Mr. Ali bin Tahir Email: ali@biit.edu.pk

Week # 13 – Lecture # 25 – Transport layer (Layer 4)

Recommended Reading:

Book: Computer Networking, A Top-Down Approach 6th edition, Authors: James F. Kurose, K W. Ross

Transport Layer

In our last lecture, we have introduced the service of reliable data delivery which is offered by TCP. We have discussed that packet drop and delays are quite common in IP networks and therefore TCP provides a mechanism of reliable data delivery. **The reliable data delivery is provided through packets retransmissions which are lost in the network**. We have also studied how ACKs and NACKs are used in this regard.

We have introduced different reliable data delivery protocols such as stop and wait. In today's lecture we will continue our discussion on stop and wait protocol. However, we will see stop and wait is a basic protocol with performance limitation. We will study more advanced protocols as well for reliable data transfer.

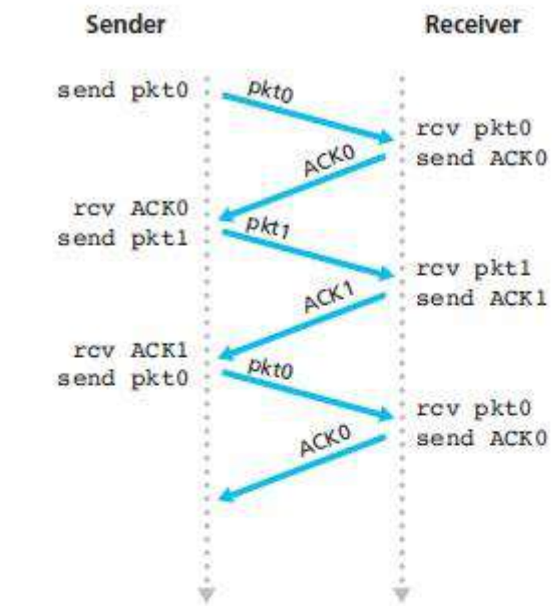
Let us start our today's lecture with reliable data transfer protocols. Here is the lecture outline:

- **Stop and Wait Protocol**
- **Pipelining and Sliding window Protocols**
  - **Go-Back-N Protocol**
  - **Selective Repeat**

Stop and Wait Protocol

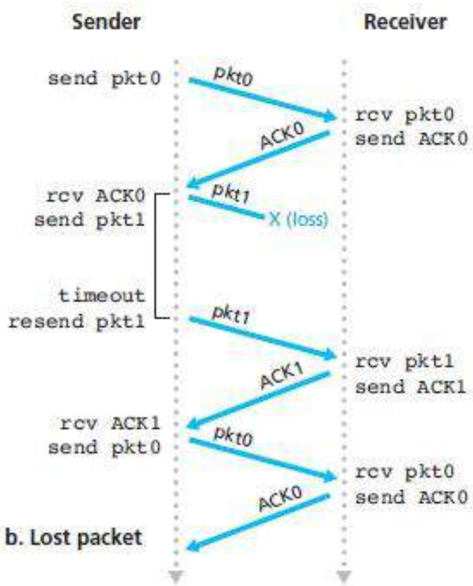
In this protocol, sender use 1 bit sequence number with every packet. There are two possibilities with a one-bit sequence number (bit will be either 0 or 1), so there will be a packet#0 ad packet#1. Sender transmits a packet and then stops and waits for the acknowledgment (positive) from the receiver side. The next packet is only transmitted once the acknowledgment is received for the earlier sent packet. The protocol pretty much works in the same way as depicted in its name (i.e. stop and wait).

