

Lab 5 Networks

106119029

Question 1

Code

- Bus topology tcp and udp based on commandline argument

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set all_trace [open all.tr w]
$ns trace-all $all_trace

#Define a 'finish' procedure
proc finish {} {
    global ns nf all_trace
    $ns flush-trace
    #Close the trace file
    close $nf
    close $all_trace
    #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
```

```

set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#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 $n3 $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_ 1mb
$cbr set random_ false

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run

```

- Grid topology tcp and udp based on commandline argument

```

#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set all_trace [open all.tr w]
$ns trace-all $all_trace

#Define a 'finish' procedure
proc finish {} {
    global ns nf all_trace
    $ns flush-trace
    #Close the trace file
    close $nf
    close $all_trace
    #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}

for {set i 0} {$i < 25} {incr i} {
    set n($i) [$ns node]

```

```

}

for {set i 0} {$i < 5} {incr i} {
    for {set j 0} {$j < 4} {incr j} {
        set k1 [expr {5*$i + $j}]
        set k2 [expr {$k1 + 1}]
        $ns duplex-link $n($k1) $n($k2) 2Mb 10ms DropTail
        $ns duplex-link-op $n($k1) $n($k2) orient right
        $ns queue-limit $n($k1) $n($k2) 2
        $ns queue-limit $n($k2) $n($k1) 2
        $ns duplex-link-op $n($k1) $n($k2) queuePos 0.5
    }
}

for {set i 0} {$i < 5} {incr i} {
    for {set j 0} {$j < 4} {incr j} {
        set k1 [expr {$i + 5*$j}]
        set k2 [expr {$k1 + 5}]
        $ns duplex-link $n($k1) $n($k2) 2Mb 10ms DropTail
        $ns duplex-link-op $n($k1) $n($k2) orient down
        $ns queue-limit $n($k1) $n($k2) 2
        $ns queue-limit $n($k2) $n($k1) 2
        $ns duplex-link-op $n($k1) $n($k2) queuePos 0.5
    }
}

set s [lindex $argv 0]
set d [lindex $argv 1]
set type [lindex $argv 2]

if {$type == "tcp"} {

    #Setup a TCP connection
    set tcp [new Agent/TCP]
    $tcp set class_ 2
    $ns attach-agent $n($s) $tcp

    set sink [new Agent/TCPSink]
    $ns attach-agent $n($d) $sink
    $ns connect $tcp $sink
    $tcp set fid_ 1

    #Setup a FTP over TCP connection
    set ftp [new Application/FTP]
    $ftp attach-agent $tcp
    $ftp set type_ FTP

```

```

$ns at 0.5 "$ftp start"
$ns at 4.5 "$ftp stop"

} elseif {$type == "udp"} {

    #Setup a UDP connection
    set udp [new Agent/UDP]
    $ns attach-agent $n($s) $udp
    set null [new Agent/Null]
    $ns attach-agent $n($d) $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_ 3mb
    $cbr set random_ false

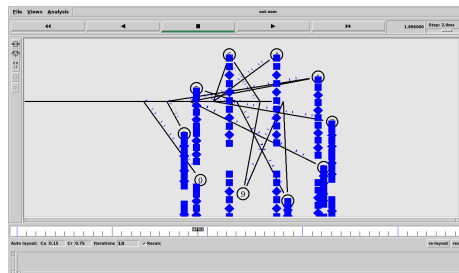
    $ns at 0.5 "$cbr start"
    $ns at 4.5 "$cbr stop"
}
# call finish after 5s
$ns at 5.0 "finish"

