Université Mohammed V - Rabat
École Nationale d'Informatique
et d'Analyse des Systèmes

Année universitaire : 2023/2024

Filière Génie Logiciel

Analyzing and Comparing the Performance of TCP NewReno and TCP Vegas

---

Directed by:
NADIR Oumaima

Supervised by:
Pr BERQIA Amine

February 9, 2024

CONTENTS

CHAPTER 1

COMPARISON BETWEEN TCP NEW RENO AND TCP NEW VEGAS

1.1 Introduction

In the field of communication networks, the performance of congestion control protocols, such as TCP, is of crucial importance to ensure efficient and reliable data transfer. In this study, we use NS2 (Network Simulator 2), a widely recognized simulation tool, to model and evaluate the performance of two variants of TCP: **TCP New Reno et TCP Vegas.**

1.2 Code TCL (Tool Command Language)

```
#=== Node Configuration options ===#
##= Channel Type =##
set val(chan) Channel/WirelessChannel;
##= Radio-propagation model =##
set val(prop) Propagation/TwoRayGround;
##= Network interface type =##
set val(netif) Phy/WirelessPhy;
##= MAC type=##
set val(mac)  Mac/802_11 ;
##= Interface queue type=##
set val(ifq) Queue/DropTail/PriQueue ;
##= Link layer type=##
set val(ll)  LL;
##= Antenna model=##
set val(ant) Antenna/OmniAntenna ;
##= Max packet in ifq=##
set val(ifqlen) 50 ;
```

```
##= Number of mobilnodes=##
set val(nn) 7 ;
##= Routing Protocol=##
set val(rp) DSDV ;
##= topo dim=##
set val(x) 1000;
set val(y) 1000;
#================================================#
#=== ===#
##= =##
set ns [new Simulator]
$ns color 2 Red
$ns color 1 Bleu
##= =##
set tracefd [open sanet.tr w]
$ns trace-all $tracefd
##= =##
set namtrace [open sanet.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
##= =##
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
##= =##
create-god $val(nn)
#================================================#
#=== ===#
set chan_1 [new $val(chan)]
set chan_2 [new $val(chan)]
#= =#
#== ==#
$ns node-config -adhocRouting $val(rp) -llType $val(ll) -macType $val(mac) -ifqType $val(ifq) -ifqLen $v
#== ==#
##= =##
for {set i 0} {$i < 7} {incr i} {
set ID_($i) $i;
set node_($i) [$ns node];
$node_($i) set id $ID_($i);
$node_($i) random-motion 0;
$ns initial_node_pos $node_($i) 20;
}
##= =##
Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 4.4613e-10    ;#250m
Phy/WirelessPhy set RXThresh_ 4.4613e-10    ;#250m
#Phy/WirelessPhy set bandwidth_ 512kb
#Phy/WirelessPhy set Pt_ 0.2818
#Phy/WirelessPhy set freq_ 2.4e+9
#Phy/WirelessPhy set L_ 1.0
```

```
66  #Antenna/OmniAntenna set X_ 0
67  #Antenna/OmniAntenna set Y_ 0
68  #Antenna/OmniAntenna set Z_ 0.25
69  #Antenna/OmniAntenna set Gt_ 1
70  #Antenna/OmniAntenna set Gr_ 1
71  ##= =##