Mr. Ali bin Tahir Email: ali@biit.edu.pk



a. Operation with no loss

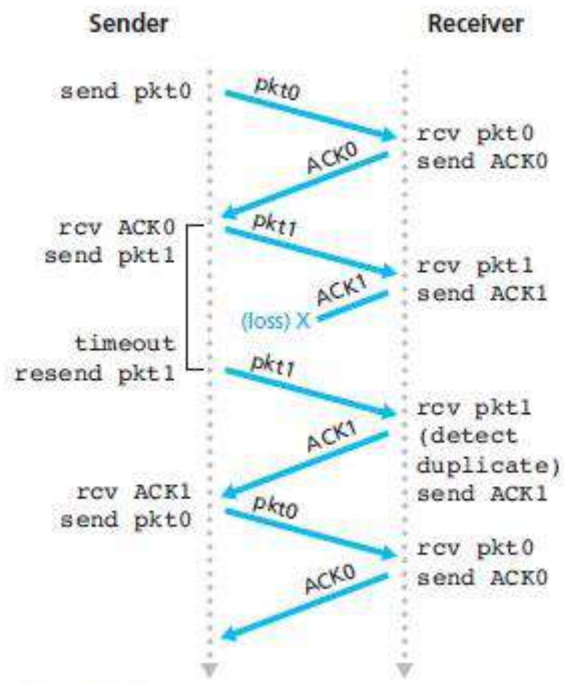
In our last lecture we have seen a simple example scenario of stop and wait protocol, when there is no packet drop in the network (shown in figure above). In today's lecture we will see more scenarios with packet drop.



Consider a packet drop scenario by modifying the same scenario (a) a little bit. Suppose pkt1 sent by the sender drops in the network and never reaches to its destination. So, sender will never receive an ACK from receiver. Then how this communication moves forward since sender is in a waiting state. To fix this problem, sender waits for ACK for a specific amount of time. When that time period expires and ACK is not received by the sender, sender declares a timeout and resend pkt1 (by

Mr. Ali bin Tahir
Email: ali@biit.edu.pk

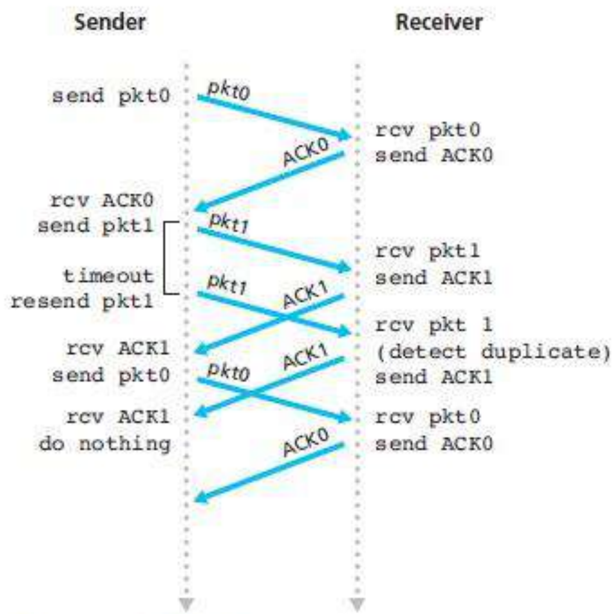
assuming packet is lost). The scenario is shown in figure b. In second attempt, pkt1 is transmitted and ACK1 received by the sender. After that, everything works in the regular fashion.



c. Lost ACK

Now suppose another scenario where an ACK is lost in the network. It can be shown in figure c that ACK1 is lost in the network and never reaches to the sender. In this case, sender behaviour will be the same as discussed in scenario b. Since sender will not receive an ACK and timeout will occur. Sender resends the pkt1. But packet is already received at the receiver. When receiver receives pkt1 again, it detects it as a duplicate reception (discard pkt1 since it is already received). The receiver resends ACK1 to notify sender that pkt1 is received and there is no need to resend. After that, everything works fine in the similar fashion as described for the earlier scenarios.

Mr. Ali bin Tahir Email: ali@biit.edu.pk



d. Premature timeout

Suppose a scenario d where ACK1 is not lost but it is actually delayed in the network. Due to this delay, timeout occurs on the sender side and sender resend pkt1. After some time, delayed ACK1 is received by the sender. As soon as an ACK is received by the sender, sender transmits the next packet, as we notice send pkt0 (second round) on the sender side. On the receiver side, duplicate pkt1 will be received and discarded in the similar fashion as described for scenario c. ACK1 will be resent by the receiver to notify that pkt1 is received. Since ACK1 (delayed one we know) is already received by the sender, what a sender will do when a duplicate ACK1 will be received? The sender will do nothing. The reason is that response of ACK1 is to send pkt0. But the pkt0 is already transmitted by the sender when ACK1 was received earlier. So, no further action needed for a duplicate ACK1.

