Simple NS-2 Network Simulation

Chaitanya Talware (MIS No: 712422005)
Yogesh Toshniwal (MIS No: 712422021)

1 Introduction

This document presents a simple NS-2 network simulation scenario. The simulation setup models a network with five nodes, including both TCP and UDP traffic. The goal is to understand how network connections, such as TCP, FTP, and CBR, can be simulated in NS-2. The network topology and traffic flow are configured using TCL scripting, and the simulation results are visualized using NAM.

2 Simulation Setup

2.1 Network Topology

The network consists of five nodes, labeled from $n0$ to $n4$. The nodes are connected as follows:

- Node $n0$ is connected to node $n2$ with a 2Mb, 10ms link.
- Node $n1$ is connected to node $n2$ with a 2Mb, 10ms link.
- Node $n2$ is connected to node $n3$ with a 1.7Mb, 20ms link.
- Node $n3$ is connected to node $n4$ with a 2Mb, 15ms link.

The link between nodes $n2$ and $n3$ has a queue size of 10 packets. Each link is configured with a 'DropTail' queueing mechanism. Additionally, positions for NAM visualization are defined for better clarity in the network topology.

2.2 Traffic Flow

The simulation involves two types of traffic:

- **TCP traffic**: A TCP connection between nodes $n0$ and $n4$ using FTP as the application layer.
- **UDP traffic**: A CBR (Constant Bit Rate) connection between nodes $n1$ and $n4$.

The traffic flows are defined with packet sizes, rates, and start/stop times for both the CBR and FTP applications.

2.3 TCL Script

The following TCL script is used to define and execute the simulation.

```
# 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 five nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$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
$ns duplex-link $n3 $n4 2Mb 15ms 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
$ns duplex-link-op $n3 $n4 orient right-up

# 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 $n4 $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 $n4 $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
    $n4 $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_]"

# for Running the simulation
$ns run
```

2.4 Explanation of Lines and Flow Representation in NAM

In NAM (Network Animator), the lines connecting the nodes represent the communication links between them. These links are essential for understanding how data flows through the network. The different types of lines in NAM represent various network characteristics and states:

- **Solid Lines (Duplex Links)**: Solid lines indicate active communication links between nodes. They represent duplex (bi-directional) links, meaning data can travel in both directions between two nodes. These lines are typically shown between the nodes where data flows.
- **Dashed Lines**: Dashed lines represent links that are not currently in use but are available in the simulation. These may represent idle or inactive links where no data transmission is occurring.
- **Arrowed Lines**: Arrowed lines in NAM indicate the direction of data flow. The arrows show the direction in which data packets are transmitted across the link. This is particularly useful to differentiate the source and destination of traffic in a network simulation.

The following diagram shows a simple example of the different lines and how they represent the flow of data in a network simulation:

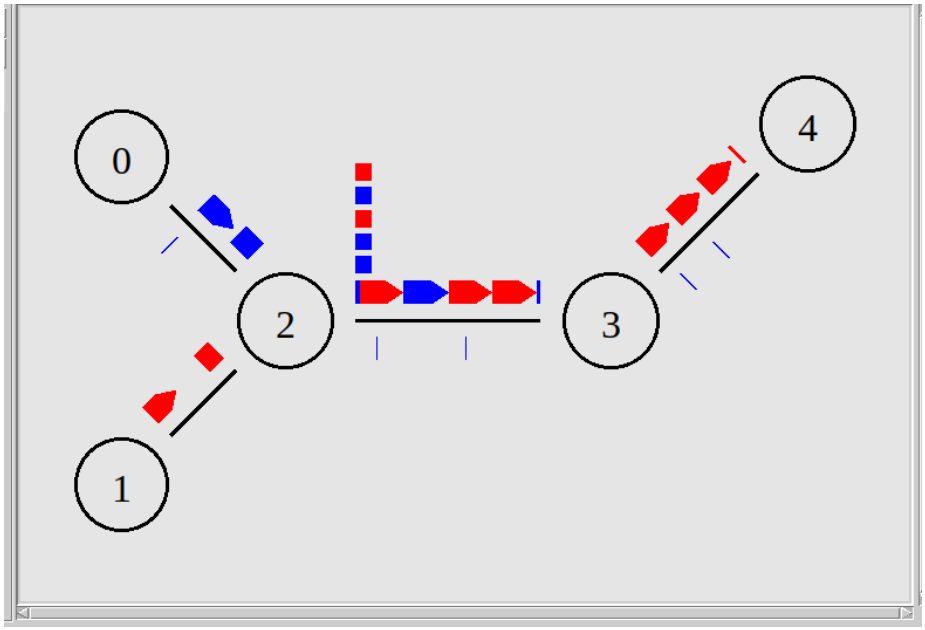

Figure 1: Representation of communication links in NAM

In this diagram:

- **Blue Solid Line**: Indicates an active communication link between Node 1 and Node 3 (duplex).
- **Red Arrowed Line**: Shows the direction of data flow from Node 2 to Node 3.
- **Dashed Line**: Represents a link between Node 3 and Node 4 that is idle and not currently being used.

2.5 Simulation Output

The simulation results are visualized using NAM (Network Animator). The output trace file, `out.nam`, is generated during the simulation run. This file can be opened using the NAM tool to observe the behavior of packets, queues, and traffic flows between the nodes in the simulation.

2.6 Snapshots

2.6.1 Network Topology

The first snapshot shows the network topology, where each node is connected by bidirectional links. The positions of the nodes are strategically defined for clarity in the NAM visualization.

