

Appendix A: NS-2 Code

UDP:

```
# start new simulator
set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

#open tracefile
set nd [open out.tr w]

$ns trace-all $nd

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

set the colors of the packets being sent

$ns color 1 Blue

$ns color 2 Red

$ns color 3 black

$ns color 4 gold

$ns color 5 orange

# nodes for users and routers connect

set U0 [$ns node]

set U1 [$ns node]

set U2 [$ns node]

set U3 [$ns node]

set U4 [$ns node]

set U5 [$ns node]
```

```
set U6 [$ns node]
set U7 [$ns node]
set R1 [$ns node]
set R2 [$ns node]

#colors for the nodes
$U0 color darkgreen
$U1 color darkgreen
$U2 color darkgreen
$U3 color darkgreen
$U4 color darkgreen
$U5 color darkgreen
$U6 color darkgreen
$U7 color darkgreen
$R1 color purple
$R2 color purple

# labeling the nodes
$U0 label "User 1"
$U1 label "CBR1 L 25.89Mb"
$U3 label "CBR2 L 25.91Mb"
$U3 label "CBR3 L 25.93Mb"
$U4 label "User 2"
$U5 label "CBR1 R 25.89Mb"
$U6 label "CBR2 R 25.91Mb"
$U7 label "CBR3 R 25.92Mb"
$R1 label "router1 "
$R2 label "router2"

# shapes for the routers
$R1 shape box
$R2 shape box

# how connections are established and bandwidth for our lines
```

```
$ns duplex-link $U0 $R1 64kb 5ms DropTail
$ns duplex-link $U1 $R1 26Mb 5ms DropTail
$ns duplex-link $U2 $R1 26Mb 5ms DropTail
$ns duplex-link $U3 $R1 26Mb 5ms DropTail
$ns duplex-link $U4 $R2 64kb 5ms DropTail
$ns duplex-link $U5 $R2 26Mb 5ms DropTail
$ns duplex-link $U6 $R2 26Mb 5ms DropTail
$ns duplex-link $U7 $R2 26Mb 5ms DropTail
# link between routers
$ns duplex-link $R1 $R2 25.92Mb 50ms DropTail
#more control over the layout
$ns duplex-link-op $U0 $R1 orient right-down
$ns duplex-link-op $U1 $R1 orient right
$ns duplex-link-op $U2 $R1 orient right-up
$ns duplex-link-op $U3 $R1 orient up
$ns duplex-link-op $R1 $R2 orient right
$ns duplex-link-op $U4 $R2 orient left-down
$ns duplex-link-op $U5 $R2 orient left
$ns duplex-link-op $U6 $R2 orient left-up
$ns duplex-link-op $U7 $R2 orient up
# creating our udp agents and attaching it to the correct nodes
set udp0 [new Agent/UDP]
$ns attach-agent $U0 $udp0
$udp0 set fid_ 1
set udp1 [new Agent/UDP]
$ns attach-agent $U1 $udp1
$udp1 set fid_ 3
set udp2 [new Agent/UDP]
$ns attach-agent $U2 $udp2
$udp2 set fid_ 4
```

```
set udp3 [new Agent/UDP]
$ns attach-agent $U3 $udp3
$udp3 set fid_ 5
set udp4 [new Agent/UDP]
$ns attach-agent $U4 $udp4
$udp4 set fid_ 2
set udp5 [new Agent/UDP]
$ns attach-agent $U5 $udp5
$udp5 set fid_ 6
set udp6 [new Agent/UDP]
$ns attach-agent $U6 $udp6
$udp6 set fid_ 7
set udp7 [new Agent/UDP]
$ns attach-agent $U7 $udp7
$udp7 set fid_ 8

#to find out what is going on in the queue
$ns duplex-link-op $U0 $R1 queuePos 0.5
$ns duplex-link-op $R1 $R2 queuePos 0.5
$ns duplex-link-op $U4 $R2 queuePos 0.5

#create a exponential traffic source and attach it to ud0
set vbr1 [new Application/Traffic/Exponential]
$vbr1 set packetSize_ 128
$vbr1 set burst_time_ 1200ms
$vbr1 set idle_time_ 800ms
$vbr1 set rate_ 64k
$vbr1 attach-agent $udp0
$udp0 set class_ 1

#create a CBR traffic source and attach it to udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 128
```

```
#sending at 25.89MB
$cbr1 set interval_ 0.000039552
$cbr1 attach-agent $udp1
$udp1 set class_ 3
#create a CBR traffic source and attach it to udp2
set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize_ 128
#sending it at 25.91MB
$cbr2 set interval_ 0.000039521
$cbr2 attach-agent $udp2
$udp2 set class_ 4
#create a CBR traffic source and attach it to udp3
set cbr3 [new Application/Traffic/CBR]
$cbr3 set packetSize_ 128
# sending at 25.92MB
$cbr3 set interval_ 0.000039506
$cbr3 attach-agent $udp3
$udp3 set class_ 5
#create a CBR traffic source and attach it to udp4
set vbr2 [new Application/Traffic/Exponential]
$vbr2 set packetSize_ 128
$vbr2 set burst_time_ 880ms
$vbr2 set idle_time_ 1280ms
$vbr2 set rate_ 64k
$vbr2 attach-agent $udp4
$udp4 set class_ 2
#create a CBR traffic source and attach it to udp5
set cbr4 [new Application/Traffic/CBR]
$cbr4 set packetSize_ 128
#sending at 25.89MB
```

```
$cbr4 set interval_ 0.000039552
$cbr4 attach-agent $udp5
$udp5 set class_ 3
#create a CBR traffic source and attach it to udp6
set cbr5 [new Application/Traffic/CBR]
$cbr5 set packetSize_ 128
#sending it at 25.91MB
$cbr5 set interval_ 0.000039521
$cbr5 attach-agent $udp6
$udp6 set class_ 4
#create a CBR traffic source and attach it to udp7
set cbr6 [new Application/Traffic/CBR]
$cbr6 set packetSize_ 128
# sending at 25.92MB
$cbr6 set interval_ 0.000039506
$cbr6 attach-agent $udp7
$udp7 set class_ 5
#Use LossMonitor agents instead of Null agents as sinks to allow collection of data
set sink0 [new Agent/Null] ;
$ns attach-agent $U0 $sink0
set sink1 [new Agent/Null] ;
$ns attach-agent $U1 $sink1
set sink2 [new Agent/Null] ;
$ns attach-agent $U2 $sink2
set sink3 [new Agent/Null] ;
$ns attach-agent $U3 $sink3
set sink4 [new Agent/Null] ;
$ns attach-agent $U4 $sink4
set sink5 [new Agent/Null] ;
$ns attach-agent $U5 $sink5
```

```
set sink6 [new Agent/Null] ;
$ns attach-agent $U6 $sink6
set sink7 [new Agent/Null] ;
$ns attach-agent $U7 $sink7
# connecting sinks to the udp agents
$ns connect $udp0 $sink4
$ns connect $udp1 $sink5
$ns connect $udp2 $sink6
$ns connect $udp3 $sink7
$ns connect $udp4 $sink0
$ns connect $udp5 $sink1
$ns connect $udp6 $sink2
$ns connect $udp7 $sink3

#setting up the time for when the simulation start and when certain background traffics turn on
and off and when the users begin to speak
$ns at 0.0 "$vbr1 start"
$ns at 0.0 "$vbr2 start"
$ns at 0.0 "$cbr1 start"
$ns at 0.0 "$cbr4 start"
$ns at 20.0 "$cbr1 stop"
$ns at 20.0 "$cbr4 stop"
$ns at 20.0 "$cbr2 start"
$ns at 20.0 "$cbr5 start"
$ns at 40.0 "$cbr5 stop"
$ns at 40.0 "$cbr2 stop"
$ns at 40.0 "$cbr3 start"
$ns at 40.0 "$cbr6 start"
$ns at 55.0 "$cbr6 stop"
$ns at 55.0 "$cbr3 stop"
$ns at 55.0 "$vbr1 stop"
```

```
$ns at 55.0 "$vbr2 stop"
```

```
$ns at 60.0 "finish"
```

```
$ns run
```