Pipelining Protocols (or Sliding Window Flow Control)

The stop and wait allows transmission of only one packet at a time. So, it limits the sender transmission rate and becomes the performance bottleneck. The solution to this particular performance problem is simple: Rather than operate in a stop-and-wait manner, the sender is allowed to send multiple packets without waiting for acknowledgments. Since there are many in-transit sender-to-receiver packets filling a pipeline, this technique is known as pipelining. Pipelining has the following consequences for reliable data transfer protocols:

• The range of sequence numbers must be increased, since each in-transit packet must have a unique sequence number and there may be multiple, in-transit, unacknowledged packets.

• The sender and receiver sides of the protocols may have to buffer more than one packet. Minimally, the sender will have to buffer packets that have been transmitted but not yet acknowledged. Buffering of correctly received packets may also be needed at the receiver.

Two basic approaches of pipelining protocols are:

- **Go-Back-N (GBN)**
- **Selective Repeat**

Go-Back-N (GBN)

In a Go-Back-N (GBN) protocol, the sender is allowed to transmit multiple packets (when available) without waiting for an acknowledgment, but is constrained to have no more than some maximum allowable number, *N,* of unacknowledged packets in the pipeline. So, after sending N packets, sender waits for the ACK. The value of N is known as window size.

GBN on Sender side

Suppose window size N = 4, it means sender will send:

Pkt0

Pkt1

Pkt2

Pkt3

Wait for ACK

Similarly suppose another case when N = 3,

Pkt0

Pkt1

Pkt2

Wait for ACK

We notice multiple packet transmissions are now allowed and sequence number is now more than 1 bit. For N = 4, we need a two-bit sequence number (pkt 00, 01, 10, 11). Indeed, if we increase the window size, we will need more bits in sequence number.

For N = 10, How many bits are needed for sequence number?

Sliding window

Now we understand what we mean by the window size (number of packets which can be transmitted unacknowledged). Suppose N=4, the size of window can visualise as shown below:

Pkt Pkt Pkt Pkt   Pkt Pkt Pkt .....

Window

Sender Waits for ACK after sending four packets. Although there more packets available for transmission but they are outside the range of window and therefore not transmitted at the moment.

Now suppose the ACK is received for the first transmitted packet, the window will move forward (slide) and next packet (which is available and waiting for transmission) will be sent.

Pkt   Pkt Pkt Pkt Pkt   Pkt Pkt .....

Window

In this way window will continue to slide whenever an ACK will be received. Due to this behaviour, GBN is known as a sliding window protocol.

GBN on receiver side

On the receiver side, packets are only accepted when they are received in order. When a packet is received out of order, it is discarded.

Example

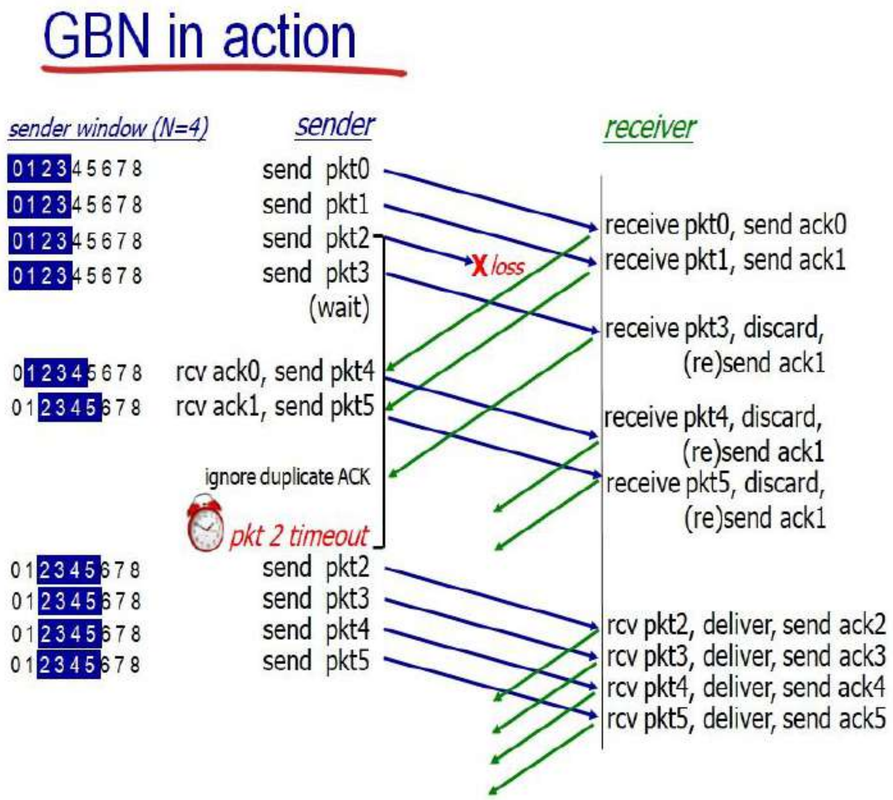
Suppose sender sends pkt0, pkt1, pkt2 and pkt3

Pkt0 and pkt1 received in order correctly by the receiver.

