CN Lab (17CSL57)

Exp 01
A Simple 3 Node Network

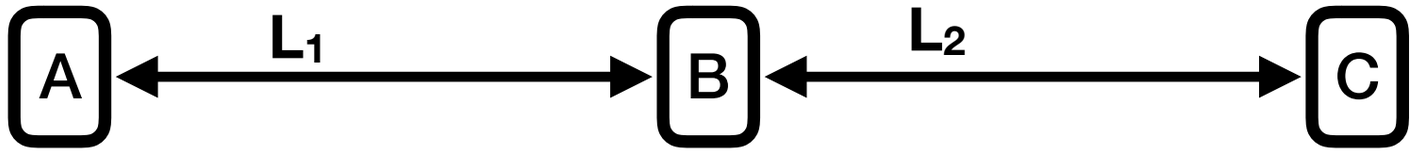
Dr. Ram P Rustagi
Dept of CSE, KSIT
KRP-KSGI
rprustagi@ksit.edu.in

Lab01 Program

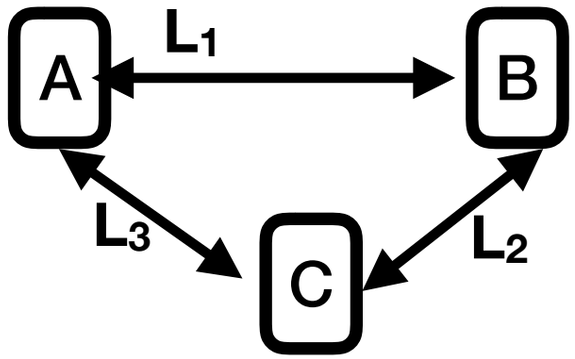
- Program 01
  - Implement three nodes point–to–point network with duplex links between them. Set the queue size, vary the bandwidth, and find the number of packets dropped.

Lab01: Design Parameters

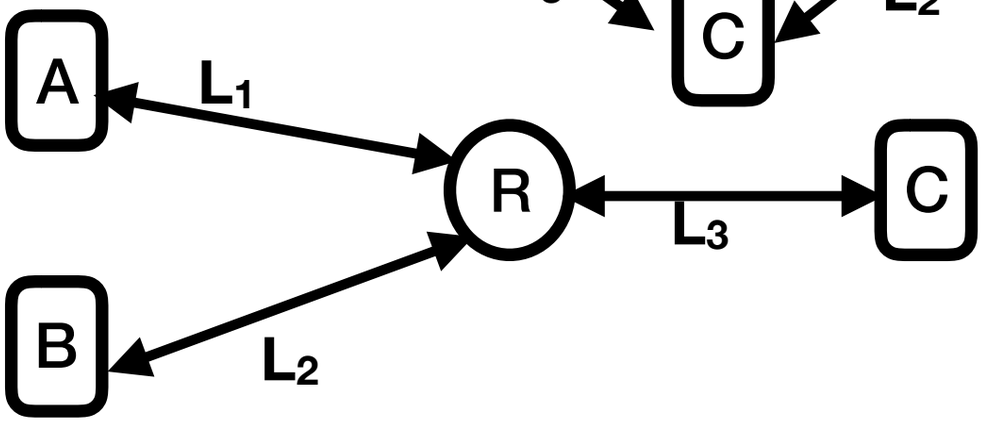
- Design of Lab01: Topologies
  - Many topologies



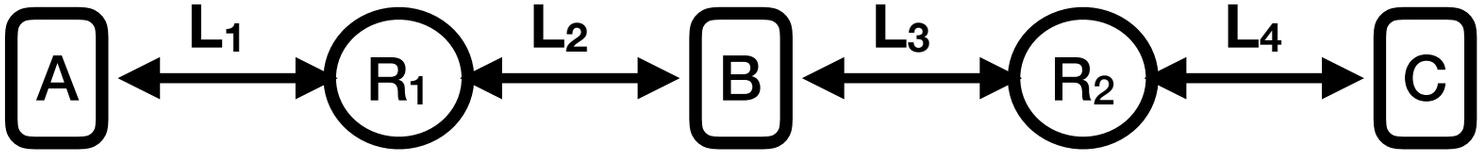
Topo 1



Topo 2



Topo 3



Topo 4

Lab01: Design Parameters

- Parameters: topology, Q size, Bandwidth, application
  - Q Size consideration
    - in conjunction with bandwidth, packet size and packet transmission rate
    - Run experiment with different value of Qsize=1, 2, 4, 8.
    - Different Q sizes for each link vs same Q size for all link
    - Links have different characteristics
      - Bandwidth, propagation delay
    - Use packets of different sizes
      - 500 bytes, 1000 bytes,
    - Duplex links should see traffic in both directions
  - Consider two topologies
    - Topology 1: A⇔B⇔C
    - Topology 2: 3 nodes connected via a router.