TCP:

```
#starting the simulator
```

```
set ns [new Simulator]
```

```
set nf [open out.nam w]
```

```
$ns namtrace-all $nf
```

```
#open tracefile
```

```
set nd [open out.tr w]
```

```
$ns trace-all $nd
```

```
proc finish {} {
```

```
global ns nf
```

```
$ns flush-trace
```

```
close $nf
```

```
exec nam out.nam &
```

```
exit 0
```

```
}
```

```
#set the colors of the packets being sent
```

```
$ns color 1 Blue
```

```
$ns color 2 Red
```

```
$ns color 3 black
```

```
$ns color 4 gold
```

```
$ns color 5 orange
```

```
# nodes for users and routers connect
```

```
set U0 [$ns node]
```

```
set U1 [$ns node]
```

```
set U2 [$ns node]
```

```
set U3 [$ns node]
```

```
set U4 [$ns node]
```

```
set U5 [$ns node]
set U6 [$ns node]
set U7 [$ns node]
set R1 [$ns node]
set R2 [$ns node]

#colors for the nodes
$U0 color darkgreen
$U1 color darkgreen
$U2 color darkgreen
$U3 color darkgreen
$U4 color darkgreen
$U5 color darkgreen
$U6 color darkgreen
$U7 color darkgreen
$R1 color purple
$R2 color purple

# labeling the nodes
$U0 label "User 1"
$U1 label "CBR1 L 25.89Mb"
$U3 label "CBR2 L 25.91Mb"
$U3 label "CBR3 L 25.93Mb"
$U4 label "User 2"
$U5 label "CBR1 R 25.89Mb"
$U6 label "CBR2 R 25.91Mb"
$U7 label "CBR3 R 25.92Mb"
$R1 label "router1"
$R2 label "router2"

# shapes for the routers
$R1 shape box
$R2 shape box
```