Pkt2 is dropped in the network

When pkt3 will be received by the receiver, it will be discarded. Since, receiver is expecting pkt2 which is next packet in order after pkt1. So pkt3 will be taken as out of order.

Now we will see a complete example of GBN Protocol with a window size of 4



You can notice pkt2 is lost in network and therefore pkt3 is discarded by the receiver. However, pkt0 and pkt1 are received correctly and therefore window is slided and pkt4 and pkt5 are transmitted by the sender (window range 2-5). Since out of order packets are not acceptable by the receiver, these packets are also discarded by the receiver and ACK1 is resent which is the last in ordered received packet. Later, when there is a timeout for pkt2 at the sender side, sender retransmits not only pkt2 but all other packets which are in the range of window. Therefore pkt3, pkt4 and pkt5 are also retransmitted. Since sender go back to the start of the window and resend all N packets, this protocol is known as GO-Back-N.

Mr. Ali bin Tahir Email: ali@biit.edu.pk

Week # 13 – Lecture # 26 – Transport layer (Layer 4)

Recommended Reading:

Book: Computer Networking, A Top-Down Approach 6th edition, Authors: James F. Kurose, K W. Ross

Transport Layer

In previous lectures we have covered two different protocols which offer the service of reliable data transfer. These protocols are stop and wait and Go-Back-N. In this lecture we will cover yet another protocol which is called selective repeat.

Here is the outline for today's lecture:

- **Selective Repeat**
- **Example Scenario**

Selective Repeat

We notice for Go-Back-N that sender retransmits all packets which are not received in order by the receiver. Retransmission means more network resources and bandwidth will be consumed. This is of course the down side of this protocol. However, it is quite simple on the receiver side since receiver only accepts packets in order and does not maintain any sliding window as we see on the sender side.

What if we allow out of order packet delivery on the receiver side? Then sender will only transmit missing packet and other out of order received packets will be accepted by the receiver. Since sender will not retransmit out of order packets, this approach will consume less bandwidth as compared to Go-Back-N protocol. **This approach is known as selective repeat protocol. We only select dropped packet for retransmission**.

The selective repeat is better in terms of bandwidth consumption. However, on the other hand, now we need to maintain a sliding window on the receiver side as well. The receiver must keep track of packets, which packets are received and which packets are missing since out of order packet is now accepted. Now the receiver side is more complicated than Go-Back-N. Let us take an example to understand this concept:

Example:

Receiver Side (Go-Back-N)

Pkt0 Pkt1 received

Pkt2 dropped in the network, never reached at the receiver

Pkt3 received but discarded by the receiver, receiver will only accept a packet with sequence number 2. A packet with any other sequence number will be discarded (simple implementation, there is no need to record any information regarding packets)

Receiver Side (Selective Repeat)

Pkt0 Pkt1 received

Pkt2 dropped in the network, never reached at the receiver

Pkt3 received and it will be stored at the receiver. The receiver will maintain the information that following packets are received:

Pkt0, pkt1 pkt3

Missing packets

Pkt2

The receiver will hold pkt3 in receiver socket buffer but it will not be delivered to the application process until pkt2 is missing. Once pkt2 is received, then in ordered packets are available to the application process.

Selective Repeat Scenario

Now let us take a complete scenario to understand the functionality of the protocol.

In figure 3.26, we are considering the same scenario which we used for Go-Back-N. The only difference is that now we use selective repeat protocol instead of Go-Back-N. It can be noticed that out of order packets (pkt3, 4 and 5) are now accepted and stored by the receiver. Sender only retransmits dropped packet pkt2 and pkt3, 4 and 5 are not transmitted again (in contrast to Go-Back-N). You can also notice a sliding window on the receiver side as well. The size of window is 4 both on the sender and receiver side. It is also important to notice that when ACK3 is received by the sender side, it does not send anything as a next step. It is because sender cannot slide its window since pkt2 is in front of window. The ACK2 is not received yet and therefore it is not possible to slide window (further towards right). Pkt3, 4 and 5 are already send and pkt6 is outside window, therefore sender has nothing to send any packet when ACK3 is received.

