**Lab 10**

**Question#1**

**Code:**

set ns [new Simulator]

$ns rtproto DV

set nf [open out.nam w]

$ns namtrace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam

exit 0

}

# Creating Nodes

for {set i 0} {$i < 7} {incr i} {

set n($i) [$ns node]

}

# Creating Links

for {set i 0} {$i < 7} {incr i} {

$ns duplex-link $n($i) $n([expr ($i + 1) % 7]) 512Kb 5ms DropTail

}

$ns duplex-link-op $n(0) $n(1) queuePos 1

$ns duplex-link-op $n(0) $n(6) queuePos 1

# Creating UDP agent and attaching to node 0

set udp0 [new Agent/UDP]

$ns attach-agent $n(0) $udp0

# Creating Null agent and attaching to node 3

set null0 [new Agent/Null]

$ns attach-agent $n(3) $null0

$ns connect $udp0 $null0

# Creating a CBR agent and attaching it to udp0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 1024

$cbr0 set interval\_ 0.01

$cbr0 attach-agent $udp0

$ns rtmodel-at 0.4 down $n(2) $n(3)

$ns rtmodel-at 1.0 up $n(2) $n(3)

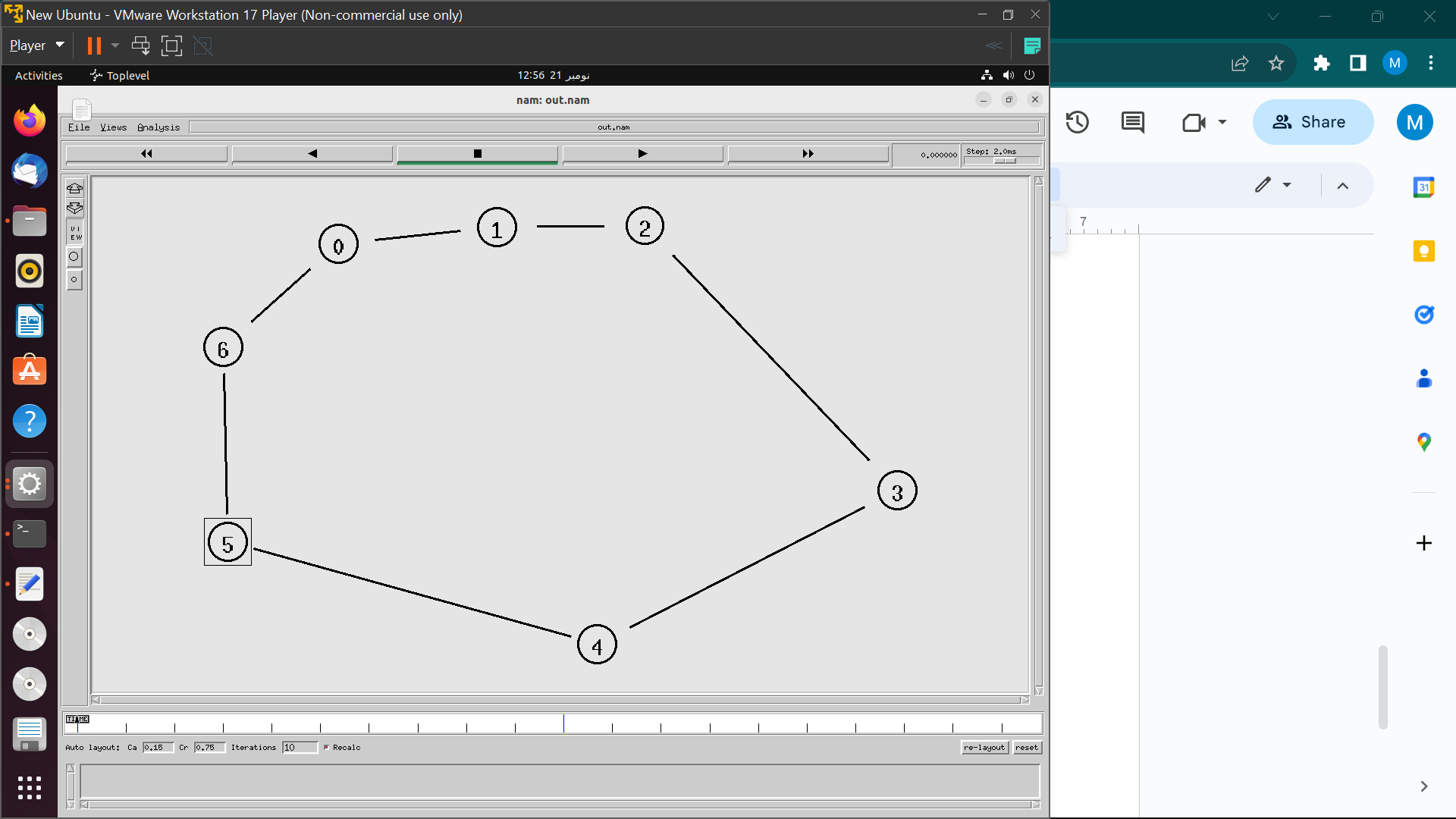
$ns at 0.01 "$cbr0 start"

