Aim: Stop and wait protocol/sliding window(selective repeat/GO BACK N).ns2/netsim.

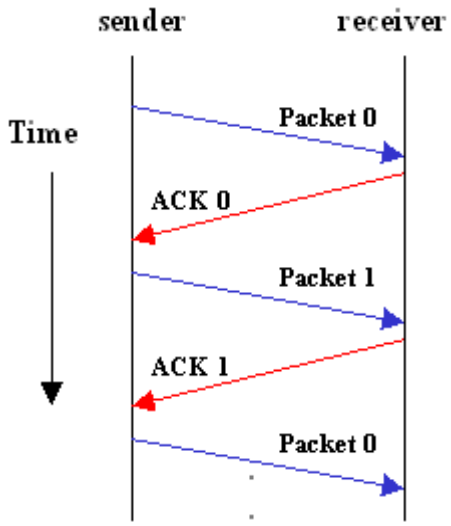
Theory:

1. Background

*"**stop-n-wait**"* (sometimes known as *"positive acknowledgement with retransmission"*) is the fundamental technique to provide reliable transfer under unreliable packet delivery system.

2.How this protocol works..

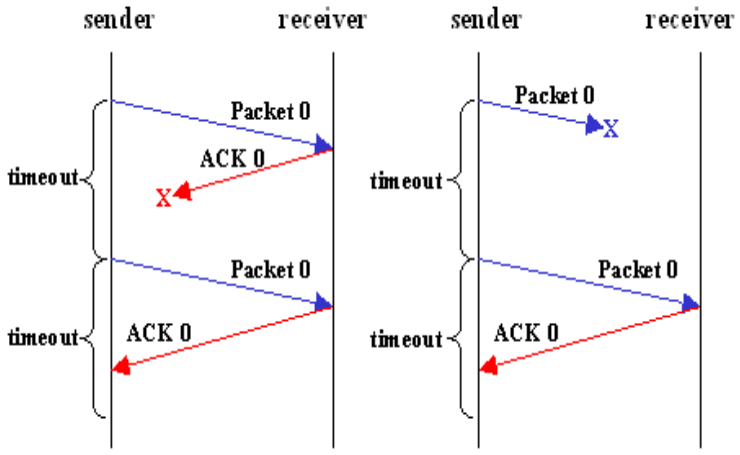
1) Normal operation



After transmitting one packet, the sender waits for an ***acknowledgment (ACK)*** from the receiver before transmitting the next one. In this way, the sender can recognize that the previous packet is transmitted successfuly and we could say "stop-n-wait" guarantees reliable transfer between nodes.

To support this feature, the sender keeps a record of each packet it sends.
Also, to avoid confusion caused by delayed or duplicated ACKs, "stop-n-wait" sends each packets with unique sequence numbers and receives that numbers in each ACKs.

