NETWORKS LABORATORY - 9

Stop and wait:

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf

proc finish {} {
        global ns nf
        $ns flush-trace
        close $nf
        exec nam out.nam &
        exit 0
}

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

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

$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link-op $n0 $n1 orient right

$ns queue-limit $n0 $n1 50

  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"
$ns at 1.0 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n1 $sink"
$ns at 1.5 "finish"

$ns run
```

Go back n/ Selective repeat:

```
set ns [new Simulator]
#Setting nodes in simulator
set n0 [$ns node]
set n1 [$ns node]
#Setting characteristics of nodes
$ns at 0.0 "$n0 label Sender"
$ns at 0.0 "$n1 label Receiver"
$n0 color red
$n1 color blue

set packet_size 200
#Saving trace and nam file
set nf [open Go-Back-N.nam w]
$ns namtrace-all $nf
set f [open Go-Back-N.tr w]
$ns trace-all $f

#Connection Specifics of nodes
$ns duplex-link $n0 $n1 0.2Mb 200ms DropTail
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n0 $n1 color green
$ns queue-limit $n0 $n1 50


#Tracing the Agent
Agent/TCP set nam_tracevar_ true

#Starting TCP agent
set tcp [new Agent/TCP]
$tcp set windowInit_ 3
$tcp set maxcwnd_ 3

#Attaching agent to node
$ns attach-agent $n0 $tcp

#Starting TCPSink Agent
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
#Attaching sink to node
$ns connect $tcp $sink

#Starting FTP
set ftp [new Application/FTP]
$ftp attach-agent $tcp

#Connection characteristics
$ns at 0.1 "$ftp start"
```

```
$ns at 0.68 "$ftp stop"
$ns at 1.07 "$ftp start"
$ns at 1.51 "$ftp stop"
$ns at 1.638 "$ftp start"
$ns at 2.078 "$ftp stop"
$ns at 2.206 "$ftp start"
$ns at 2.646 "$ftp stop"
$ns at 2.774 "$ftp start"
$ns at 3.214 "$ftp stop"
$ns at 3.342 "$ftp start"
$ns at 3.654 "$ftp stop"
$ns at 3.8 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n1 $sink"
$ns at 4.0 "finish"

#Procedure to finish file
proc finish {} {
        global ns
        $ns flush-trace
#       puts "filtering..."
#       exec tclsh ../bin/namfilter.tcl Go-Back-N.nam
        puts "running nam..."
        exec nam Go-Back-N.nam &
        exit 0
}
$ns run
```

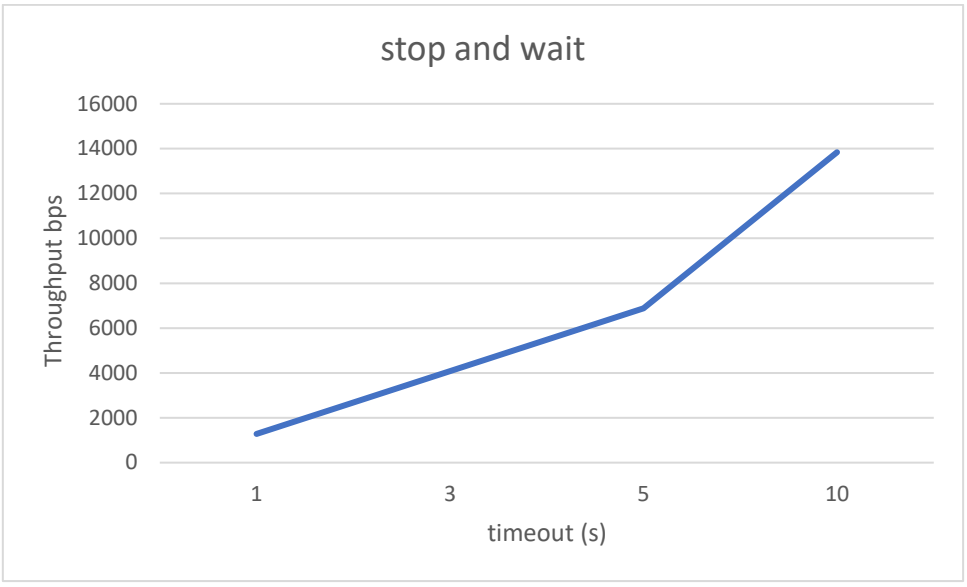
Finding throughput:

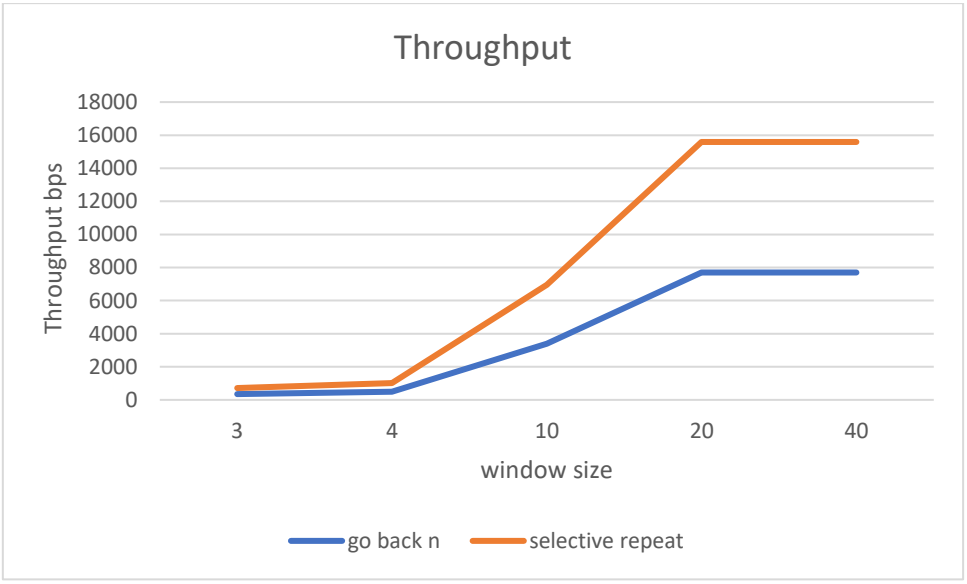
```
packets_sent = 0  # Total packets sent
packets_received = 0  # Total packets received

