Open system Interconnection (OSI)

DataLink Layer

Munesh Singh

Indian Institute of Information Technology, Design and Manufacturing Kancheepuram,
Chennai, Tamil Nadu 600127

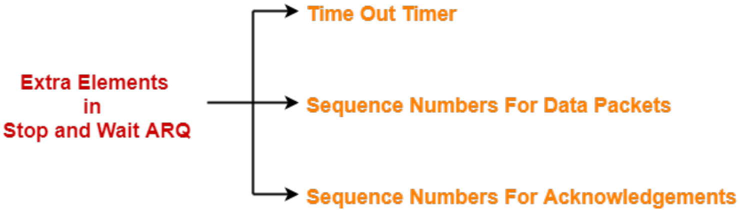
September 11, 2020

Flow Control Protocols

- Sliding Window Protocols
  - **Stop and Wait ARQ**
  - **Go Back N ARQ**
  - **Selective Repeat ARQ**
- Sender sends one data packet and then waits for its acknowledgment.
- Sender sends the next packet only after it receives the acknowledgment for the previous packet.
- The main problem faced by the Stop and Wait protocol is the occurrence of **deadlock** due to-
  - **Loss of data packet**
  - **Loss of acknowledgment**

Working of Stop and Wait ARQ

- Stop and Wait ARQ is an improved and modified version of Stop and Wait protocol.
- Stop and Wait ARQ assumes-
  - The communication channel is noisy.
  - Errors may get introduced in the data during the transmission.
- Stop and wait ARQ works similar to stop and wait protocol.
- It provides a solution to all the limitations of stop and wait protocol.
- Stop and wait ARQ includes the following three extra elements.



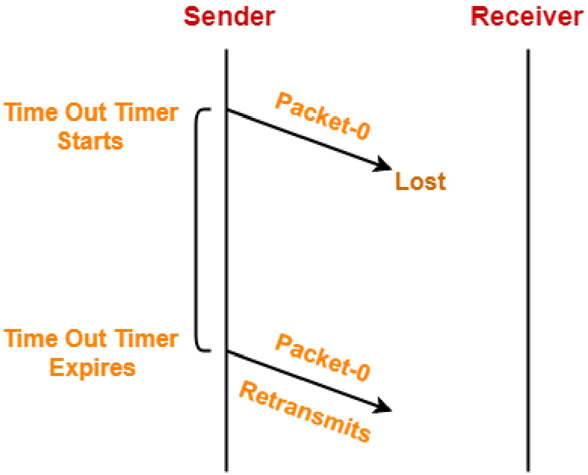
Stop and Wait ARQ

- Stop and Wait ARQ =
  Stop and Wait Protocol + Time Out Timer + Sequence Numbers for Data Packets and Acknowledgments
- **Number of Sequence Numbers Required-**
  - For any sliding window protocol to work without any problem, the following condition must be satisfied-
    Available Sequence Numbers >= Sender Window Size + Receiver Window Size
- **Stop and wait ARQ is a one bit sliding window protocol where-**
  - Sender window size = 1
  - Receiver window size = 1
- Thus, in stop and wait ARQ
  - Minimum number of sequence numbers required= Sender Window Size + Receiver Window Size

Stop and Wait ARQ

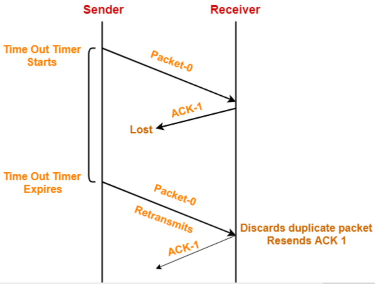
- Minimum number of sequence numbers required in Stop and Wait ARQ = 2.
  - The two sequence numbers used are 0 and 1.
- How Stop and Wait ARQ Solves All Problems?
  - **Problem of Lost Data Packet-**
    - Time out timer helps to solve the problem of lost data packet.
    - After sending a data packet to the receiver, sender starts the time out timer.
    - If the data packet gets acknowledged before the timer expires, sender stops the time out timer.
    - If the timer goes off before receiving the acknowledgment, sender retransmits the same data packet.
    - After retransmission, sender resets the timer.
    - This prevents the occurrence of deadlock.

