Install NS2 in ubuntu 18.04

1. sudo apt update
2. sudo apt install build-essential autoconf automake libxmu-dev
3. Download ns-allinone-2.35 in ubuntu 18.04
4. tar zxvf ns-allinone-2.35.tar.gz
5. cd ns-allinone-2.35
6. gcc –version
7. g++ –version
8. sudo nano /etc/apt/sources.list
9. deb http://in.archive.ubuntu.com/ubuntu bionic main universe
10. sudo apt update
11. sudo apt install gcc-4.8 g++-4.8
12. Try ./install (Error)
13. cd ns-2.35
14. gedit Makefile.in (…. CC = @CC@, CPP = @CXX@) CC = gcc -4.8, CPP = g++-4.8
15. gedit linkstate/ls.h (void eraseAll() {this->erase})
16. ~nam-1.15/Makefile.in
17. ~xgraph-12.2/Makefile.in
18. ~otcl-1.14/Makefile.in
19. cd ..
20. ./install
21. Ns-allinone package has been installed successfully
22. gedit /home/preeti/ .bashrc
23. export PATH=$PATH:/home/preeti
    /ns-allinone-2.35/bin:/home/preeti
    /ns-allinone-2.35/tcl8.5.10/unix
    :/home/preeti/ns-allinone-2.35/tk8.5.10/unix
24. export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/preeti
    /ns-allinone-2.35/otcl-1.14:/home/preeti/ns-allinone-2.35/lib
25. source /home/preeti/ .bashrc
26. Ns
27. %
28. Nam
29. Sudo apt install nam
30. Xgraph

Write an example of a Network Simulator 2 (NS2) Tcl script that sets up a simple network simulation involving two nodes, a UDP agent, and a Constant Bit Rate (CBR) traffic generator? The script should include trace file generation, scheduling of events, and visualization of the simulation results using NAM.

```
set ns [new Simulator]
set tracefile [open out.tr w]
$ns trace-all $tracefile
set nf [open out.nam w]
$ns namtrace-all $nf

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

set n0 [$ns node]
set n1 [$ns node]
$ns simplex-link $n0 $n1 1Mb 10ms DropTail

set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $udp0 $null0

$ns at 1.0 "$cbr start"
$ns at 3.0 "finish"

$ns run
```

Write an NS2 simulation script to set up a network with five nodes, configure UDP and TCP traffic with CBR and FTP applications, and generate a NAM trace file, including scheduling events and defining a procedure to finalize the simulation.

```
# myfirst_ns.tcl
# Create a Simulator
set ns [new Simulator]

# Create a trace file
set mytrace [open out.tr w]
$ns trace-all $mytrace

# Create a NAM trace file
set myNAM [open out.nam w]
$ns namtrace-all $myNAM

# Define a procedure finish
proc finish { } {
global ns mytrace myNAM
$ns flush-trace
close $mytrace
close $myNAM
exec nam out.nam &
exit 0
}
# Create Nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

# Connect Nodes with Links
$ns duplex-link $n0 $n2 100Mb 5ms DropTail
$ns duplex-link $n1 $n2 100Mb 5ms DropTail
$ns duplex-link $n2 $n4 54Mb 10ms DropTail
$ns duplex-link $n2 $n3 54Mb 10ms DropTail
$ns simplex-link $n3 $n4 10Mb 15ms DropTail
$ns queue-limit $n2 $n3 40


# Create a UDP agent
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
```

```
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 1

#Create a CBR traffic source
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set packetSize_ 1000
$cbr set rate_ 2Mb

#Create a TCP agent
set tcp [new Agent/TCP]
$ns attach-agent $n1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 2

# Create an FTP session
set ftp [new Application/FTP]
$ftp attach-agent $tcp

# Schedule events
$ns at 0.05 "$ftp start"
$ns at 0.1 "$cbr start"
$ns at 60.0 "$ftp stop"
$ns at 60.5 "$cbr stop"
$ns at 61 "finish"

# Start the simulation
$ns run
```

Create a NS2 simulation script that accomplishes the following:

Setup a Network Simulator:
Instantiate a Simulator object.
Define two colors for NAM visualization: Blue for one data flow and Red for another.

Configure NAM Trace:
Open a NAM trace file named out.nam for writing.
Define a procedure called finish to close the trace file, execute NAM to visualize the trace, and exit the simulation.

Define Network Nodes and Links:
Create four nodes in the simulation.
Establish duplex links between these nodes with specified bandwidth, delay, and queue parameters.

Node Placement and Link Attributes:
Set node orientations and link queue positions for NAM visualization.

Setup TCP and FTP Connections:
Create a TCP agent, attach it to a node, and connect it to a TCP Sink agent on another node.
Set up an FTP application over the TCP connection and schedule its start and stop times.

Setup UDP and CBR Connections:
Create a UDP agent, attach it to a different node, and connect it to a Null agent on another node.
Configure a CBR (Constant Bit Rate) application over the UDP connection, specifying packet size and rate.

Schedule Events:
Schedule the start and stop times for the CBR and FTP applications.
Detach TCP and Sink agents after the simulation period.

Output Configuration:
Print out the CBR packet size and interval before running the simulation.

Run the Simulation:
Execute the simulation script.

```
#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

#Define a 'finish' procedure
proc finish {} {
        global ns nf
        $ns flush-trace
        #Close the NAM trace file
        close $nf
        #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"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

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

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
```

```
puts "CBR interval = [$cbr set interval_]"

#Run the simulation
$ns run
```