72
73  ##= =##
74
75  $node_(0) set X_ 465.0;
76  $node_(0) set Y_ 800.0;
77  $node_(0) set Z_ 0.0;
78
79  $node_(1) set X_ 685.0;
80  $node_(1) set Y_ 800.0;
81  $node_(1) set Z_ 0.0;
82
83  $node_(2) set X_ 575.0;
84  $node_(2) set Y_ 500.0;
85  $node_(2) set Z_ 0.0;
86
87  $node_(6) set X_ 685.0;
88  $node_(6) set Y_ 200.0;
89  $node_(6) set Z_ 0.0;
90
91  $node_(5) set X_ 465.0;
92  $node_(5) set Y_ 200.0;
93  $node_(5) set Z_ 0.0;
94
95  $node_(3) set X_ 575.0;
96  $node_(3) set Y_ 650.0;
97  $node_(3) set Z_ 0.0;
98
99  $node_(4) set X_ 575.0;
100 $node_(4) set Y_ 350.0;
101 $node_(4) set Z_ 0.0;
102
103 $ns at 0.0 "$node_(0) setdest 465.0 800.0 0.0";
104 $ns at 0.0 "$node_(1) setdest 685.0 800.0 0.0";
105 $ns at 0.0 "$node_(2) setdest 575.0 500.0 0.0";
106 $ns at 0.0 "$node_(6) setdest 685.0 200.0 0.0";
107 $ns at 0.0 "$node_(5) setdest 465.0 200.0 0.0";
108 $ns at 0.0 "$node_(3) setdest 575.0 650.0 0.0";
109 $ns at 0.0 "$node_(4) setdest 575.0 350.0 0.0";
110
111 $ns at 0.0 "$node_(0) label Sender_a";
112 $ns at 0.0 "$node_(1)  label Sender_b";
113 $ns at 0.0 "$node_(2)  label Gateway";
```

```
114 $ns at 0.0 "$node_(3)  label Gateway";
115 $ns at 0.0 "$node_(4)  label Gateway";
116 $ns at 0.0 "$node_(5)  label Recevier_a";
117 $ns at 0.0 "$node_(6)  label Recevier_b";
118 ##= =##
119 #for {set i 0} {$i < $val(nn) } { incr i } {
120 #        $ns at [ expr 15+round(rand()*500) ] "$node_($i) setdest [ expr 10+round(rand()*500) ] [ expr 1
121 #    }
122 #$ns at 200.0 "$node_(3) setdest 450.0 320.0 20.0";
123 #=====================================================#
124 ##
125 set tcp1 [new Agent/TCP/Newreno];
126 $tcp1 set class_ 2;
127 set sink1 [new Agent/TCPSink];
128 $ns attach-agent $node_(1) $tcp1;
129 $ns attach-agent $node_(6) $sink1;
130 $ns connect $tcp1 $sink1;
131 set ftp1 [new Application/FTP];
132 $ftp1 attach-agent $tcp1;
133 $ns at 3.0 "$ftp1 start";
134 $ns at 590.0 "$ftp1 stop";
135 ##
136 set tcp2 [new Agent/TCP/Vegas];
137 $tcp2 set class_ 1;
138 set sink2 [new Agent/TCPSink];
139 $ns attach-agent $node_(0) $tcp2;
140 $ns attach-agent $node_(5) $sink2;
141 $ns connect $tcp2 $sink2;
142 set ftp2 [new Application/FTP];
143 $ftp2 attach-agent $tcp2;
144 $ns at 3.0 "$ftp2 start";
145 $ns at 590.0 "$ftp2 stop";
146 #=====================================================#
147 ##
148 for {set i 0} {$i < $val(nn)} {incr i} {
149 $ns at 600.0 "$node_($i) reset";
150 }
151 ##
152 $ns at 600.0 "stop";
153 $ns at 600.01 "puts \"NS EXITING\" ;$ns halt"
154 proc stop {} {
155 global ns tracefd namtrace
156 $ns flush-trace
157 close $tracefd
158 close $namtrace
159 }
160 puts "Starting NS"
161 $ns run
```