Stop and Wait ARQ



Stop and Wait ARQ

- **Problem of Lost Acknowledgment-**
  - Sequence number on data packets help to solve the problem of delayed acknowledgment.
  - Consider the acknowledgment sent by the receiver gets lost.
  - Then, sender retransmits the same data packet after its timer goes off.
  - This prevents the occurrence of deadlock.
  - The sequence number on the data packet helps the receiver to identify the duplicate data packet.
  - Receiver discards the duplicate packet and re-sends the same acknowledgment.



Stop and Wait ARQ

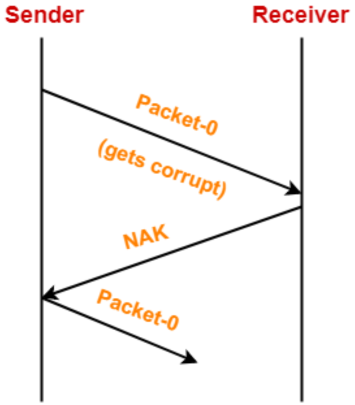
- Role of Sequence Number on Data Packets:
  - Sender sends a data packet with sequence number-0 to the receiver.
  - Receiver receives the data packet correctly.
  - Receiver now expects data packet with sequence number-1.
  - Receiver sends the acknowledgment ACK-1
  - Acknowledgment ACK-1 sent by the receiver gets lost on the way.
  - Sender receives no acknowledgment and time out occurs.
  - Sender retransmits the same data packet with sequence number-0.
  - This will be a duplicate packet for the receiver.
  - Receiver receives the data packet and discovers it is the duplicate packet.
  - It expects the data packet with sequence number-1 but receiving the data packet with sequence number-0.
  - It discards the duplicate data packet and re-sends acknowledgement ACK-1.
  - ACK-1 requests the sender to send a data packet with sequence number-1. This avoids the inconsistency of data.

Stop and Wait ARQ

- Role of Sequence Number on Acknowledgements
  - Two acknowledgments ACK1 reaches the sender.
  - When first acknowledgment ACK1 reaches the sender, sender sends the next data packet with sequence number 1.
  - When second acknowledgment ACK1 reaches the sender, sender rejects the duplicate acknowledgment.
  - This is because it has already sent the data packet with sequence number-1 and now sender expects the acknowledgment with sequence number 0 from the receiver.

Stop and Wait ARQ

- Problem of Damaged Packet:
  - If receiver receives a corrupted data packet from the sender, it sends a negative acknowledgment (NAK) to the sender.
  - NAK requests the sender to send the data packet again.



Stop and Wait Protocol Vs Stop and Wait ARQ

Stop and Wait Protocol

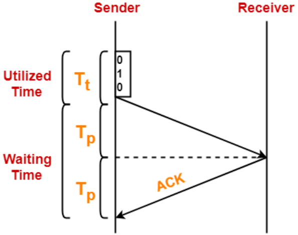
1. It assumes that the communication channel is perfect and noise free.
2. Data packet sent by the sender can never get corrupt.
3. There is no concept of negative acknowledgments.
4. There is no concept of time out timer.
5. There is no concept of sequence numbers.

Stop and Wait ARQ

1. It assumes that the communication channel is imperfect and noisy.
2. Data packet sent by the sender may get corrupt.
3. A negative acknowledgment is sent by the receiver if the data packet is found to be corrupt.
4. Sender starts the time out timer after sending the data packet.
5. Data packets and acknowledgments are numbered using sequence numbers.

Limitation of Stop and Wait ARQ

- The major limitation of Stop and Wait ARQ is its very less efficiency.
- To increase the efficiency, protocols like **Go back N** and **Selective Repeat** are used.
- **Explanation**
  - Sender window size is 1.
  - This allows the sender to keep only one frame unacknowledged.
  - So, sender sends one frame and then waits until the sent frame gets acknowledged.
  - After receiving the acknowledgment from the receiver, sender sends the next frame.



Stop and Wait ARQ

- Sender uses T_t time for transmitting the packet over the link.
- Then, sender waits for 2 x T_p time.
- After 2 x T_p time, sender receives the acknowledgment for the sent frame from the receiver.
- Then, sender sends the next frame.
- This 2 x T_p waiting time is the actual cause of less efficiency.
- **Efficiency Improvement**
  - The efficiency of stop and wait ARQ can be improved by increasing the window size.
  - This allows the sender to keep more than one unacknowledged frame in its window.
  - Thus, sender can send frames in the waiting time too.
  - This gives rise to the concept of sliding window protocols.

Thank You