# run ns
$ns run

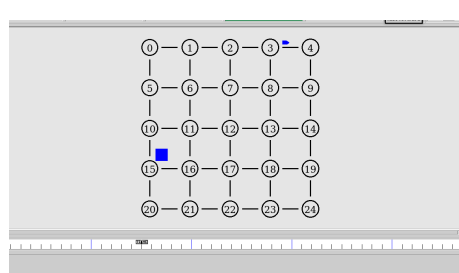
```

Output

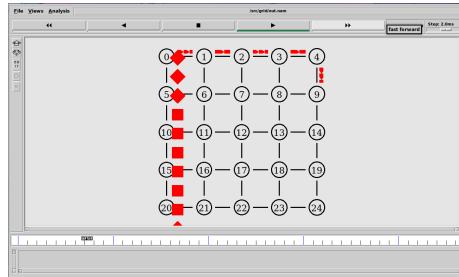
Nam Output(TCP BUS)



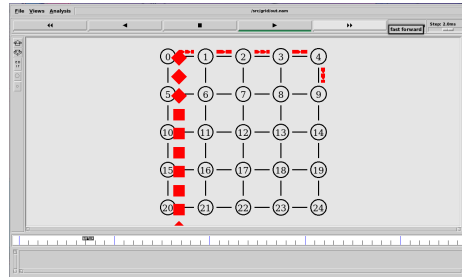
Nam Output (TCP GRID)



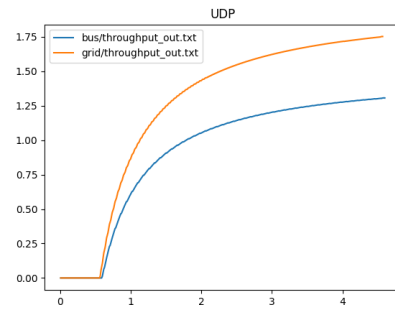
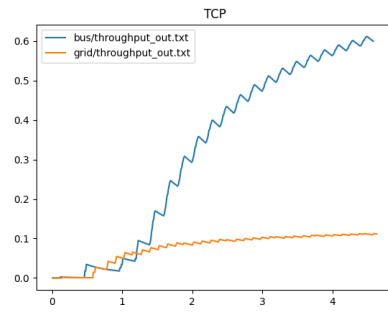
Nam Output(UDP BUS)



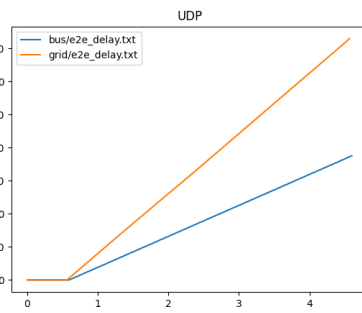
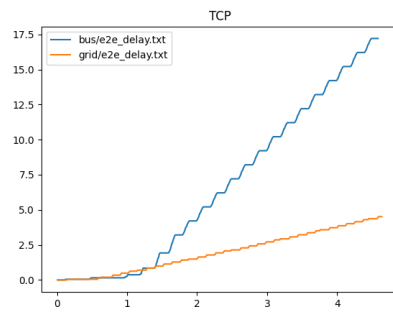
Nam Output (UDP GRID)



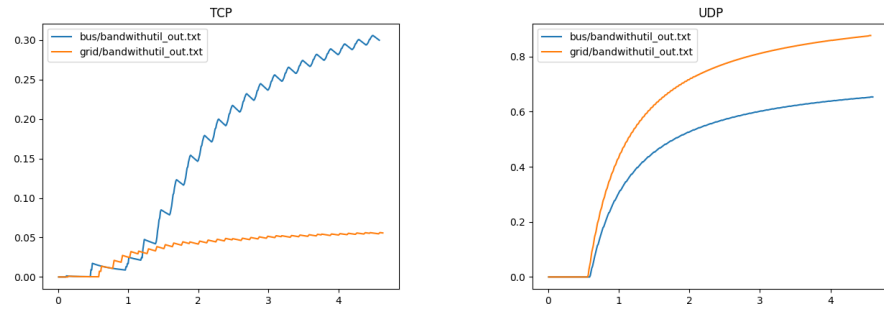
Analysis output(Throughput)



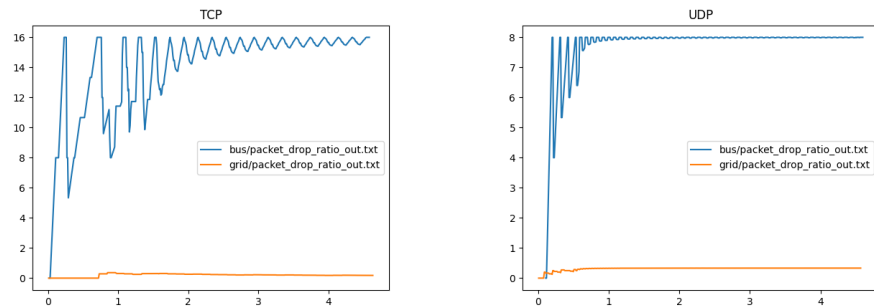
Analysis output(End To End Delay)



Analysis output(Bandwith Utilization)



Analysis output(Packet Drop Ratio)



Question 2

Code

- Chooses Tahoe or Reno based on Command line argument

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf
```

```

set all_trace [open all.tr w]
$ns trace-all $all_trace

#Define a 'finish' procedure
proc finish {} {
    global ns nf all_trace
    $ns flush-trace
    #Close the trace file
    close $nf
    close $all_trace
    #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}

set cols 10
set rows 5
set total [expr $cols * $rows]

for {set i 0} {$i < $total} {incr i} {
    set n($i) [$ns node]
}

for {set i 0} {$i < $rows} {incr i} {
    for {set j 0} {$j < [expr {$cols-1}]} {incr j} {
        set k1 [expr {$cols*$i + $j}]
        set k2 [expr {$k1 + 1}]
        $ns duplex-link $n($k1) $n($k2) 1Mb 10ms DropTail
        $ns duplex-link-op $n($k1) $n($k2) orient right
        $ns queue-limit $n($k1) $n($k2) 10
        $ns queue-limit $n($k2) $n($k1) 10
        $ns duplex-link-op $n($k1) $n($k2) queuePos 0.5
    }
}

for {set i 0} {$i < $cols} {incr i} {
    for {set j 0} {$j < [expr {$rows - 1}]} {incr j} {
        set k1 [expr {$i + $cols*$j}]
        set k2 [expr {$k1 + $cols}]
        $ns duplex-link $n($k1) $n($k2) 1Mb 10ms DropTail
        $ns duplex-link-op $n($k1) $n($k2) orient down
        $ns queue-limit $n($k1) $n($k2) 10
        $ns queue-limit $n($k2) $n($k1) 10
        $ns duplex-link-op $n($k1) $n($k2) queuePos 0.5
    }
}

