

Assignment B2

Title: Traffic monitoring for a given topology using NS2.

Software and Hardware Requirements:

NS2 tool, Fedora 20 with Pentium IV and other
1GB RAM, 120 GB HDD,
Monitor, Keyboard, Mouse.

Learning Objectives:

- (i) To understand the use of NS2 tool.
- (ii) To understand how to monitor traffic for a given network using NS2 tool.

Outcomes:

- (i) Students should be able to use NS2 and monitor traffic for a given network using it.

Theory:

For monitoring traffic for a given topology -
1. Create a simulator object.
set ns [new simulator]

Tell simulator to use dynamic routing
\$ ns setproto DV.

Open name trace file
set nf [open cut-nam w]
\$ ns nametrace -all \$ nf


```

# Define a 'finish' procedure
proc finish() {
    global ns nf
    $ns flush-trace
    # close tracefile
    close $nf
    # exec name on tracefile
    exec name out-nam &
    exit 0
}

```

```

for (set i 0) ($ i < 7) { inc i } {
    $ns duplex-link $n($i)
    $n (expr [$i + 1] = 7)
    1 Mb 10 ms Drop.
}

```

```

# Create UDP agent and attach it to
node n(0)
$ns attach-agent $n(0) $udp 0.

```

```

# Create a CBR traffic source and attach it to
UDP 0.

```

```

# Create null agent (traffic sink) and attach
it to node(n3).
$ns attach agent
$n(3) $null.

```

```

# Connect the traffic source with the traffic sink.
$ns connect
$udp 0 $null 0.

```


Schedule events for CBR agent and network dynamics.

Run
\$ ns run.

Network Simulation (NS) is one of the types of simulation, which is used to simulate the networks such as in MANETs, VANETs etc.

It provides simulation for routing and multicast protocols for both wired and wireless networks.

NS is licensed for use under version 2 of the GNU (General Public License) and is popularly known as NS2.

It is an object-oriented, discrete event-driven simulator written in C++ and Otcl/tcl.

Conclusion:

Thus the network topology was created and traffic was monitored successfully using NS2.