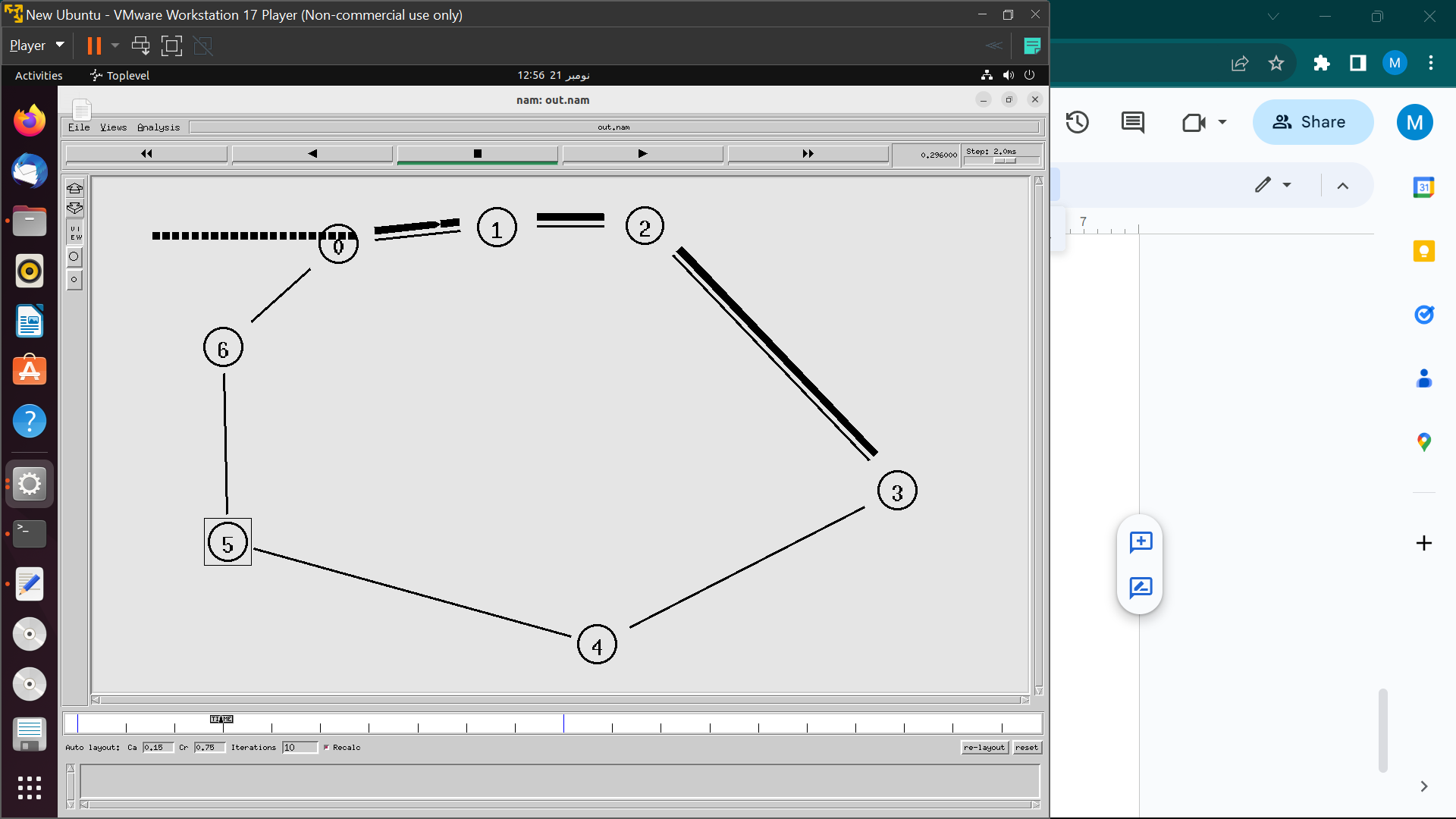
$ns at 1.5 "$cbr0 stop"