```



```

}

set type [lindex $argv 0]
set traffic [lindex $argv 1]

set no_of_source [expr {int($traffic*$total/100.0)}]

for {set i 0} {$i < $no_of_source} {incr i} {

    if {$type == "tahoe"} {
        set tcp($i) [new Agent/TCP]
    } elseif {$type == "reno"} {
        set tcp($i) [new Agent/TCP/Reno]
    }

    set test ""

    set s [expr {int(rand()*50)}]
    set test "$test $s"

    set d [expr {int(rand()*50)}]
    set test "$test $d"

    puts "$test"

    $tcp($i) set class_ 2
    $ns attach-agent $n($s) $tcp($i)
    $tcp($i) tracevar cwnd_

    set sink($i) [new Agent/TCPSink]
    $ns attach-agent $n($d) $sink($i)
    $ns connect $tcp($i) $sink($i)
    $tcp($i) set fid_ 1

    #Setup a FTP over TCP connection
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
    $ftp($i) set type_ FTP

    $ns at 0.5 "$ftp($i) start"
    $ns at 49.5 "$ftp($i) stop"
}

```

```

# call finish after 5s
$ns at 50.0 "finish"

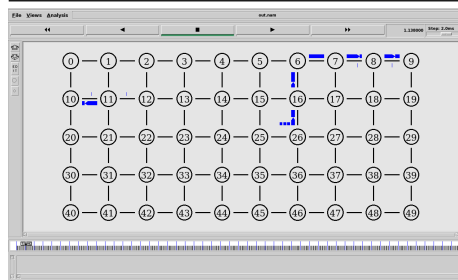
# run ns
$ns run

```

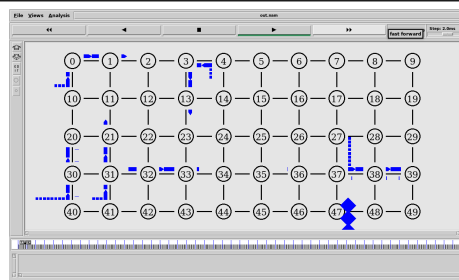
Output

Tahoe Nam

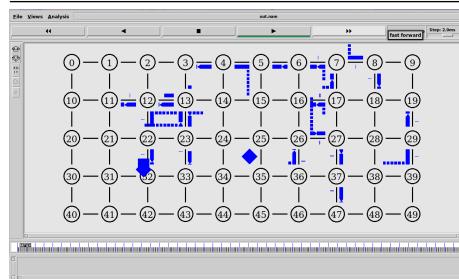
Nam Output (Tahoe Low)



Nam Output (Tahoe Medium)

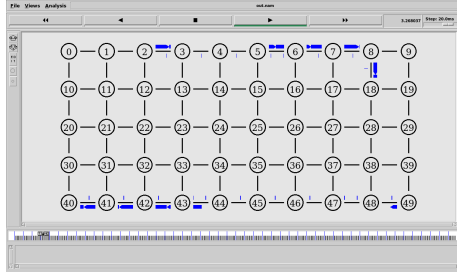


Nam Output (Tahoe High)