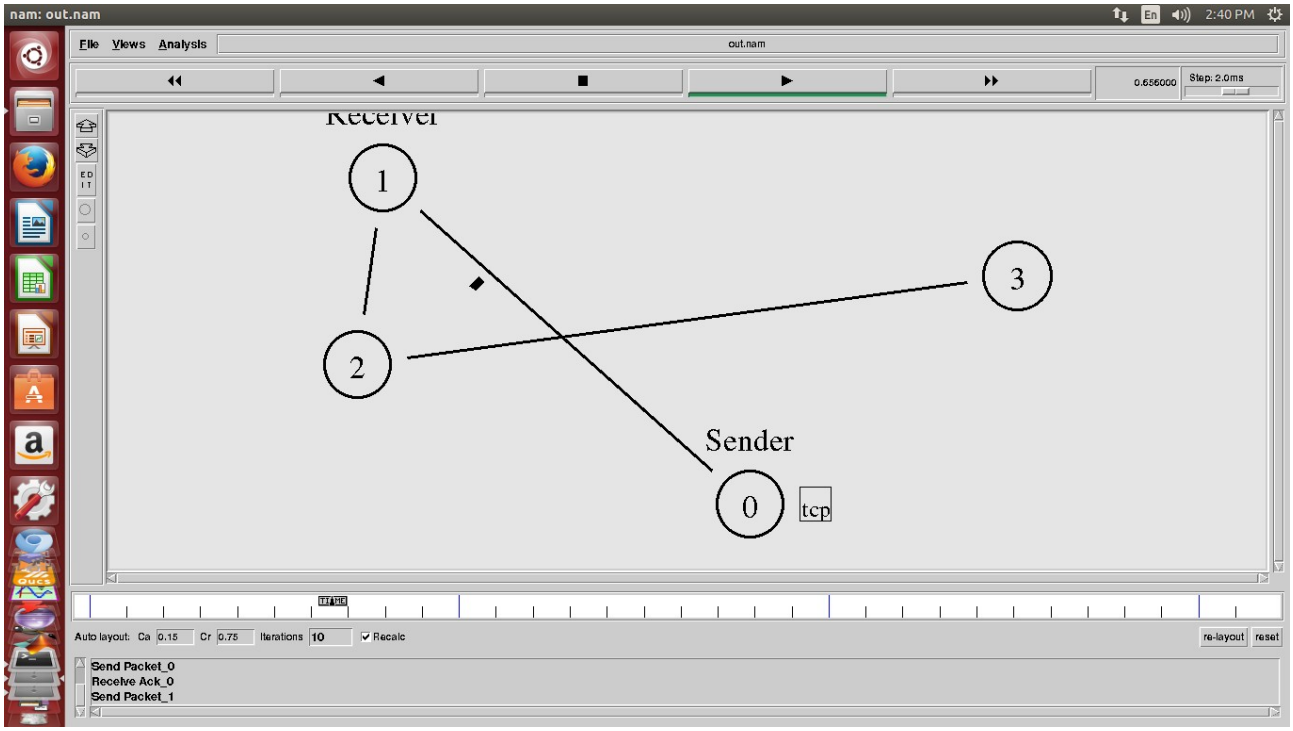
2) Timeout

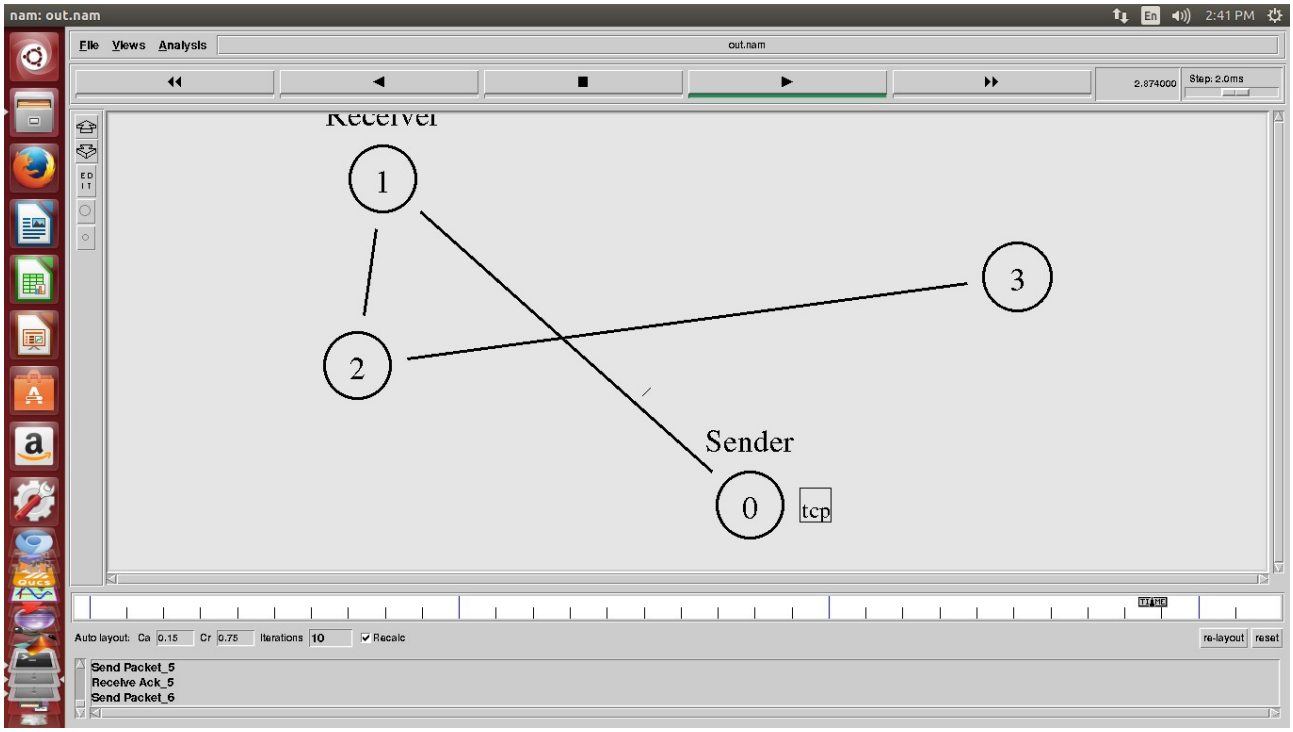


If the sender doesn't receive ACK for previous sent packet after a certain period of time, the sender *times out* and *retransmits* that packet again. There are two cases when the sender doesn't receive ACK; One is when the ACK is lost and the other is when the frame itself is not transmitted

3. How it is shown in nam (network animator)..

"stop-n-wait" protocol can be shown as below in nam.





4. Shortcoming

The main shortcoming of the stop-and-wait algorithm is that it allows the sender to have only one outstanding frame on the link at a time. The sender should wait till it gets an ACK of previous frame before it sends next frame. As a result, it wastes a substantial amount of network bandwidth. To improve efficiency while providing reliability, "sliding window" protocol is appeared.

CODE:

```
set ns [new Simulator]
#$ns color 1 Blue
#$ns color 2 Red
set namfile [open out.nam w]
$ns namtrace-all $namfile
proc finish {} {
global ns namfile
$ns flush-trace
close $namfile
exec nam out.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n0 $n1 2Mb 200ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 10ms DropTail
$ns queue-limit $n0 $n1 15
Agent/TCP set nam_tracevar_ true
set tcp [new Agent/TCP]