how connections are established and bandwidth for our lines

\$ns duplex-link \$U0 \$R1 64kb 5ms DropTail

\$ns duplex-link \$U1 \$R1 26Mb 5ms DropTail

\$ns duplex-link \$U2 \$R1 26Mb 5ms DropTail

\$ns duplex-link \$U3 \$R1 26Mb 5ms DropTail

\$ns duplex-link \$U4 \$R2 64kb 5ms DropTail

\$ns duplex-link \$U5 \$R2 26Mb 5ms DropTail

\$ns duplex-link \$U6 \$R2 26Mb 5ms DropTail

\$ns duplex-link \$U7 \$R2 26Mb 5ms DropTail

link between routers

\$ns duplex-link \$R1 \$R2 25.92Mb 50ms DropTail

#more control over the layout

\$ns duplex-link-op \$U0 \$R1 orient right-down

\$ns duplex-link-op \$U1 \$R1 orient right

\$ns duplex-link-op \$U2 \$R1 orient right-up

\$ns duplex-link-op \$U3 \$R1 orient up

\$ns duplex-link-op \$R1 \$R2 orient right

\$ns duplex-link-op \$U4 \$R2 orient left-down

\$ns duplex-link-op \$U5 \$R2 orient left

\$ns duplex-link-op \$U6 \$R2 orient left-up

\$ns duplex-link-op \$U7 \$R2 orient up

creating our TCP agents for the two users speaking

set tcp0 [new Agent/TCP]

\$ns attach-agent \$U0 \$tcp0

\$tcp0 set fid_ 1

set tcp1 [new Agent/TCP]

