

# Лабораторная работа №2

Имитационное моделирование

---

Волгин И.А.

20 февраля 2025

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Волгин Иван Алексеевич
- Студент группы НФИбд-01-22
- Российский университет дружбы народов

## Выполнение лабораторной работы

---

Целью работы является исследовать протокол TCP и алгоритм управления очередью RED.

- Создать пример сети с дисциплиной RED
- Выполнить дополнительное упражнение

## Код для модели сети из задания 1

```
set N 5
for {set i 1} {$i < $N} {incr i} {
set node_($i) [$ns node]
}
set node_(r1) [$ns node]
set node_(r2) [$ns node]

$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail

set tcp1 [$ns create-connection TCP/Reno $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

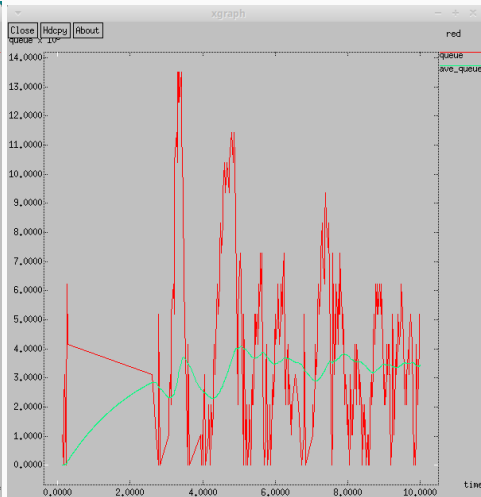
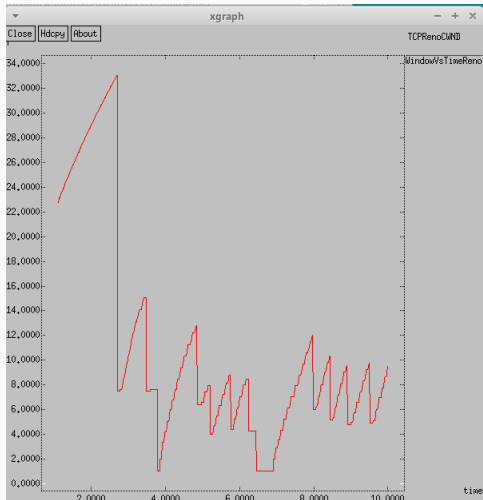
set windowVsTime [open WindowVsTimeReno w]
set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
[$ns link $node_(r1) $node_(r2)] queue-sample-timeout;

set redq [[$ns link $node_(r1) $node_(r2)] queue]
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_

$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"

$ns run
```

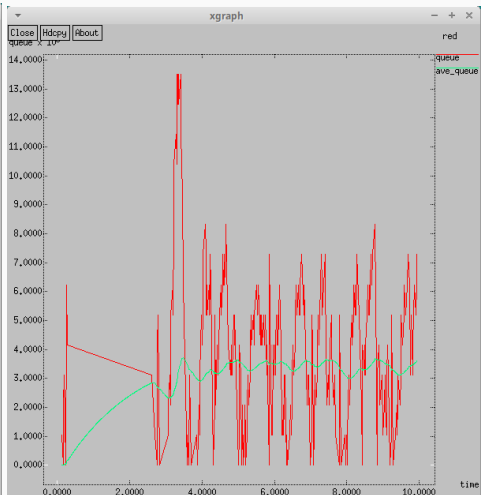
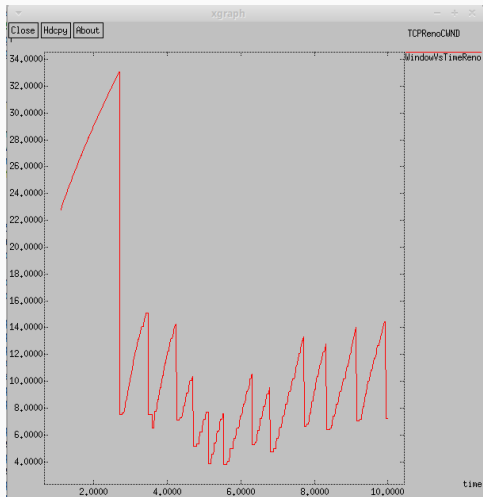
## Графики размер окна и размера очереди





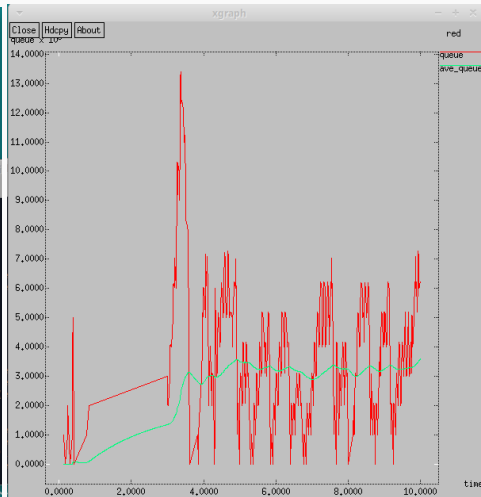
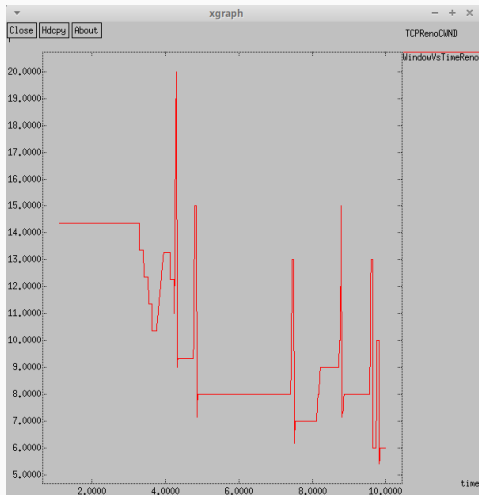
```
set tcp1 [$ns create-connection TCP/Newreno $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]
```

# Графики рамера окна и размера очереди с типом TCP Newreno



```
set tcp1 [$ns create-connection TCP/Vegas $node_(s1) TCPSink $node_(s3) 0]  
$tcp1 set window_ 15  
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]  
$tcp2 set window_ 15  
set ftp1 [$tcp1 attach-source FTP]  
set ftp2 [$tcp2 attach-source FTP]
```

# Графики рамера окна и размера очереди с типом TCP Vegas



# Код для изменения оформления

```
}
}
set f {open temp.queue w}
puts $f "TitleText: red"
puts $f "Device: Postscript"
puts $f "0.Color: Yellow"
puts $f "1.Color: Green"
if { [info exists tchan_] } {
close $tchan_
}
exec rm -f temp.q temp.a
exec touch temp.a temp.q
exec awk $awkCode all.q
puts $f "\"QUEUE\""
exec cat temp.q >@ $f
puts $f "\n\"AVG QUEUE"
exec cat temp.a >@ $f
close $f
exec xgraph -fg blue -bg white -bb -tk -x time -t "TCPrenoCWND" WindowVsTimeReno
exec xgraph -fg gray -bg brown -bb -tk -x time -y queue temp.queue &
exit 0
}

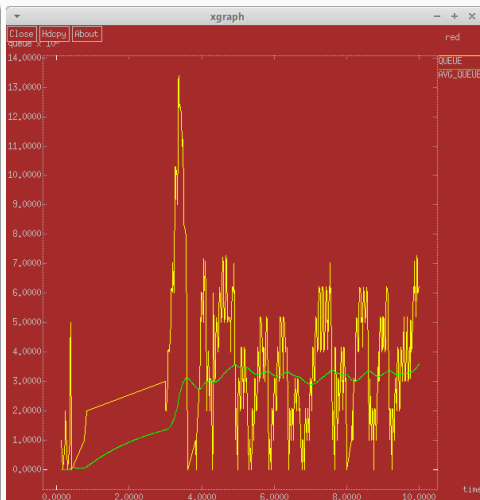
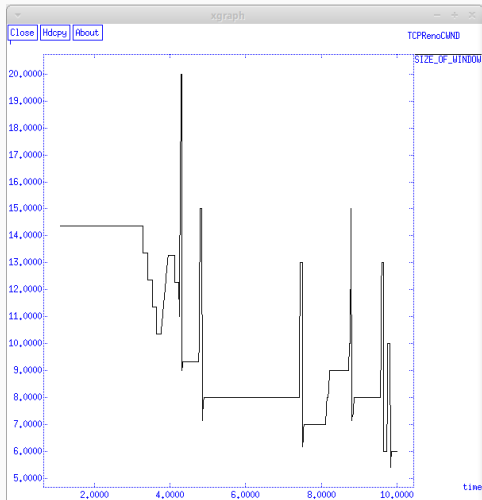
set N 5
for {set i 1} {$i < $N} {incr i} {
set node_($i) [$ns node]
}
set node_(r1) [$ns node]
set node_(r2) [$ns node]

$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail

set tcp1 [$ns create-connection TCP/Vegas $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window 15
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window 15
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]

set windowVsTime [open WindowVsTimeReno w]
puts $windowVsTime "0.Color: Black"
puts $windowVsTime "\"SIZE OF WINDOW\""
set qmon [$ns monitor-queue $node_(r1) $node_(r2) [open qm.out w] 0.1];
[ens link $node_(r1) $node_(r2)] queue-sample-timeout
```

# Графики рамера окна и размера очереди с новым оформлением



В ходе выполнения лабораторной работы я исследовал прокол TCP и алгоритм управления очередью RED