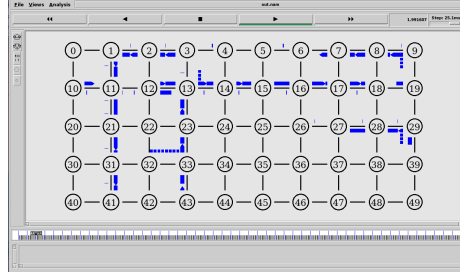


Reno Nam

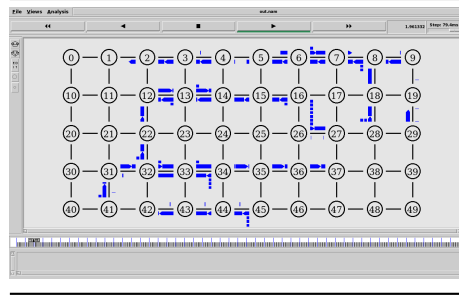
Nam Output(Reno Low)



Nam Output (Reno Medium)

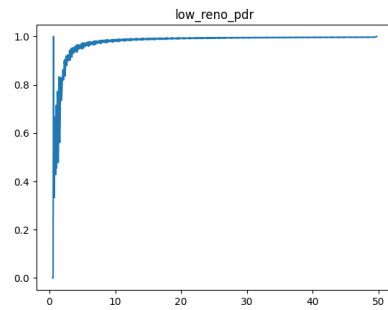


Nam Output (Reno High)

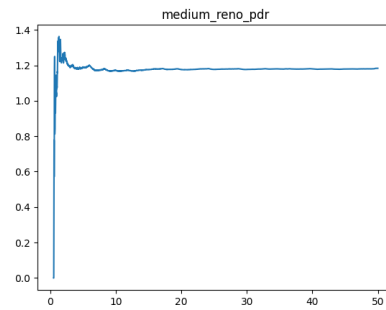


PDR

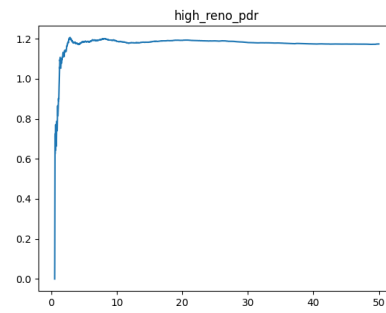
PDR Output(Reno Low)



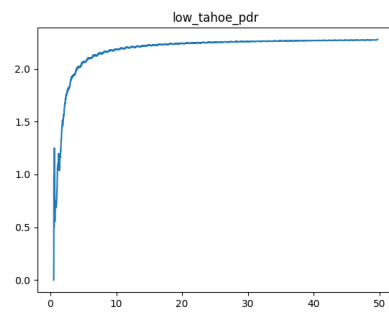
PDR Output (Reno Medium)



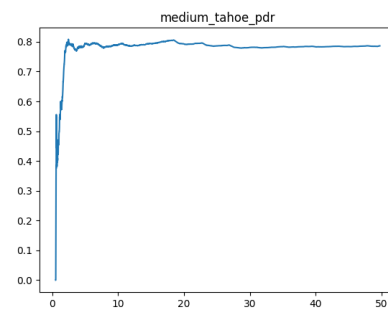
PDR Output (Reno High)



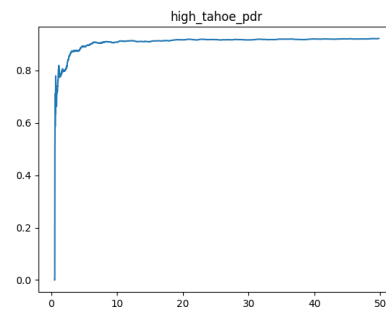
PDR Output(Tahoe Low)



PDR Output (Tahoe Medium)

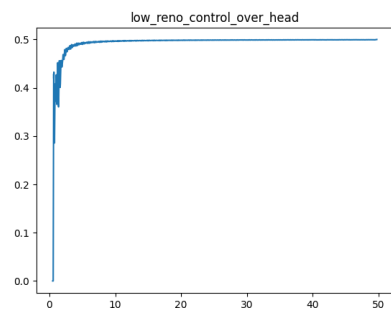


PDR Output (Tahoe High)