\$ns attach-agent \$U4 \$tcp1

\$tcp1 set fid_ 2

#creating the UDP agents for background traffic

set udp1 [new Agent/UDP]

```
$ns attach-agent $U1 $udp1
$udp1 set fid_ 3
set udp2 [new Agent/UDP]
$ns attach-agent $U2 $udp2
$udp2 set fid_ 4
set udp3 [new Agent/UDP]
$ns attach-agent $U3 $udp3
$udp3 set fid_ 5
set udp5 [new Agent/UDP]
$ns attach-agent $U5 $udp5
$udp5 set fid_ 6
set udp6 [new Agent/UDP]
$ns attach-agent $U6 $udp6
$udp6 set fid_ 7
set udp7 [new Agent/UDP]
$ns attach-agent $U7 $udp7
$udp7 set fid_ 8
#to find out what is going on in the queue
$ns duplex-link-op $U0 $R1 queuePos 0.5
$ns duplex-link-op $R1 $R2 queuePos 0.5
$ns duplex-link-op $U4 $R2 queuePos 0.5
# creating the file transfer agent
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ftp0 set class_ 1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ftp1 set class_ 1
#create a CBR traffic source and attach it to udp1
set cbr1 [new Application/Traffic/CBR]
```

```
$cbr1 set packetSize_ 128
#sending at 25.89MB
$cbr1 set interval_ 0.000039552
$cbr1 attach-agent $udp1
$udp1 set class_ 3
#create a CBR traffic source and attach it to udp2
set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize_ 128
#sending it at 25.91MB
$cbr2 set interval_ 0.000039521
$cbr2 attach-agent $udp2
$udp2 set class_ 4
#create a CBR traffic source and attach it to udp3
set cbr3 [new Application/Traffic/CBR]
$cbr3 set packetSize_ 128
# sending at 25.92MB
$cbr3 set interval_ 0.000039506
$cbr3 attach-agent $udp3
$udp3 set class_ 5
#create a CBR traffic source and attach it to udp5
set cbr4 [new Application/Traffic/CBR]
$cbr4 set packetSize_ 128
#sending at 25.89MB
$cbr4 set interval_ 0.000039552
$cbr4 attach-agent $udp5
$udp5 set class_ 3
#create a CBR traffic source and attach it to udp6
set cbr5 [new Application/Traffic/CBR]
$cbr5 set packetSize_ 128
#sending it at 25.91MB
```

```
$cbr5 set interval_ 0.000039521
$cbr5 attach-agent $udp6
$udp6 set class_ 4
#create a CBR traffic source and attach it to udp7
set cbr6 [new Application/Traffic/CBR]
$cbr6 set packetSize_ 128
# sending at 25.92MB
$cbr6 set interval_ 0.000039506
$cbr6 attach-agent $udp7
$udp7 set class_ 5
#Use LossMonitor agents instead of Null agents as sinks to allow collection of data
set sink0 [new Agent/TCPSink] ;
$ns attach-agent $U0 $sink0
set sink1 [new Agent/Null] ;
$ns attach-agent $U1 $sink1
set sink2 [new Agent/Null] ;
$ns attach-agent $U2 $sink2
set sink3 [new Agent/Null] ;
$ns attach-agent $U3 $sink3
set sink4 [new Agent/TCPSink] ;
$ns attach-agent $U4 $sink4
set sink5 [new Agent/Null] ;
$ns attach-agent $U5 $sink5
set sink6 [new Agent/Null] ;
$ns attach-agent $U6 $sink6
set sink7 [new Agent/Null] ;
$ns attach-agent $U7 $sink7
# attaching the agents to the sinks
$ns connect $tcp0 $sink4
$ns connect $udp1 $sink5
```

```

$ns connect $udp2 $sink6
$ns connect $udp3 $sink7
$ns connect $tcp1 $sink0
$ns connect $udp5 $sink1
$ns connect $udp6 $sink2
$ns connect $udp7 $sink3

#setting up the time for when the simulation start and when certain background traffics turn on
and off and when the users begin to speak

$ns at 0.0 "$ftp0 start"
$ns at 0.0 "$ftp1 start"
$ns at 0.0 "$cbr1 start"
$ns at 0.0 "$cbr4 start"
$ns at 20.0 "$cbr1 stop"
$ns at 20.0 "$cbr4 stop"
$ns at 20.0 "$cbr2 start"
$ns at 20.0 "$cbr5 start"
$ns at 40.0 "$cbr5 stop"
$ns at 40.0 "$cbr2 stop"
$ns at 40.0 "$cbr3 start"
$ns at 40.0 "$cbr6 start"
$ns at 55.0 "$cbr6 stop"
$ns at 55.0 "$cbr3 stop"
$ns at 60.0 "$ftp0 stop"
$ns at 60.0 "$ftp1 stop"
$ns at 60.0 "finish"

$ns run

```

RTP:

```

# starting the simulator
set ns [new Simulator]

set nf [open out.nam w]

