# 118A1084

# EXPERIMENT NO. 7

**AIM:** Implement Stop and wait protocol using NS2.

**THEORY:**

**Stop-and-wait** is a method used in telecommunications to send information between two connected devices. It ensures that information is not lost due to dropped packets and that packets are received in the correct order. It is the simplest kind of automatic repeat-request (ARQ) method. A stop-and-wait ARQ sender sends one frame at a time; it is a special case of the general sliding window protocol with both transmit and receive window sizes equal to 1. After sending each frame, the sender doesn't send any further frames until it receives an acknowledgement (ACK) signal. After receiving a good frame, the receiver sends an ACK. If the ACK does not reach the sender before a certain time, known as the timeout, the sender sends the same frame again.

**PROGRAM:**

**NS2 Code:**

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

#stop and wait protocol in normal situation   
# features : labeling, annotation, nam-graph, and window size monitoring

set ns [new Simulator]

set n0 [$ns node]   
set n1 [$ns node]

$ns at 0.0 "$n0 label Sender"   
$ns at 0.0 "$n1 label Receiver"

set nf [open A1-stop-n-wait.nam w]   
$ns namtrace-all $nf   
set f [open A1-stop-n-wait.tr w]   
$ns trace-all $f

$ns duplex-link $n0 $n1 0.2Mb 200ms DropTail   
$ns duplex-link-op $n0 $n1 orient right   
$ns queue-limit $n0 $n1 10

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\""

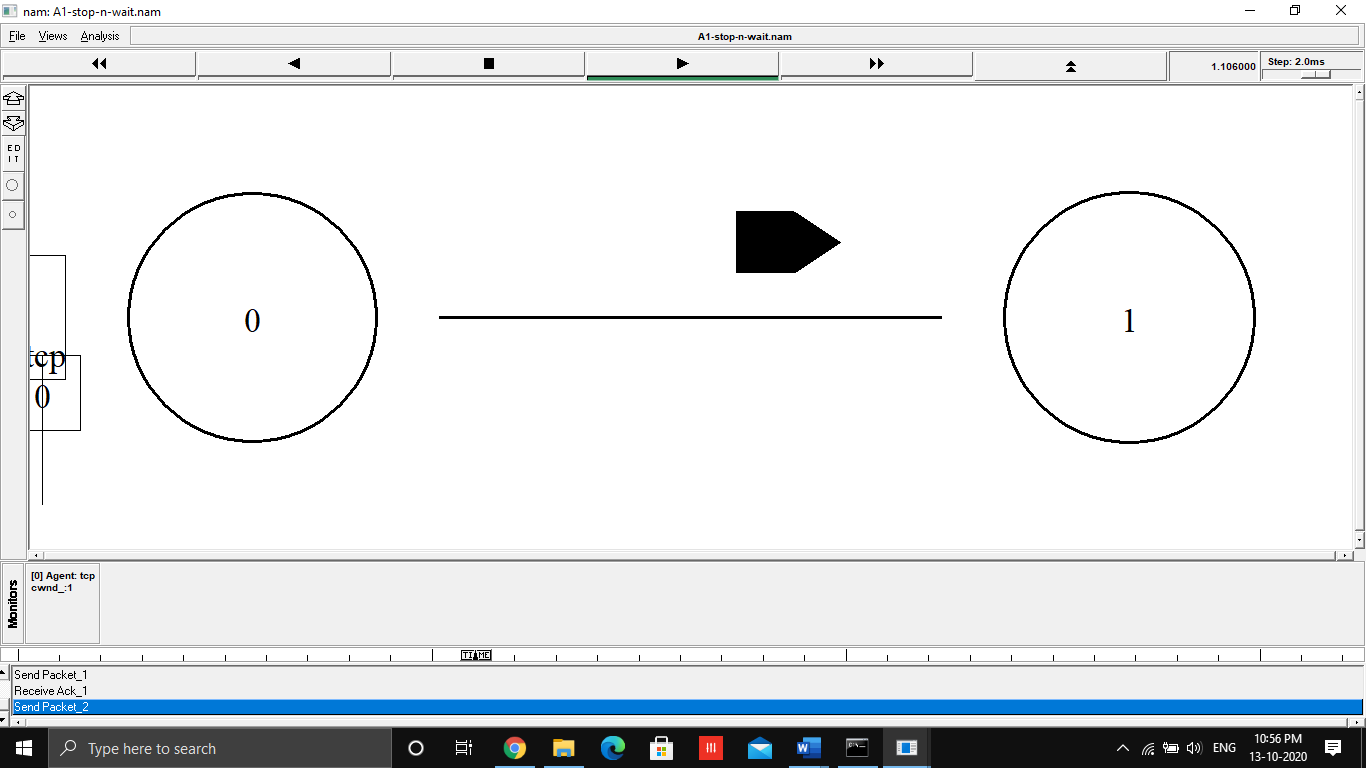
$ns at 3.1 "$ns trace-annotate \"FTP stops\""