162

1.3 Architecture

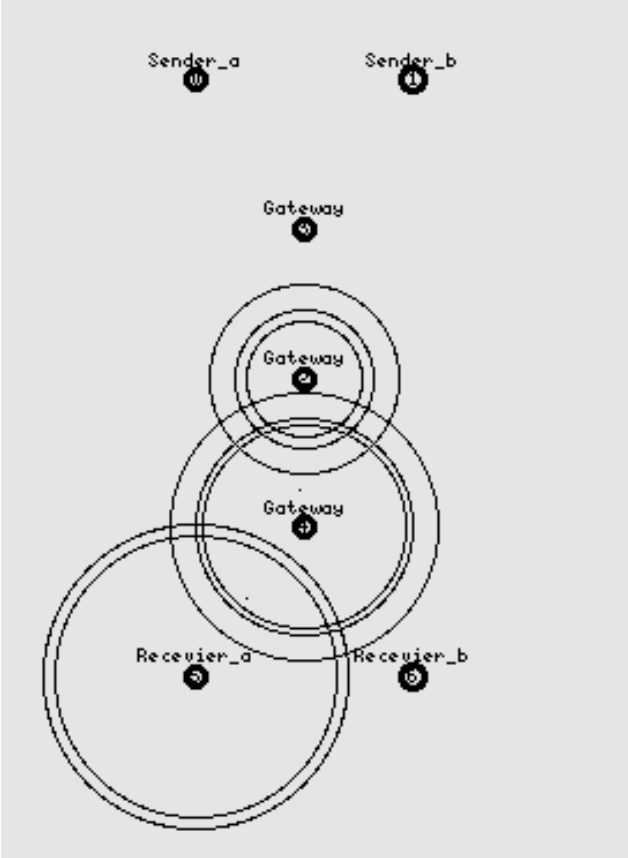

Figure 1.1: SANET Architecture

1.4 AWK Analysis Script

```
BEGIN {
    # Initialization as before
    newreno_sent_pkt = 0
    newvegas_sent_pkt = 0
    newreno_rcv_pkt = 0
    newvegas_rcv_pkt = 0
    startTimeNewreno = 0
    isStartNewreno = 0
    startTimeNewvegas = 0
    isStartNewvegas = 0
```

```
    newreno_arrival_rate = 0
    newvegas_arrival_rate = 0
    packet_sent[7] = 0
    packet_recv[7] = 0
    packet_forw[7] = 0
    newreno_loss_over_time[10] = 0
    newvegas_loss_over_time[10] = 0

    # Create arrays to store loss ratios over time
    time_points = 60  # Adjust the number of time points as needed
    time_interval = 10  # Adjust the time interval as needed
}

{
    # Main processing block as before
    event = $1
    time = $2
    node_id = $3
    level = $4
    pkt_type = $7

    for (i = 0; i < time_points; i++) {

    if (i * time_interval <= time && time < time_interval * (i + 1)) {

   if (node_id == "_0_" && pkt_type == "tcp") {
if (event == "s" && level == "AGT") {
   packet_sent[0] = packet_sent[0] + 1
}
if (event == "r" && level == "AGT") {
   packet_recv[0] = packet_recv[0] + 1
}
if (event == "f") {
   packet_forw[0] = packet_forw[0] + 1
}
   }
   if (node_id == "_1_" && pkt_type == "tcp") {
if (event == "s" && level == "AGT") {
   packet_sent[1] = packet_sent[1] + 1
}
if (event == "r" && level == "AGT") {
   packet_recv[1] = packet_recv[1] + 1
}
if (event == "f") {
```

```
55      packet_forw[1] = packet_forw[1] + 1
56  }
57      }
58      if (node_id == "_2_" && pkt_type == "tcp") {
59  if (event == "s" && level == "AGT") {
60      packet_sent[2] = packet_sent[2] + 1
61  }
62  if (event == "r" && level == "AGT") {
63      packet_recv[2] = packet_recv[2] + 1
64  }
65  if (event == "f") {
66      packet_forw[2] = packet_forw[2] + 1
67  }
68      }
69      if (node_id == "_3_" && pkt_type == "tcp") {
70  if (event == "s" && level == "AGT") {
71      packet_sent[3] = packet_sent[3] + 1
72  }
73  if (event == "r" && level == "AGT") {
74      packet_recv[3] = packet_recv[3] + 1
75  }
76  if (event == "f") {
77      packet_forw[3] = packet_forw[3] + 1
78  }
79      }
80      if (node_id == "_4_" && pkt_type == "tcp") {
81  if (event == "s" && level == "AGT") {
82      packet_sent[4] = packet_sent[4] + 1
83  }
84  if (event == "r" && level == "AGT") {
85      packet_recv[4] = packet_recv[4] + 1
86  }
87  if (event == "f") {
88      packet_forw[4] = packet_forw[4] + 1
89  }
90      }
91      if (node_id == "_5_" && pkt_type == "tcp") {
92  if (event == "s" && level == "AGT") {
93      packet_sent[5] = packet_sent[5] + 1
94  }
95  if (event == "r" && level == "AGT") {
96      packet_recv[5] = packet_recv[5] + 1
97  }
98  if (event == "f") {
```

```
99      packet_forw[5] = packet_forw[5] + 1
100  }
101     }
102     if (node_id == "_6_" && pkt_type == "tcp") {
103  if (event == "s" && level == "AGT") {
104     packet_sent[6] = packet_sent[6] + 1
105  }
106  if (event == "r" && level == "AGT") {
107     packet_recv[6] = packet_recv[6] + 1
108  }
109  if (event == "f") {
110     packet_forw[6] = packet_forw[6] + 1
111  }
112     }
113
114     newreno_sent_pkt = packet_sent[1] - packet_forw[1];
115     newreno_rcv_pkt = packet_recv[6] - packet_forw[6];
116     newvegas_sent_pkt = packet_sent[0] - packet_forw[0];
117     newvegas_rcv_pkt = packet_recv[5] - packet_forw[5];
118
119     newreno_loss_over_time[i] = (newreno_sent_pkt - newreno_rcv_pkt) / newreno_sent_pkt * 100;
120     newvegas_loss_over_time[i] = (newvegas_sent_pkt - newvegas_rcv_pkt) / newvegas_sent_pkt * 100;
121      }
122
123      }
124  }
125
126  END {
127      for (i = 5; i < time_points; i++) {
128              printf("Time Point %d: NewReno Loss Ratio = %f, NewVegas Loss Ratio = %f\n", i * time_interv
129      }
130  }
```