$tcp set window_ 1
$tcp set maxcwnd_ 1
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns add-agent-trace $tcp tcp
#$ns monitor-agent-trace $tcp
```

```
$tcp tracevar cwnd_
$ns at 0.1 "$ftp start"
$ns at 3.0 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n1 $sink"
$ns at 3.5 "finish"
$ns at 0.0 "$ns trace-annotate \"Stop and Wait with normal operation\""
$ns at 0.05 "$ns trace-annotate \"FTP starts at 0.1\""
$ns at 0.11 "$ns trace-annotate \"Send Packet_0\""
$ns at 0.35 "$ns trace-annotate \"Receive Ack_0\""
$ns at 0.56 "$ns trace-annotate \"Send Packet_1\""
$ns at 0.79 "$ns trace-annotate \"Receive Ack_1\""
$ns at 0.99 "$ns trace-annotate \"Send Packet_2\""
$ns at 1.23 "$ns trace-annotate \"Receive Ack_2  \""
$ns at 1.43 "$ns trace-annotate \"Send Packet_3\""
$ns at 1.67 "$ns trace-annotate \"Receive Ack_3\""
$ns at 1.88 "$ns trace-annotate \"Send Packet_4\""
$ns at 2.11 "$ns trace-annotate \"Receive Ack_4\""
$ns at 2.32 "$ns trace-annotate \"Send Packet_5\""
$ns at 2.55 "$ns trace-annotate \"Receive Ack_5    \""
$ns at 2.75 "$ns trace-annotate \"Send Packet_6\""
$ns at 2.99 "$ns trace-annotate \"Receive Ack_6\""
$ns at 3.1 "$ns trace-annotate \"FTP stops\""
$ns at 0.0 "$n0 label Sender"
$ns at 0.0 "$n1 label Receiver"
#$ns at 0.00 "$cbr start"
#$ns at 3.5 "$cbr stop"
#$ns at 4.0 "finish"
$ns run
```