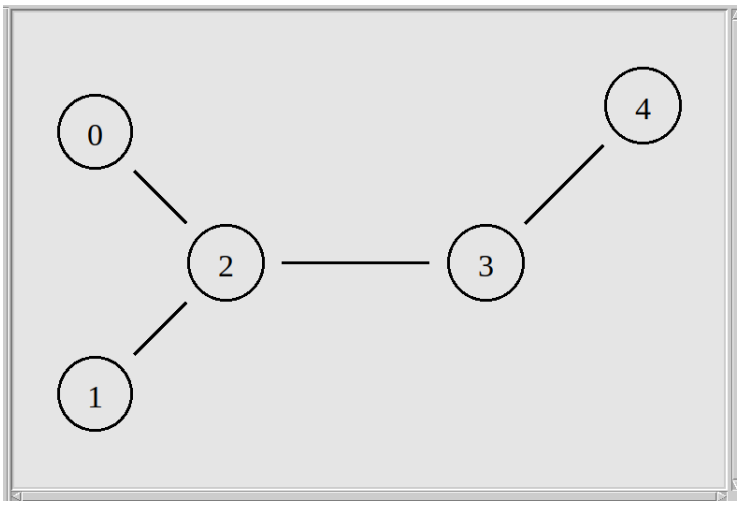

Figure 2: Network Topology: Visualization of the five nodes and their connections.

The image above shows how the five nodes are arranged and interconnected with different bandwidth and delay configurations.

2.6.2 Data Flow from Node $n0$ to Node $n4$

This snapshot shows the flow of data between nodes $n0$ and $n4$, where TCP traffic (FTP application) is used. The data flow is represented with arrows, and you can observe the packet transmission along the path.

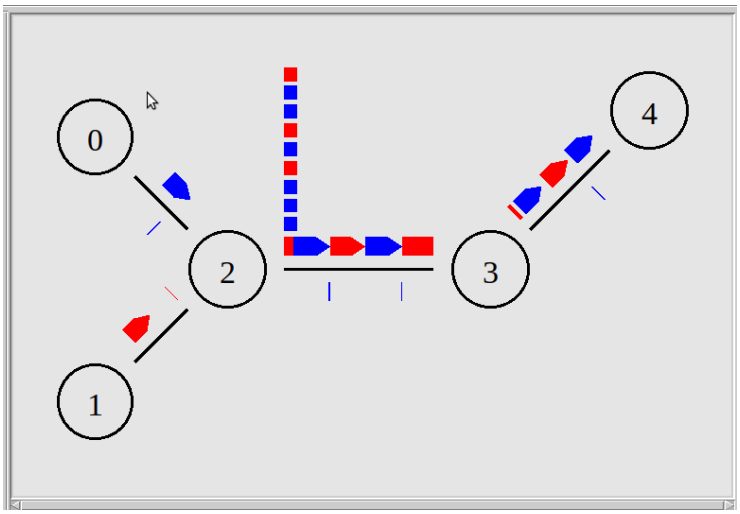

Figure 3: Data Flow from Node $n0$ to Node $n4$: Visualization of FTP data transmission over the TCP connection.

This image demonstrates how packets are transmitted between nodes $n0$ and $n4$ using TCP, with delays and congestion visualized in the network animator.

2.6.3 UDP Traffic and CBR Data Flow

The third snapshot illustrates the UDP traffic between nodes $n1$ and $n4$, where CBR (Constant Bit Rate) is used. This flow is represented with different colors to differentiate it from the TCP flow.

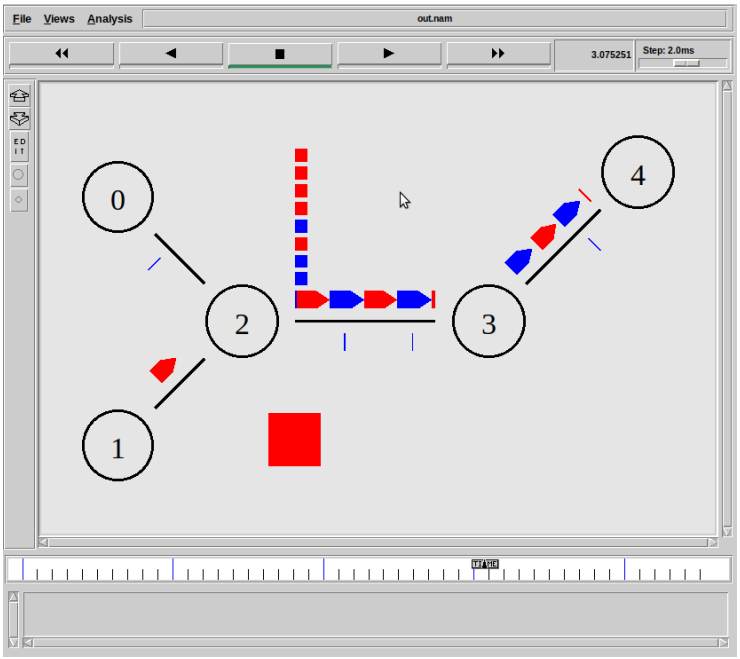

Figure 4: UDP Traffic and CBR Data Flow: Visualization of CBR traffic from Node $n1$ to Node $n4$.

The image above shows the flow of UDP packets with constant bitrate between nodes $n1$ and $n4$. The arrows indicate the transmission of packets at a fixed rate.