1.5 Results

NewReno:

- Sent packets = 21,032
- Received packets = 14,068
- Arrival ratio = 66.89%
- Packet loss ratio = 33.11%

Vegas:

- Sent packets = 10,979
- Received packets = 8,272
- Arrival ratio = 75.34%
- Packet loss ratio = 24.66%

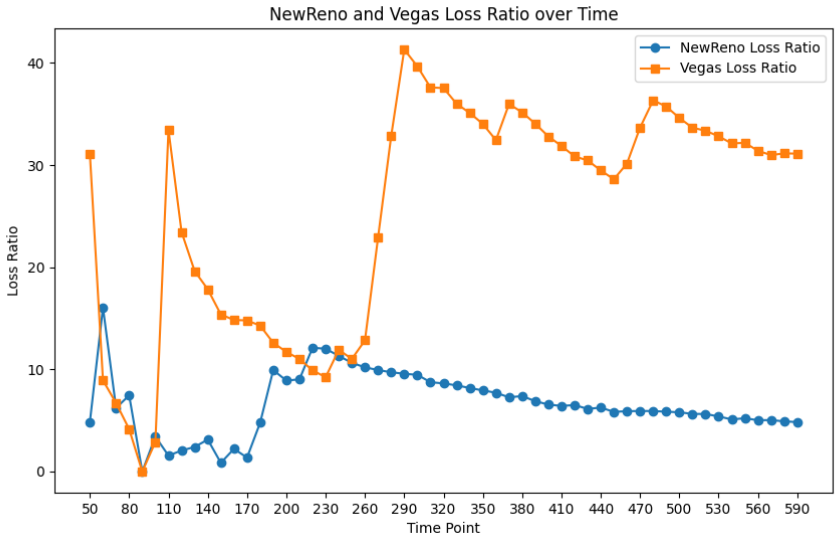

Figure 1.2: Comparative Curves of Packet Loss Ratio (PLR) for TCP NewReno and Vegas

Interpretation :

At first, it seemed like TCP Vegas had fewer lost packets compared to TCP New Reno. However, when we looked at the overall or average results later, it turned out that, on average, TCP New Reno had a lower rate of lost packets. This difference highlights how tricky it can be to figure out which method is better for controlling traffic in computer networks. It suggests that we need to look at more than just lost packets – things like how fast the data is moving and how quickly it gets to its destination are also important. These findings help us understand the subtle differences in how TCP Vegas and TCP New Reno perform, guiding decisions on how to make computer networks work better.

WEBOGRAPHIE

[1] https://www.nsnam.com

[2] https://https://www.nsnam.com/2023/02/ns2-installation-in-ubuntu-2204.html

[3] https://www.researchgate.net

[4] https://en.wikipedia.org

[5] https://www.geeksforgeeks.org

[6] https://ns2simulator.com/trace-file-in-ns2

[7] https://networksimulationtools.com