Cairo University **Computer Networks1**
Faculty of Engineering **Fall 2020**

Sheet3

Unrestricted Simplex Protocol

1. In the "Unrestricted Simplex Protocol" why we don't need to initialize the frame header ?

   Because we don't need to add a sequence number to the frame as no frames get lost or damaged.

2. In the "Unrestricted Simplex Protocol" why we don't need to set a timer at the sender or at the receiver?

   We don't need a timer at the receiver because the channel is noise free, so no frames get lost or damaged so the receiver will always receive the sent messages and the sender will always send as the network layer has infinite number of packets to send.

   We don't need a timer at the sender as the receiver sends nothing.

Simplex Stop and Wait Protocol

3. In the "Simplex Stop and Wait Protocol" why we don't need to set a sequence number to the sent frames?

   Because only one frame is sent at a time , so no confusion happens between frames that need to be identified by sequence numbers.

4. In the "Simplex Stop and Wait Protocol" why we send a dummy empty ack i.e why we don't set an ack number ?

   Because we ack on one frame only at a time , no multiple acks can get confused about such that we need to use sequence number for acks.

5. In the "Simplex Stop and Wait Protocol" why we don't need to set a timer at the sender?

   Because the channel is error free, so that nether frames nor acks can be lost or damaged.

6. In the "Simplex Stop and Wait Protocol" why we don't need to set a timer at the receiver?

Because the channel is error free, so that nether frames nor acks can be lost or damaged.

Simplex Protocol for a noisy channel

7. In the "Simplex Protocol for a noisy channel" why we need to set a sequence number to the sent frames? / why we use one bit only for the sequence number?

   As the channel is noisy , while sending a frame, a late frame from the previous sending round could reach at the receiver , so the receiver needs to check in a window of two frames for the correct frame.

8. In the "Simplex Protocol for a noisy channel" why we need to set an ack number to the sent acks?

   As the channel is noisy , while sending an Ack, a late Ack from the previous sending round could reach at the sender , so the receiver needs to check in a window of two frames for the correct Acked frame.

9. In the "Simplex Protocol for a noisy channel" why we need to set a timer at the sender?

   As the channel is noisy , the frame or it's corresponding Ack can get lost , so the sender has to set a timer to prevent deadlocks "i.e waiting forever from both sides the sender and the receiver".

10. In the "Simplex Protocol for a noisy channel" why we may need to set a timer at the receiver?

    If the timer is at the receiver side, and as the channel is noisy , the frame or it's corresponding Ack can get lost. And the sender only will send the next frame if it receives an Ack for the previous one , so the receiver has to set a timer to prevent deadlocks "i.e waiting forever from both sides the sender and the receiver".

Link Capacity

11. Assuming a digital data transmission system between Cairo and Alexandria (round trip delay is about 250 µs) using a T1-line (1500 kBps), and a frame size of 512 Bytes. The overhead can be neglected. What is the total throughput and efficiency using a stop and wait ARQ protocol?

Rules:

η = Efficiency

T_t =Transmission time

T_p= propagation delay

Th= Throughput

B= Bandwidth

D= distance

V= velocity

L= Frame length

$$T_t = \frac{L}{B}$$

$$\eta = \frac{useful\ time}{total\ time} = \frac{T_t}{T_t+2T_p}$$

$$Th=\frac{actual\ transmission\ bits}{time} = \frac{L}{T_t+2T_p} = \frac{B.T_t}{T_t+2T_p} = \eta * B$$

Note that KBps = kilo byte per second ,and Kbps = Kilo bit per second

Note that: round trip delay =2 T_p

Solution:

2T_p=250 micro second // round trip delay =2 T_p

B= 1500 kilo Byte per second

L=512 Byte

$$T_t = \frac{L}{B} = \frac{512}{1500 * 10^3} = 0.34133\ millisecond$$

$$\eta = \frac{T_t}{T_t+2T_p} = \frac{0.34133}{0.34133+0.25} = 0.58$$

$$Th = \eta * B = 0.58 * 1500 = 870\ KBps$$

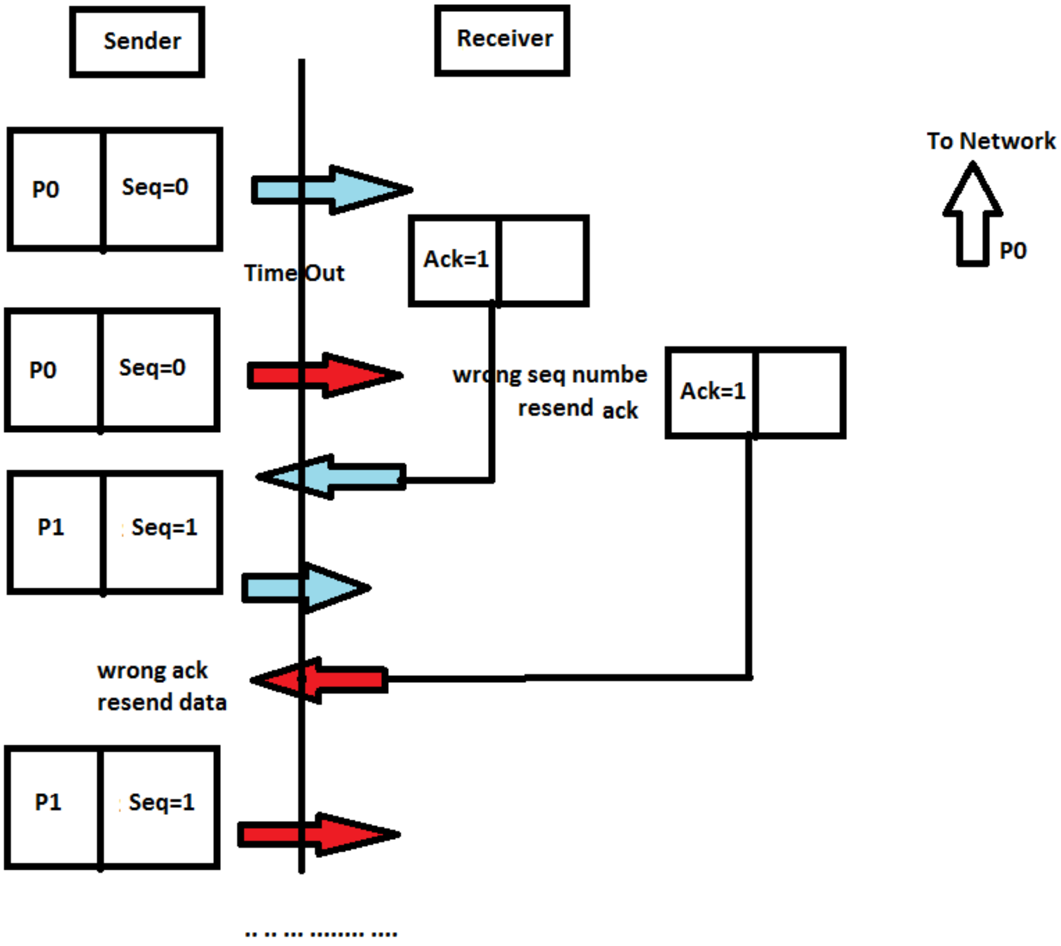
Design problems

12. In the “Simplex Protocol for a noisy channel” What will happen if the timeout at the sender is smaller than the round-trip delay (even temporarily when sending one packet only)?

I.e (Timer < 2 T_p)

In this case every frame and every Ack will be sent twice and they will live inefficiently ever after :D .

Illustration:



13. In the "Simplex Protocol for a noisy channel" Why does the sender re send the last packet it sent when it gets duplicate ACK (I.e. what assumption is made)?
    It assumes whether the previous frame was lost and then the receiver timed out, or the Ack is modified by the noisy channel so that the number is modified or the Ack is duplicate of a last Ack or the Ack is a delayed old Ack. In all cases it cannot proceed unless it makes sure that the receiver is well synchronized with the sender.
14. In the "Unrestricted Simplex Protocol" assume that the only error that can occur is packet duplication on the reverse channel. That is, a single packet

sent by the receiver may be duplicated and arrive as two consecutive identical packets at the sender . what are the minimum changes needed for the protocol to be fully functionable?

No change is needed as no data is sent through the reverse channel (from the receiver to the sender).