Mr. Ali bin Tahir
Email: ali@biit.edu.pk

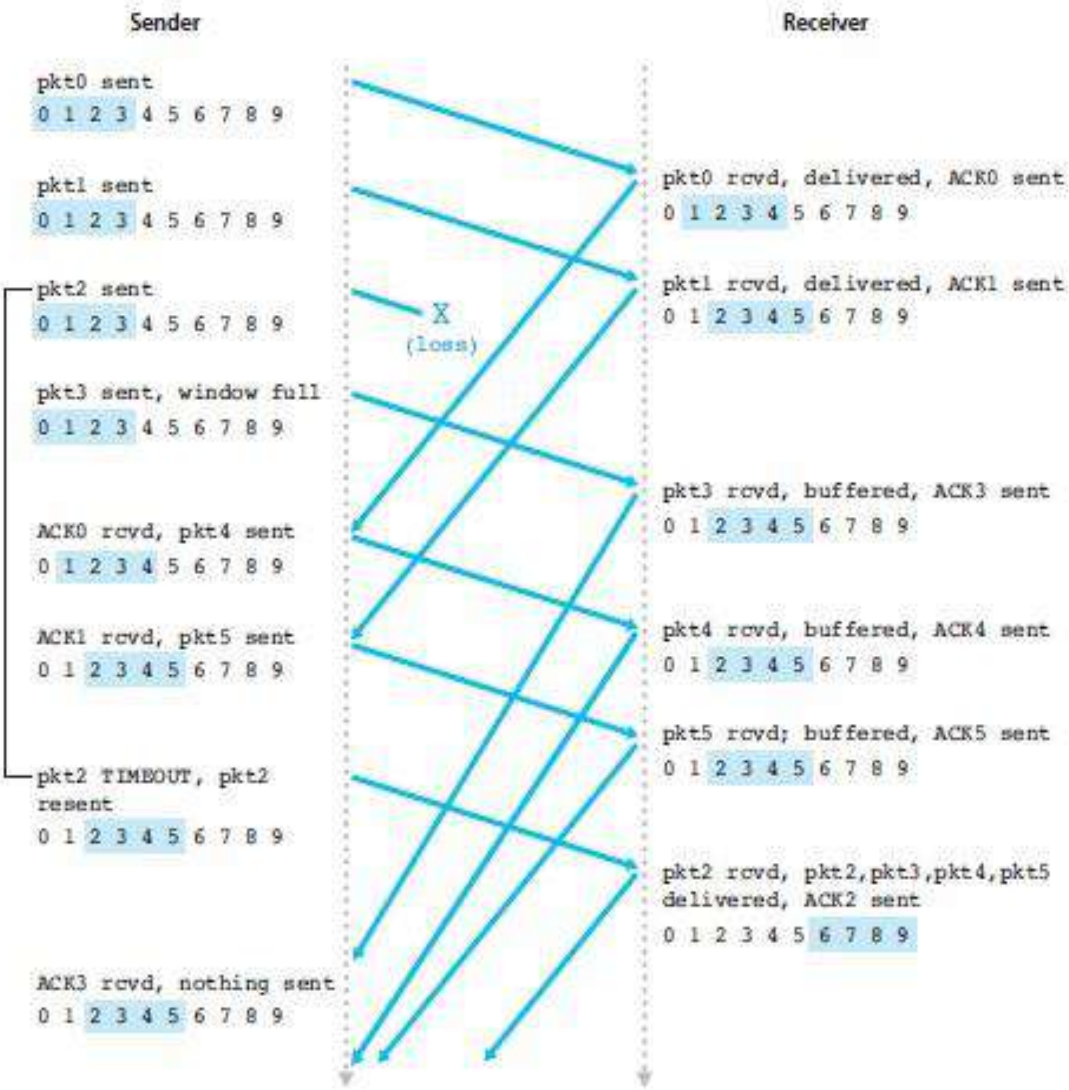


Figure 3.26 ♦ SR operation