```

```
$ns namtrace-all $nf
#open tracefile
set nd [open out.tr w]
$ns trace-all $nd
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam &
    exit 0
}
#set the colors of the packets being sent
$ns color 1 Blue
$ns color 2 Red
$ns color 3 black
$ns color 4 gold
$ns color 5 orange
# nodes for users and routers connect
set U0 [$ns node]
set U1 [$ns node]
set U2 [$ns node]
set U3 [$ns node]
set U4 [$ns node]
set U5 [$ns node]
set U6 [$ns node]
set U7 [$ns node]
set R1 [$ns node]
set R2 [$ns node]
#colors for the nodes
$U0 color darkgreen
```

```
$U1 color darkgreen
$U2 color darkgreen
$U3 color darkgreen
$U4 color darkgreen
$U5 color darkgreen
$U6 color darkgreen
$U7 color darkgreen
$R1 color purple
$R2 color purple
# labeling the nodes
$U0 label "User 1"
$U1 label "CBR1 L 25.89Mb"
$U3 label "CBR2 L 25.91Mb"
$U3 label "CBR3 L 25.93Mb"
$U4 label "User 2"
$U5 label "CBR1 R 25.89Mb"
$U6 label "CBR2 R 25.91Mb"
$U7 label "CBR3 R 25.92Mb"
$R1 label "router1 "
$R2 label "router2"
# shapes for the routers
$R1 shape box
$R2 shape box
# how connections are established and bandwidth for our lines
$ns duplex-link $U0 $R1 64kb 5ms DropTail
$ns duplex-link $U1 $R1 26Mb 5ms DropTail
$ns duplex-link $U2 $R1 26Mb 5ms DropTail
$ns duplex-link $U3 $R1 26Mb 5ms DropTail
$ns duplex-link $U4 $R2 64kb 5ms DropTail
$ns duplex-link $U5 $R2 26Mb 5ms DropTail
```

```
$ns duplex-link $U6 $R2 26Mb 5ms DropTail
$ns duplex-link $U7 $R2 26Mb 5ms DropTail
# link between routers
$ns duplex-link $R1 $R2 25.92Mb 50ms DropTail
#more control over the layout
$ns duplex-link-op $U0 $R1 orient right-down
$ns duplex-link-op $U1 $R1 orient right
$ns duplex-link-op $U2 $R1 orient right-up
$ns duplex-link-op $U3 $R1 orient up
$ns duplex-link-op $R1 $R2 orient right
$ns duplex-link-op $U4 $R2 orient left-down
$ns duplex-link-op $U5 $R2 orient left
$ns duplex-link-op $U6 $R2 orient left-up
$ns duplex-link-op $U7 $R2 orient up
# creating our udp agents
set rtp0 [new Agent/UDP]
$ns attach-agent $U0 $rtp0
$rtp0 set fid_ 1
set udp1 [new Agent/UDP]
$ns attach-agent $U1 $udp1
$udp1 set fid_ 3
set udp2 [new Agent/UDP]
$ns attach-agent $U2 $udp2
$udp2 set fid_ 4
set udp3 [new Agent/UDP]
$ns attach-agent $U3 $udp3
$udp3 set fid_ 5
set rtp1 [new Agent/UDP]
$ns attach-agent $U4 $rtp1
$rtp1 set fid_ 2
```

```
set udp5 [new Agent/UDP]
$ns attach-agent $U5 $udp5
$udp5 set fid_ 6
set udp6 [new Agent/UDP]
$ns attach-agent $U6 $udp6
$udp6 set fid_ 7
set udp7 [new Agent/UDP]
$ns attach-agent $U7 $udp7
$udp7 set fid_ 8

#to find out what is going on in the queue
$ns duplex-link-op $U0 $R1 queuePos 0.5
$ns duplex-link-op $R1 $R2 queuePos 0.5
$ns duplex-link-op $U4 $R2 queuePos 0.5

#create a exponential traffic source and attach it to RTP0
set vbr1 [new Application/Traffic/Exponential]
$vbr1 set packetSize_ 128
$vbr1 set burst_time_ 1200ms
$vbr1 set idle_time_ 800ms
$vbr1 set rate_ 64k
$vbr1 attach-agent $rtp0
$rtp0 set class_ 1

#create a CBR traffic source and attach it to udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 128
#sending at 25.89MB
$cbr1 set interval_ 0.000039552
$cbr1 attach-agent $udp1
$udp1 set class_ 3

#create a CBR traffic source and attach it to udp2
set cbr2 [new Application/Traffic/CBR]
```

```
$cbr2 set packetSize_ 128
#sending it at 25.91MB
$cbr2 set interval_ 0.000039521
$cbr2 attach-agent $udp2
$udp2 set class_ 4
#create a CBR traffic source and attach it to udp3
set cbr3 [new Application/Traffic/CBR]
$cbr3 set packetSize_ 128
# sending at 25.92MB
$cbr3 set interval_ 0.000039506
$cbr3 attach-agent $udp3
$udp3 set class_ 5
#create a CBR traffic source and attach it to RTP1
set vbr2 [new Application/Traffic/Exponential]
$vbr2 set packetSize_ 128
$vbr2 set burst_time_ 880ms
$vbr2 set idle_time_ 1280ms
$vbr2 set rate_ 64k
$vbr2 attach-agent $rtp1
$rtp1 set class_ 2
#create a CBR traffic source and attach it to udp5
set cbr4 [new Application/Traffic/CBR]
$cbr4 set packetSize_ 128
#sending at 25.89MB
$cbr4 set interval_ 0.000039552
$cbr4 attach-agent $udp5
$udp5 set class_ 3
#create a CBR traffic source and attach it to udp6
set cbr5 [new Application/Traffic/CBR]
$cbr5 set packetSize_ 128
```

```
#sending it at 25.91MB
$cbr5 set interval_ 0.000039521
$cbr5 attach-agent $udp6
$udp6 set class_ 4
#create a CBR traffic source and attach it to udp7
set cbr6 [new Application/Traffic/CBR]
$cbr6 set packetSize_ 128
# sending at 25.92MB
$cbr6 set interval_ 0.000039506
$cbr6 attach-agent $udp7
$udp7 set class_ 5
#Use LossMonitor agents instead of Null agents as sinks to allow collection of data
set sink0 [new Agent/Null] ;
$ns attach-agent $U0 $sink0
set sink1 [new Agent/Null] ;
$ns attach-agent $U1 $sink1
set sink2 [new Agent/Null] ;
$ns attach-agent $U2 $sink2
set sink3 [new Agent/Null] ;
$ns attach-agent $U3 $sink3
set sink4 [new Agent/Null] ;
$ns attach-agent $U4 $sink4
set sink5 [new Agent/Null] ;
$ns attach-agent $U5 $sink5
set sink6 [new Agent/Null] ;
$ns attach-agent $U6 $sink6
set sink7 [new Agent/Null] ;
$ns attach-agent $U7 $sink7
# attaching the sinks to the agents
$ns connect $rtp0 $sink4
```