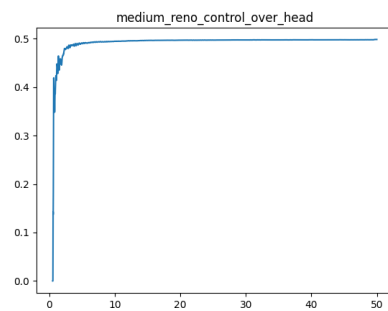


Control Overhead

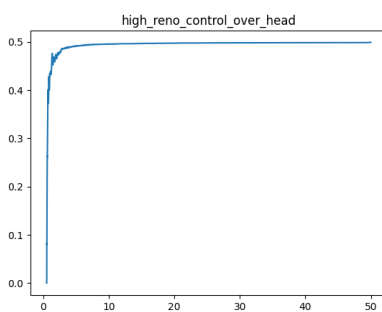
Control Overhead(Reno Low)



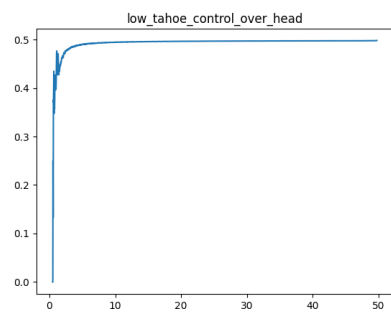
Control Overhead (Reno Medium)



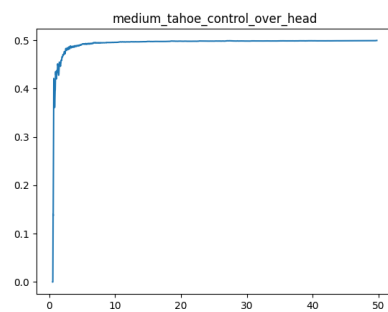
Control Overhead (Reno High)



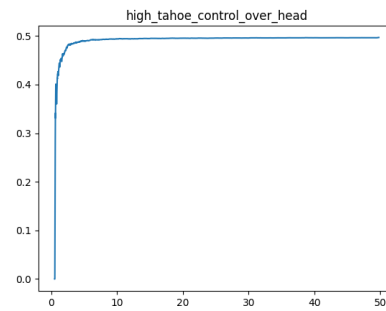
Control Overhead(Tahoe Low)



Control Overhead (Tahoe Medium)

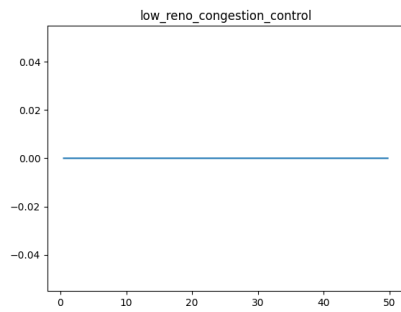


Control Overhead (Tahoe High)

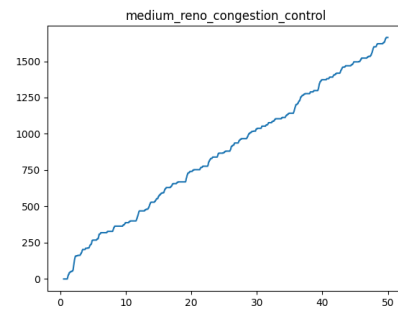


Congestion Control

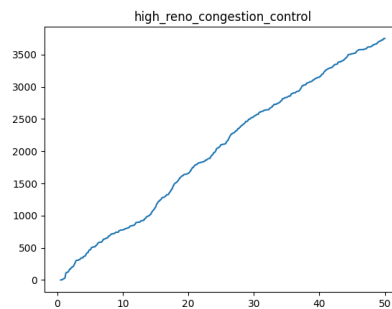
Congestion Control(Reno Low)



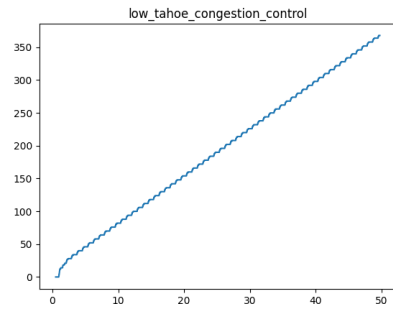
Congestion Control (Reno Medium)



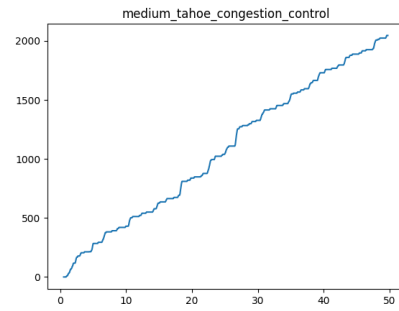
Congestion Control (Reno High)



Congestion Control(Tahoe Low)



Congestion Control (Tahoe Medium)



Congestion Control (Tahoe High)

