Program:

```
include <wifi_lte/wifi_lte_rtable.h>
  struct r_hist_entry *elm, *elm2;
int num_later = 1;
elm = STAILQ_FIRST(&r_hist_);
while (elm != NULL && num_later <= num_dup_acks_){
            num_later;
            elm = STAILQ_NEXT(elm, linfo_);
}

      if (elm != NULL){
            elm = findDataPacketInRecvHistory(STAILQ_NEXT(elm,linfo_));

            if (elm != NULL){
                        elm2 = STAILQ_NEXT(elm, linfo_);
                        while(elm2 != NULL){
                                    if (elm2->seq_num_ < seq_num && elm2->t_recv_ < time){
                                                STAILQ_REMOVE(&r_hist_,elm2,r_hist_entry,linfo_);
                                          delete elm2;
                                    } else
                                                elm = elm2;
                                    elm2 = STAILQ_NEXT(elm, linfo_);
                        }
            }
}
}
void DCCPTFRCAgent::removeAcksRecvHistory(){
struct r_hist_entry *elm1 = STAILQ_FIRST(&r_hist_);
struct r_hist_entry *elm2;

int num_later = 1;
while (elm1 != NULL && num_later <= num_dup_acks_){
            num_later;
            elm1 = STAILQ_NEXT(elm1, linfo_);
}

if(elm1 == NULL)
            return;

elm2 = STAILQ_NEXT(elm1, linfo_);
while(elm2 != NULL){
            if (elm2->type_ == DCCP_ACK){
                        STAILQ_REMOVE(&r_hist_,elm2,r_hist_entry,linfo_);
                        delete elm2;
            } else {
                        elm1 = elm2;
            }
            elm2 = STAILQ_NEXT(elm1, linfo_);
}
}
```

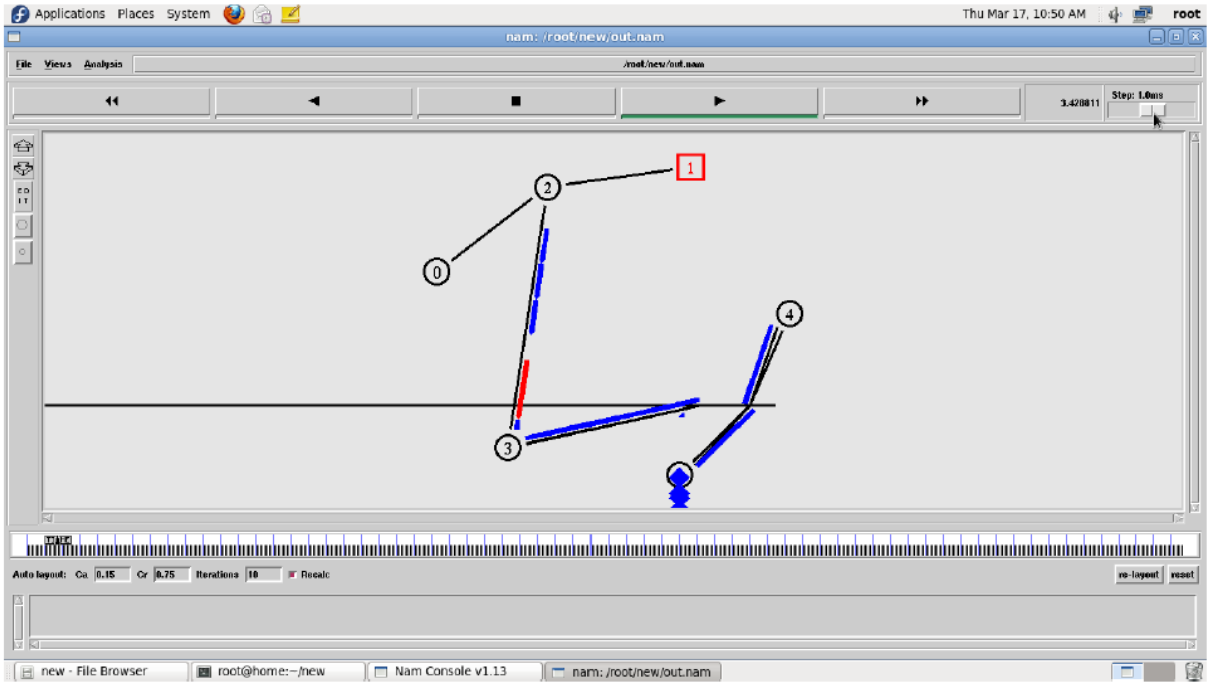
```
inline r_hist_entry *DCCPTFRCAgent::findDataPacketInRecvHistory(r_hist_entry *start){
while(start != NULL && start->type_ == DCCP_ACK)
        start = STAILQ_NEXT(start,linfo_);
return start;
}
```

Program:

```
set ns [new Simulator]
#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
#Open the Trace files
set file1 [open out.tr w]
set winfile [open WinFile w]
$ns trace-all $file1
#Open the NAM trace file
set file2 [open out.nam w]
$ns namtrace-all $file2
#Define a 'finish' procedure
proc finish {} {
global ns file1 file2
$ns flush-trace
close $file1
close $file2
exec nam out.nam &
exit 0
}
#Create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color red
$n1 shape box
#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Ca Channel]
Setup a TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Setup a UDP connection
```

```
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 0.01mb
$cbr set random_ false
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
$ns at 124.5 "$cbr stop"
# next procedure gets two arguments: the name of the
# tcp source node, will be called here "tcp",
# and the name of output file.
proc plotWindow {tcpSource file} {
global ns
set time 0.1
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
set wnd [$tcpSource set window_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file" }
$ns at 0.1 "plotWindow $tcp $winfile"
$ns at 5 "$ns trace-annotate \"packet drop\""
# PPP
$ns at 125.0 "finish"
$ns run
```

Output :


Applications Places System
Thu Mar 17, 10:50 AM
root
nam: /root/new/out.nam
File Views Analysis
/root/new/out.nam
Step: 1.0ms
0
1
2
3
4
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
re-layout reset
new - File Browser
root@home:~/new
Nam Console v1.13
nam: /root/new/out.nam