\$ns connect \$udp1 \$sink5

\$ns connect \$udp2 \$sink6

\$ns connect \$udp3 \$sink7

\$ns connect \$rtp1 \$sink0

\$ns connect \$udp5 \$sink1

\$ns connect \$udp6 \$sink2

\$ns connect \$udp7 \$sink3

#setting up the time for when the simulation start and when certain background traffics turn on

and off and when the users begin to speak

\$ns at 0.0 "\$vbr1 start"

\$ns at 0.0 "\$vbr2 start"

\$ns at 0.0 "\$cbr1 start"

\$ns at 0.0 "\$cbr4 start"

\$ns at 20.0 "\$cbr1 stop"

\$ns at 20.0 "\$cbr4 stop"

\$ns at 20.0 "\$cbr2 start"

\$ns at 20.0 "\$cbr5 start"

\$ns at 40.0 "\$cbr5 stop"

\$ns at 40.0 "\$cbr2 stop"

\$ns at 40.0 "\$cbr3 start"

\$ns at 40.0 "\$cbr6 start"

\$ns at 55.0 "\$cbr6 stop"

\$ns at 55.0 "\$cbr3 stop"

\$ns at 55.0 "\$vbr1 stop"

\$ns at 55.0 "\$vbr2 stop"

\$ns at 60.0 "finish"

\$ns run