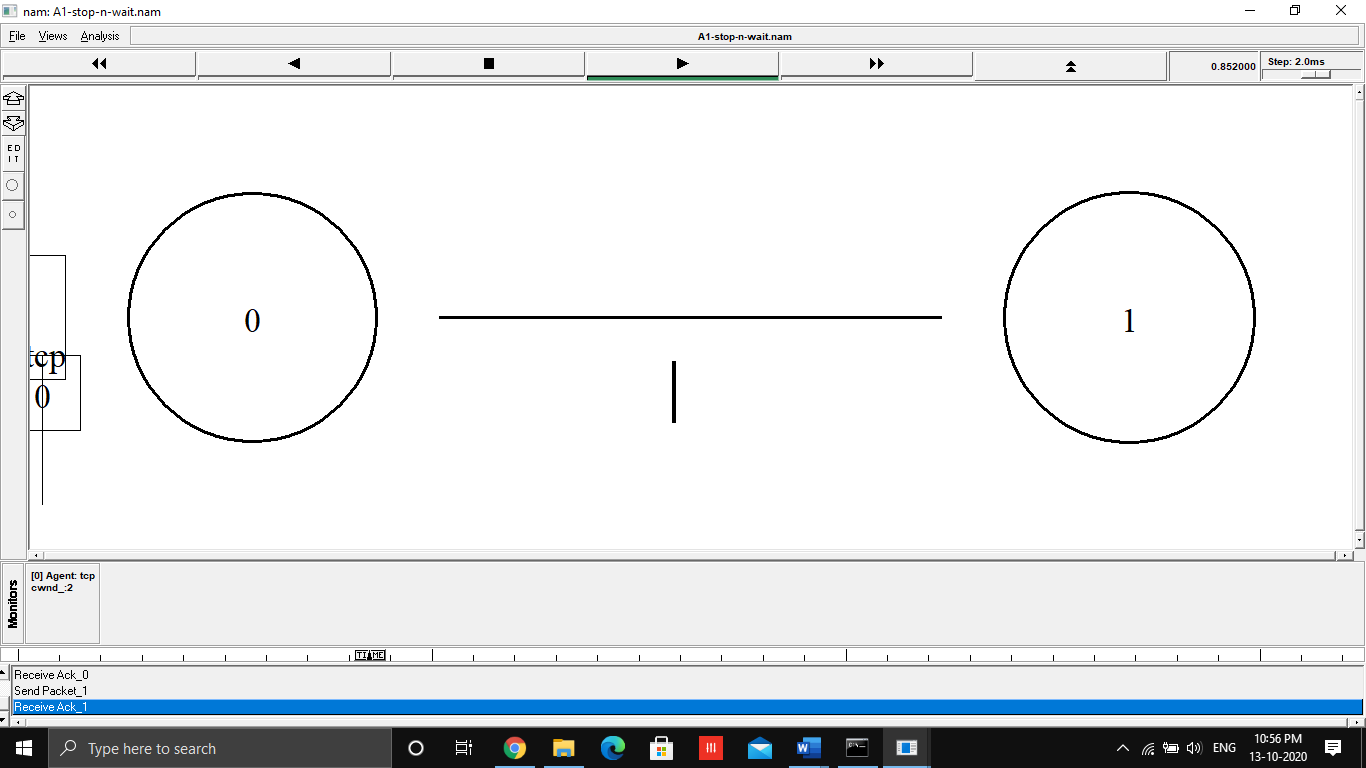
proc finish {} {   
 global ns nf   
 $ns flush-trace   
 close $nf

 puts "filtering..."   
 exec tclsh ../ns-allinone-2.1b5/nam-1.0a7/bin/namfilter.tcl A1-stop-n-wait.nam   
        puts "running nam..."   
        exec nam A1-stop-n-wait.nam &   
 exit 0   
}

$ns run 

**OUTPUT:**





**CONCLUSION:** The Program was successfully implemented and executed.