with open('Go-Back-N.nam', 'r') as trace_file:
    for line in trace_file:
        elements = line.split()
        event_type = elements[0]
        if event_type == '-':
            rout = elements[8]
            src=int(elements[4])
            dest=int(elements[6])
            if rout=='tcp' and src ==0:
                packets_sent += 1
        if event_type=='r':
            if rout=='tcp' and dest ==1:
                packets_received += 1

throughput = (packets_received/4)*200

print(f"Throughput: {throughput} bps")
```

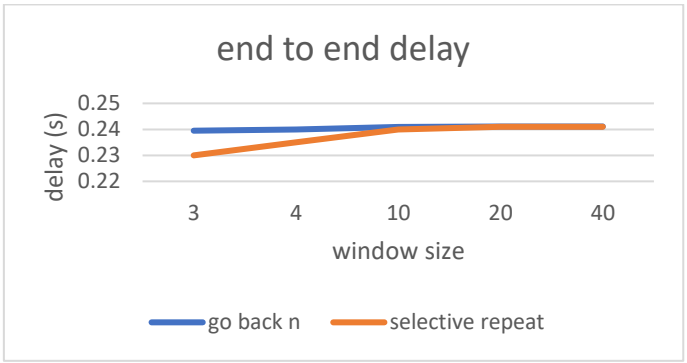
Finding end-to-end delay:

```
BEGIN {
    received = 0;
    delay = 0;
    src = 0;
    sink = 1;
}

{
    if ($1 == "-" && ($5 == src) ) {
        times[$15,"s"] = $3
    }

    if ($1 == "r" && $7 == sink) {
        times[$15,"r"] = $3
    }
}

END {
    for (id in times) {
        tempid = substr(id, 1, length(id)-2)
        if (times[tempid,"s"] && times[tempid,"r"]) {
            delay = times[tempid,"r"] - times[tempid,"s"]
            total_delay += delay
            received++
        }
    }

    if (received > 0) {
        avg_tcp_delay = total_delay / received
        print "Average TCP Delay:", avg_tcp_delay, "seconds"
    } else {
        print "No TCP packets received."
    }

}
```

Graphs:

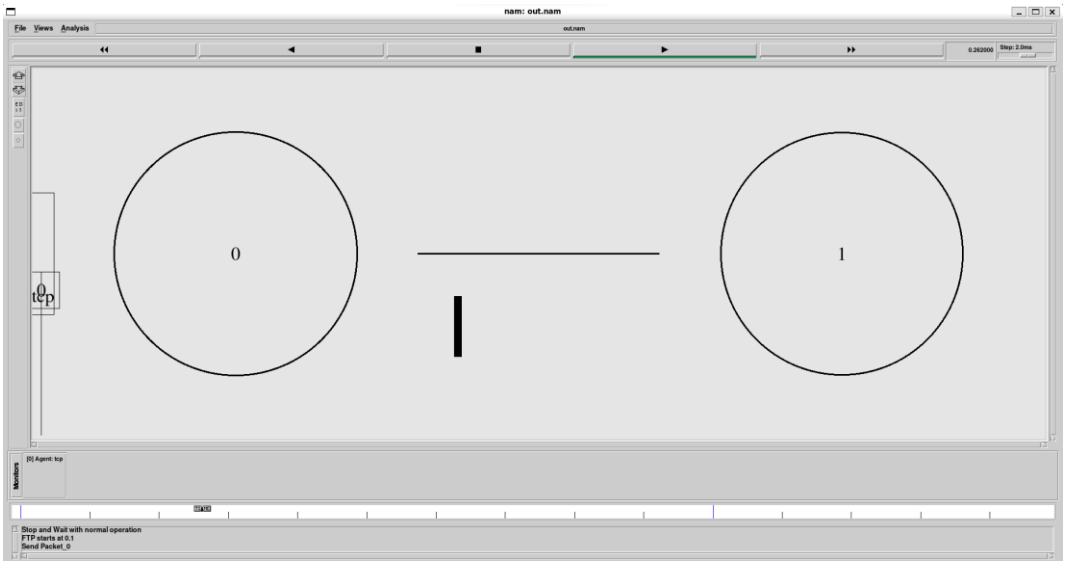

stop and wait
Throughput bps
16000
14000
12000
10000
8000
6000
4000
2000
0
1
3
5
10
timeout (s)


Throughput
Throughput bps
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
3
4
10
20
40
window size
go back n
selective repeat


end to end delay
delay (s)
0.25
0.24
0.23
0.22
3
4
10
20
40
window size
go back n
selective repeat

Simulations:

Stop and wait



Go back n/ Selective repeat: