

# Лабораторная работа №2

Исследование протокола TCP и алгоритма управления очередью RED

---

Городянский Ф.Н.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Городянский Фёдор Николаевич
- студент
- Российский университет дружбы народов
- 1132226456@pfur.ru
- <https://Fedass.github.io/ru/>

Исследовать протокол TCP и алгоритм управления очередью RED.

1. Выполнить пример с дисциплиной RED;
2. Изменить в модели на узле s1 тип протокола TCP с Reno на NewReno, затем на Vegas.  
Сравнить и пояснить результаты;
3. Внести изменения при отображении окон с графиками (изменить цвет фона, цвет траекторий, подписи к осям, подпись траектории в легенде).

# Выполнение лабораторной работы

```
for (set i 1) ($i < $N) {incr i} {
set node_($i) [$ns node]
}
set node_r1 [$ns node]
set node_r2 [$ns node]

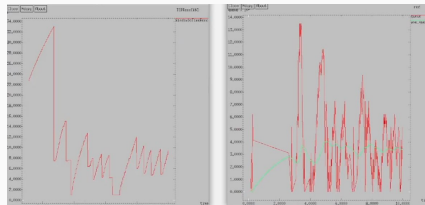
# Соединения:
$ns duplex-link $node_s1 $node_r1 10Mb 2ms DropTail
$ns duplex-link $node_s2 $node_r1 10Mb 3ms DropTail
$ns duplex-link $node_r1 $node_r2 1.5Mb 20ms RED
$ns queue-limit $node_r1 $node_r2 25
$ns queue-limit $node_r2 $node_r1 25
$ns duplex-link $node_s3 $node_r2 10Mb 4ms DropTail
$ns duplex-link $node_s4 $node_r2 10Mb 5ms DropTail
# Агенты и приложения:
set tcp1 [$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]

# Мониторинг размера окна TCP:
set windowVsTime [open WindowVsTimeReno w]
set qmon [$ns monitor-queue $node_r1 $node_r2 [open qn.out w] 0.1];
[$ns link $node_r1 $node_r2] queue-sample-timeout;
# Мониторинг очереди:
set redq [$ns link $node_r1 $node_r2] queue]
set tchan_ [open all.q w]
$redq trace curq_
$redq trace ave_
$redq attach $tchan_

# ат-событие для планировщика событий, которое запускает
# процедуру finish через 10 с после начала моделирования
# Добавление ат-событий:
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow $tcp1 $windowVsTime"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"

# запуск модели
$ns run
```

Рис. 1: TCP/Reno



**Рис. 2:** Графики динамики размера окна TCP, динамики длины очереди и средней длины очереди

# Изменение протокола TCP

```
}  
# Визуализация файла с данными о размере окна TCP:  
proc plotwindow (tcpSource file) {  
    global ns  
    set time 0.01  
    set now [$ns now]  
    set cwnd [tcpSource cwnd]  
    puts $file "show $cwnd"  
    $ns at [expr $now+$time] "plotwindow tcpSource $file"  
}  
  
# Визу сетки:  
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]  
  
# Соединения:  
$ns duplex-link $node_{$i1} $node_r1 10Mb 2ms DropTail  
$ns duplex-link $node_{$i2} $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 $node_{$i3} $node_r2 10Mb 4ms DropTail  
$ns duplex-link $node_{$i4} $node_r2 10Mb 5ms DropTail  
# Агенты и приложения:  
set tcp1 [$ns create-connection TCP/Newreno $node_{$i1} TCPSink $node_{$i3} 0]  
$tcp1 set window 15  
set tcp2 [$ns create-connection TCP/Reno $node_{$i2} TCPSink $node_{$i3} 1]  
$tcp2 set window 15  
set ftp1 [$tcp1 attach-source FTP]  
set ftp2 [$tcp2 attach-source FTP]  
  
# Мониторинг размера окна TCP:  
set windowVsTime [open WindowVsTime.rno w]  
set que [ $ns monitor-queue $node_r1 $node_r2 [open que.out w] 8.1];  
$ns link $node_r1 $node_r2 queue-sample-timeout;  
# Мониторинг очереди:  
set rodq [[$ns link $node_r1 $node_r2] queue]  
set tchan [open all.q w]  
$rodq trace curq
```

Рис. 3: Тип NewReno



## Изменение протокола TCP



**Рис. 4:** График динамики длины очереди и средней длины очереди. Тип NewReno

# Изменение протокола TCP

```
* /usr/share/model/traffic-scripts/changed.tcl - Modeler

Файл Правка Поиск Вид Документ Справка

set conn [tcpSource set conn]
puts $file "Snow Start"
$ns at [expr $now+$time] "plotWindow tcpSource $file"
}

# Узлы сети:
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]

# Соединения:
$ns duplex-link $node_($i) $node_r1 10Mb 2ms DropTail
$ns duplex-link $node_($i) $node_r2 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 $node_($i) $node_r2 10Mb 4ms DropTail
$ns duplex-link $node_($i) $node_r2 10Mb 5ms DropTail

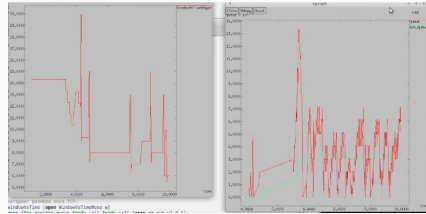
# Агенты и приложения:
set tcp1 [$ns create-connection TCP/Vegas $node_($i) TCPSink $node_($i) 0]
tcp1 set window 15
set tcp2 [$ns create-connection TCP/Reno $node_($i) TCPSink $node_($i) 1]
tcp2 set window 15
set ftp1 [tcp1 attach-source FTP]
set ftp2 [tcp2 attach-source FTP]

# Мониторинг размера окна TCP:
set windowVstime [open WindowVstimeReno w]
set open [$ns monitor-queue $node_r1 $node_r2] [open qn.out w] 0.1;
[$ns link $node_r1 $node_r2] queue-sample-timeout;
# Мониторинг очереди:
set redq [[$ns link $node_r1 $node_r2] queue]
set tchan [open all.q w]
$redq trace curq
$redq trace ave
$redq attach $tchan

# at-событие для планировка событий, которое запускает
# процедуру finish через 10 с после начала моделирования
# Добавление at-событий:
$ns at 0.0 "$ftp1 start"
$ns at 1.1 "plotWindow tcp1 $windowVstime"
$ns at 3.0 "$ftp2 start"
```

Рис. 5: Тип Vegas

# Изменение протокола TCP



**Рис. 6:** Графики динамики размера окна TCP, динамики длины очереди и средней длины очереди. Тип Vegas

# Изменение отображения окон с графиками

```
set end $2
}
else if ($1 == "a" && NF>2)
print $2, $3 >> "temp.a";
}
}
set f [open temp.queue w]
puts $f "TitleText: red"
puts $f "Device: Postscript"
puts $f "0.Color: Blue"
puts $f "1.Color: Purple"
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 "\nqueue Vegas"
exec cat temp.q >> $f
puts $f "\nave.queue Vegas"
exec cat temp.a >> $f
close $f

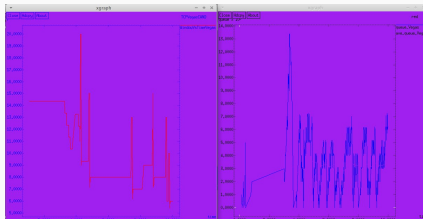
exec xgraph -fg blue -bg purple -bb -tk -x time -t "TCPVegasCMD" WindowVsTimeVegas &
exec xgraph -fg black -bg purple -bb -tk -x time -y queue temp.queue &
exit 0
}

# Сохранение файла с данными о размере окна TCP:
proc plotWindow {tcpSource file} {
global ns
set time 0.01
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}

# Узлы сети:
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]
```

Рис. 7: Тип Vegas с настройкой цветов

## Изменение отображения окон с графиками



**Рис. 8:** Графики динамики размера окна TCP, динамики длины очереди и средней длины очереди с изменением отображения

В процессе выполнения данной лабораторной работы я исследовал протокол TCP и алгоритм управления очередью RED.