**Positive Acknowledge with Retransmission (PAR) & TCP**

To understand (PAR) and TCP, we will get to know few concepts or characteristics of TCP.

1. Stream Orientation is when large volumes of data are transfer from the user to the receiver. For example, transmitting a movie from the sender to the host. TCP sends a stream of bytes in small chunks, keeping track of the stream on how far it is into the movie.
2. Virtual Circuit Connection is the notion of an agreement to connection between the sender and receiver application programs. Once the connection has been stablished, then the transfer can begin.
3. Buffered transfer, sends a data stream across the virtual circuit, holding the data until confirmation has been send back of delivery, otherwise the same data stream is send again.
4. Unstructured Stream is a message that contains a part of a stream.
5. Full duplex connection is the flexibility of the of the host being the sender and the sender being the host once a connection has been stablished.

Positive acknowledgment with retransmission is the events of transmitting back an ack of receiving data to the sender for each message sent to the host. Retransmitting data back is known as an ack. This positive acknowledgment transmission is necessary because for every message sent by TCP, it uses IP to send messages, which the delivery is not guaranteed. The sender waits until an acknowledgment is received to send the following message, if the acknowledgment is not received then the message is resent until the acknowledgement is received to the sender. Another scenario of the sender retransmitting the same message is if the acknowledgment message gets lost in the network. The receiver will get the same message twice and basically send the acknowledgment back to the sender twice. Each connection that TCP manage tracks a time which it is called a roundtrip time. The retransmission time is usually about the double of the roundtrip time to guarantee delivery in case of heavy traffic in the network.

The fixed window allows to send certain number of messages, the messages are sent based on few factors, the application may have only space to send few messages and the window sizes may be set bigger from what the application can send messages when the connection is being stablished. The window size can change, which we will get into it later on. However, an example of this situation could be if the window size is 8, but the application can only send 2 messages, then two messages are sent at the time. However, for the window to slide over, the first few messages must be acknowledged, for example, if message 3 has been ack but message 1 has not, then the window cannot be slide over to the right side until message 1 get acknowledged. The window can slide over to the right only if all the messages have been ack from the left side of the window.

Another example to understand this concept is that the window size is set to 8, then the messages are sent in small groups. However, once the messages have been received and ack has been sent back from messages 3 to 5 at this point the window cannot slide over to the right yet, because message 1 has not been ack. If message 1 gets acknowledged, then the window slide over one position only, because message 2 has not been ack yet. Once message 2 get an ack then, the window slides over to the right to message 5. There is an important factor to consider on how the sliding window works, the window will not slide over to the right until messages has been acknowledged from the left side of the window, then window can slide over to the right. Another important point is that positive acknowledgement with retransmission has the capability to handle multiple messages at one time.

On the other hand, because TCP is a transport layer, it will handle the messages in stream of bytes, each message is being consider as a byte. Because of it, TCP will send the messages based on the application availability to receive bytes. For example, if the application request to be sent three bytes then, TCP will send three messages, remember that messages are handle as bytes in TCP. TCP does not keep track of the bytes individually sent to the application, instead TCP will keep track of it as a sequence of bytes for each message sent to the application. TCP check for ack messages by going throughout the buffer and checking for the missing bytes in the buffer that has not been ack. TCP send back which ack is missing in the sequence of bytes.

Going back to the example earlier the sender sent three messages, each message has a sequence of bytes, message 1 has a sequence of three bytes, message 2 a sequence of three bytes and message 3 a sequence of two bytes. The sender will check for the missing ack in the sequence and send the missing ack byte from the sequence that it is missing. Then, the sender will retransmit byte #4 in the sequence from message 2 because that is the byte that the ack is missing.

The window size is set when a full duplex connection is being stablished between the two applications. However, a half-duplex connection could be stablished as well, for this type of connection the sender can only send, and the receiver can only receive. For each ack send, a window advert is sent along with it. The window size can be change at any moment based on the space available in the buffer. When the message is sent, the window size, the sequence, and the missing byte numbers are encoded into the TCP header in the message. The retransmitting data window size is half from the original window size, this could occur due to loss of data in the network. However, if data stops from getting loss in the network, the window will increase one byte at a time when retransmitting. This window is known as the effective window size, the effective window size could go back to the size it starts at, half from the original window size.

Another advantage about TCP when retransmitting is that, TCP will not always retransmit missing ack, which it is the missing bytes that TCP retransmit that does not have an ack, because each ack carries with it, the ack that it was acknowledged before. TCP only will only retransmit a missing ack if the following ack is not carrying the ack that it was acknowledged before.