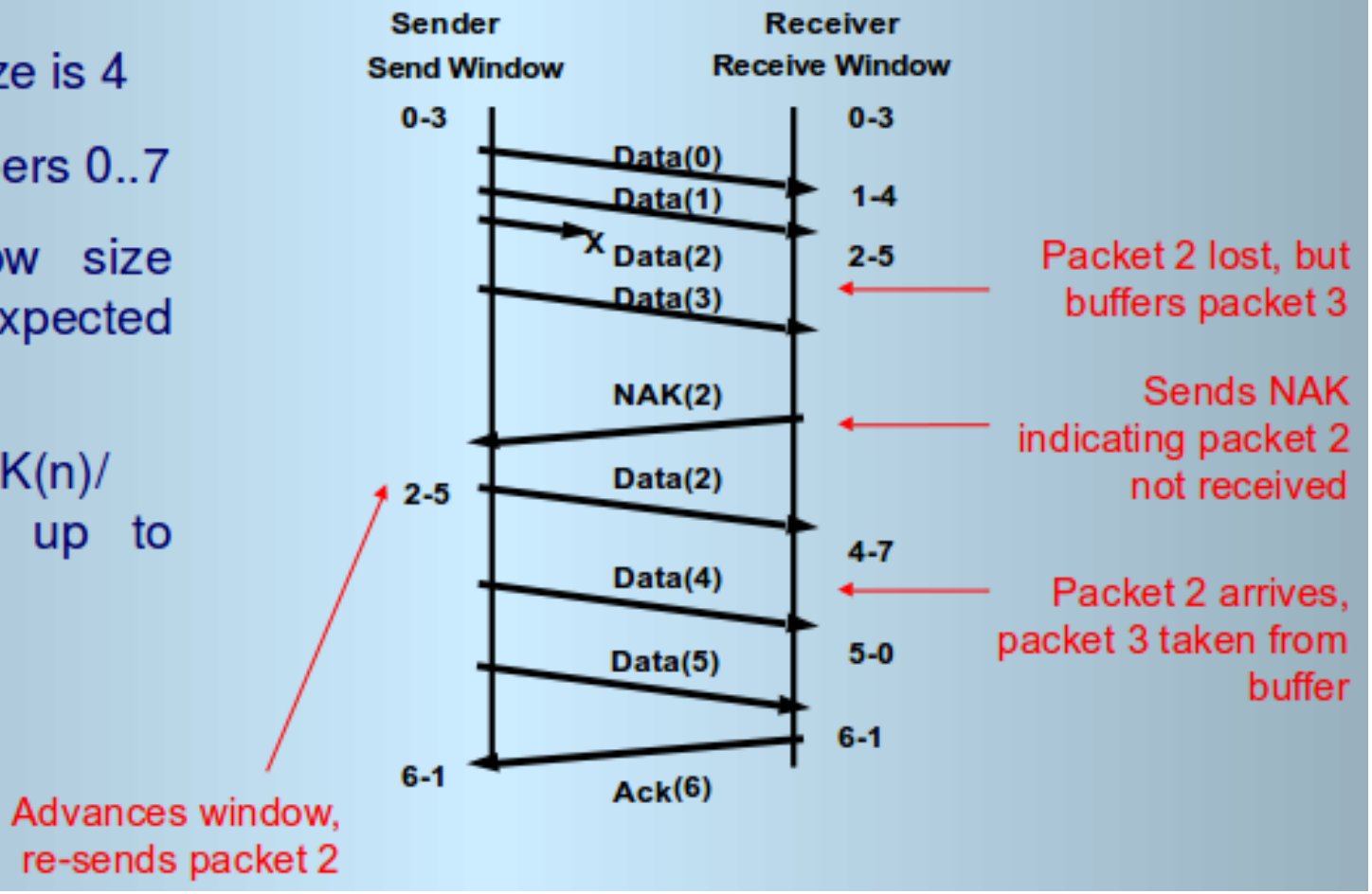
**Assume:**

Send window size is 4

Sequence numbers 0..7

Receiver window size of 4 of expected messages

Receiver's ACK(n)/ NAK(n) means up to but excluding *n*.



# Summary

- Simple Stop-Wait protocol is inefficient.
- Sliding Window (pipeline) protocols allow:
  - Transmitter to transmit sequences of messages.
  - Remove the delay waiting for individual ACKs.
- Go-Back-N is a simple technique which requires minimum resources at the receiver but involves sme unnecessary re-transmissions.
- Selective Repeat avoids unnecessary re-transmissions but requires additional buffering at the receiver.
- Both need additional buffering, multiple timers.

# The End