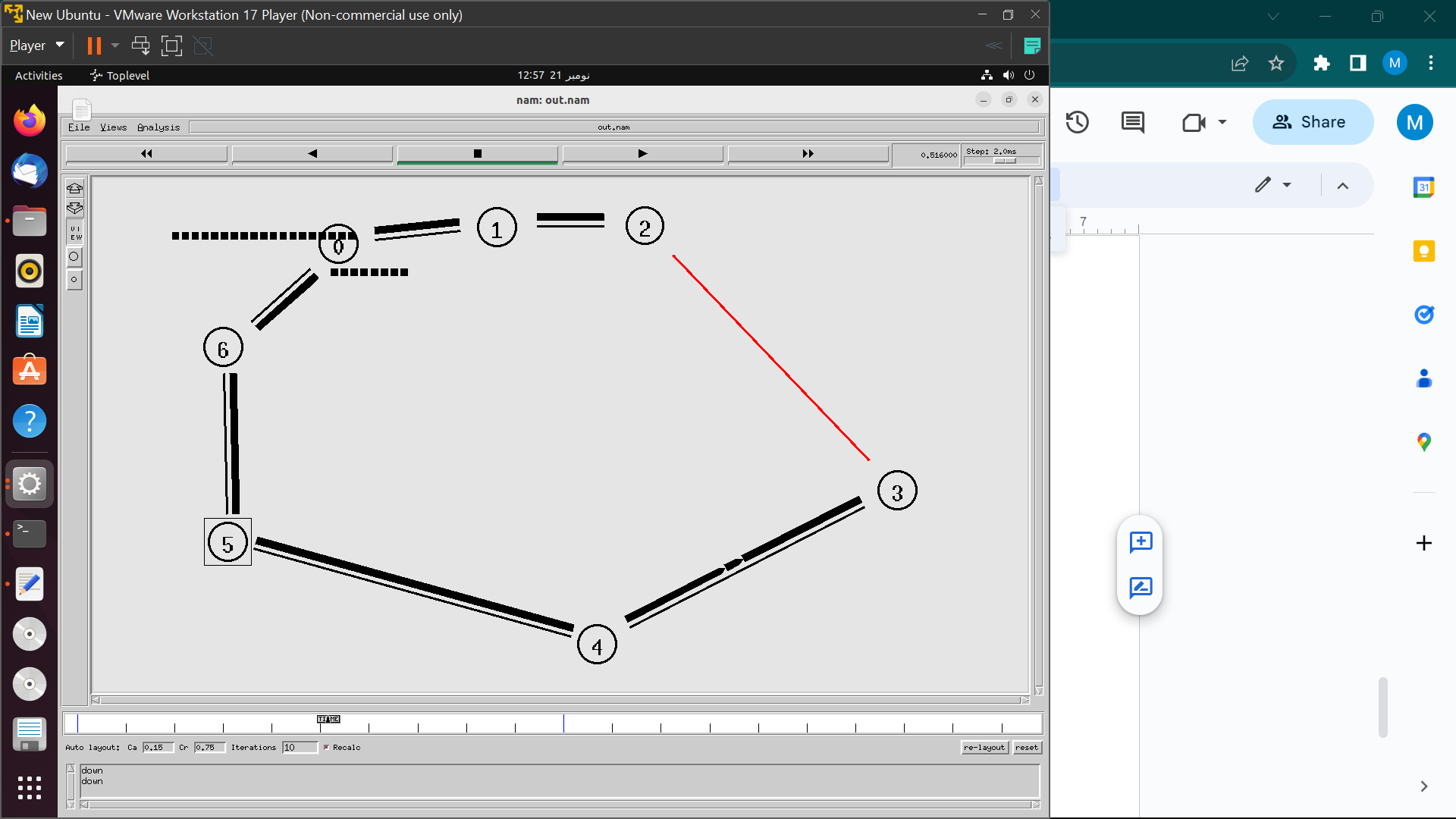
$ns at 2.0 "finish"

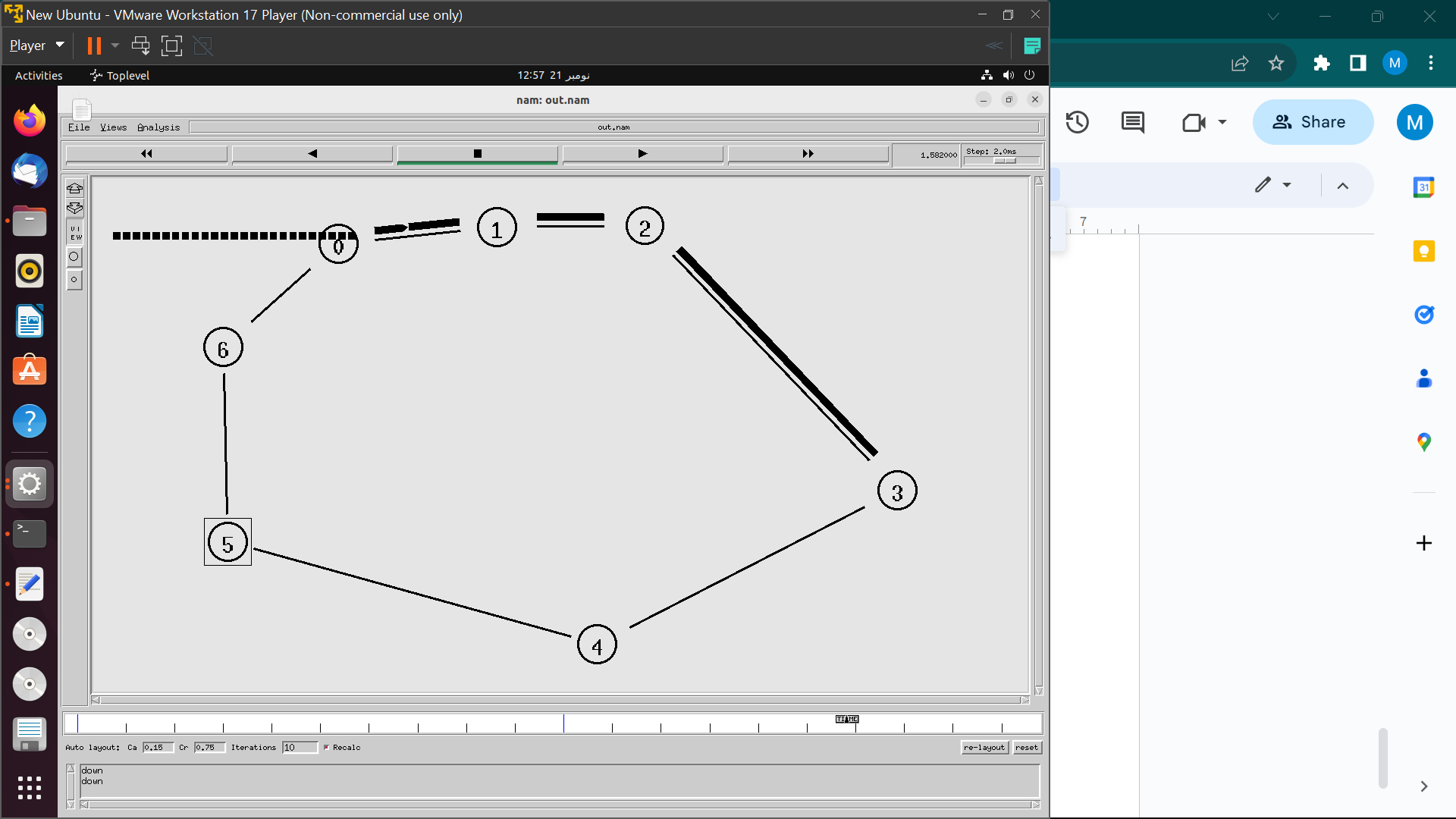
$ns run

**Screenshot:**









a. Initial Path:Initially, packets follow a ring topology established in the for loop, where each node (0-6) is connected to its adjacent nodes in a circular manner.

b. Path After Link Failure:

The code doesn't simulate link failure explicitly. In the absence of a specific failure event, packets would continue to follow the ring topology.

c. Path After Removing Distance Vector Algorithm:

Without the Distance Vector Algorithm, packets would still follow the ring, as the simulator likely defaults to the Shortest Path First algorithm, routing through the shortest path between nodes.

**Question#2**

**Code:**

set ns [new Simulator]

$ns color 1 blue

$ns color 2 red

$ns rtproto DV

set nf [open out.nam w]

$ns namtrace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam

exit 0

}

# Creating Nodes

for {set i 0} {$i < 7} {incr i} {

set n($i) [$ns node]

}

# Creating Links

for {set i 1} {$i < 7} {incr i} {

$ns duplex-link $n(0) $n($i) 512Kb 10ms SFQ

}

# Orienting The nodes

$ns duplex-link-op $n(0) $n(1) orient left-up

$ns duplex-link-op $n(0) $n(2) orient right-up

$ns duplex-link-op $n(0) $n(3) orient right

$ns duplex-link-op $n(0) $n(4) orient right-down

$ns duplex-link-op $n(0) $n(5) orient left-down

$ns duplex-link-op $n(0) $n(6) orient left

# TCP\_Config

set tcp0 [new Agent/TCP]

$tcp0 set class\_ 1

$ns attach-agent $n(1) $tcp0

set sink0 [new Agent/TCPSink]

$ns attach-agent $n(4) $sink0

$ns connect $tcp0 $sink0

# UDP\_Config

set udp0 [new Agent/UDP]

$udp0 set class\_ 2

$ns attach-agent $n(2) $udp0

set null0 [new Agent/Null]

$ns attach-agent $n(5) $null0

$ns connect $udp0 $null0

# CBR Config

set cbr0 [new Application/Traffic/CBR]

$cbr0 set rate\_ 256Kb

$cbr0 attach-agent $udp0

# FTP Config

set ftp0 [new Application/FTP]

$ftp0 attach-agent $tcp0

# Scheduling Events

$ns rtmodel-at 0.5 down $n(0) $n(5)

$ns rtmodel-at 0.9 up $n(0) $n(5)

$ns rtmodel-at 0.7 down $n(0) $n(4)

$ns rtmodel-at 1.2 up $n(0) $n(4)

$ns at 0.1 "$ftp0 start"

$ns at 1.5 "$ftp0 stop"

$ns at 0.2 "$cbr0 start"

$ns at 1.3 "$cbr0 stop"

$ns at 2.0 "finish"

$ns run

**Screenshot:**