Creating Network Topology

- Network consists of nodes and links (simple/duplex)
- Create each of these (as objects)
  – Object numbering starts from 0.
  – May create a dummy node for better correlation
- Connect nodes with links

```
set dummynode [$ns node] #0
set n1 [$ns node] #1
set n2 [$ns node] #2
set n3 [$ns node] #3
#$ns <link_type>  <n1> <n2> <bandwidth> <delay> <qtype>
$ns duplex-link $n1 $n2 4Mb 5ms DropTail
$ns duplex-link $n2 $n3 1Mb 20ms DropTail
```

- Qtype indicates which packet to be dropped
  – DropTail, RED, SFQ, FQ

Queue Limits, Ping AppIn

- Define Q sizes on a link (from one node to next)

```
$ns queue-limit $n1 $n2 3
$ns queue-limit $n2 $n3 2
```

- Default Q size:
  - 50 packets
- Packets will be dropped when as per queueing policy.
  - `DropTail`: last received pkts dropped on queue full

Queue Limits, Ping AppIn

- Ping sender application agents

```
set pings [new Agent/Ping]
set pingr [new Agent/Ping]
```

- Ping application expects response back
- Receiver application must define `recv` procedure

```
Agent/Ping instproc recv {from rtt} {
  $self instvar node_
  puts "node [$node_ id] rcvd ans
 from $from with RTT $rtt msec"
}
```

Queue Limits, Ping AppIn

- Associate the ping application to nodes

```
$ns attach-agent $n1 $pings
$ns attach-agent $n2 $pingr
```

- Connect sender to receiver

```
$ns connect $pings $pingr
```

Generating Ping Traffic…

- List out events at which to send ping traffic

```
$ns at 0.1 "$pings send"
$ns at 0.2 "$pings send"
:
$ns at 0.5 “$pings send”
```

- Execute in a loop if sending large number of packets
  – Define a procedure/function and invoke the same
- Run the simulation for ping application
  – ns simpleping.tcl
- Explore the animation of ping application

Generating Ping Traffic (Loop)

- Execute in a loop if sending large number of packets

```
proc myping {sender cnt gap} {
   global ns
   set num 1
   set now [$ns now]
   set time $now
   while {$cnt > 0} {
      $ns at $time "$sender send"
      set time [expr $time + $gap]
      set cnt [expr $cnt - 1]
   }
}
```

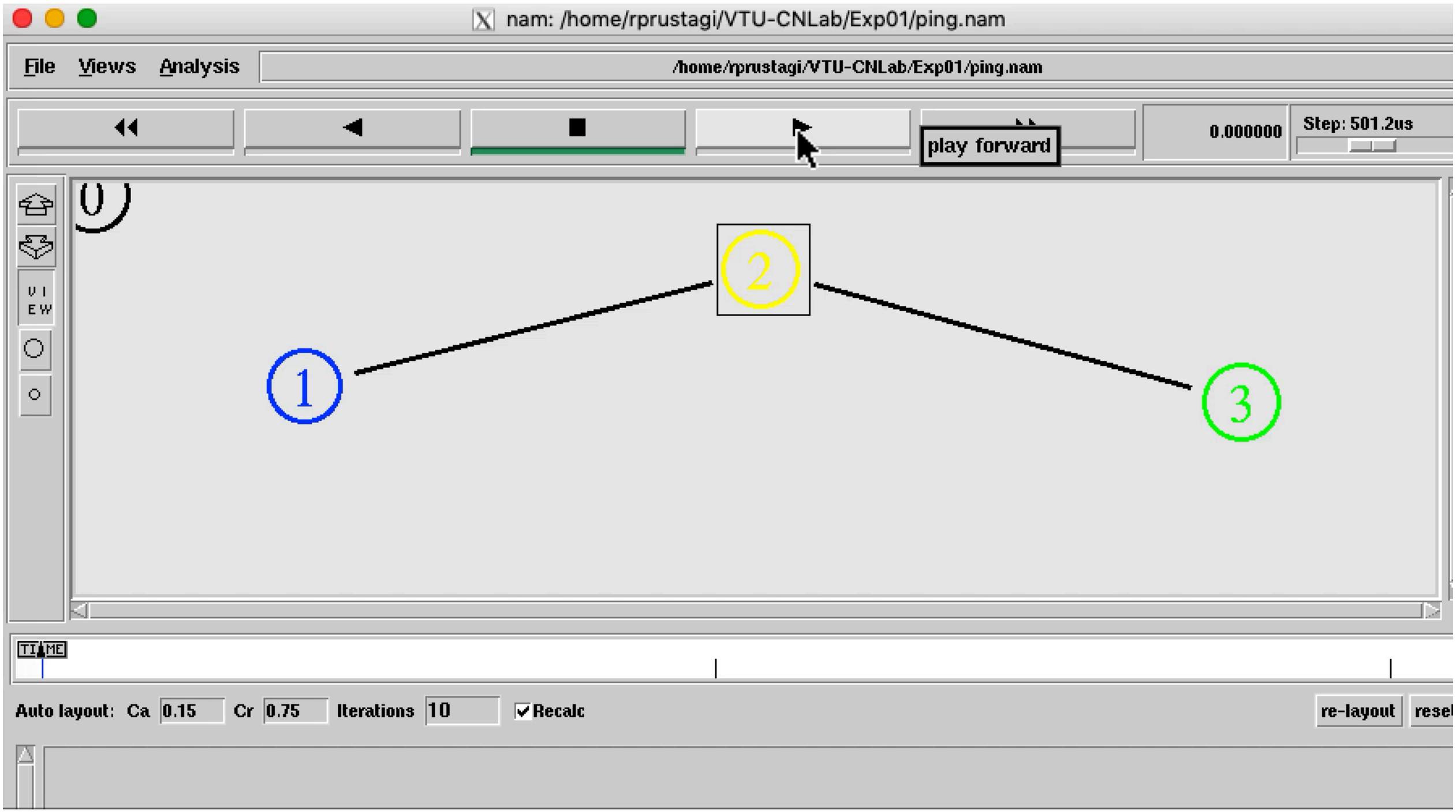
- Invocation

```
myping $pings 10 0.005
```

Experiment 1a

- Topology 1: A⇔B⇔C
- Generate Ping traffic from A to C
  - 1000 bytes packets every 5ms
    - Total of 20 packets
  - Link 1 Parameters
    - Bandwidth (both): 4 Mbps
    - Propagation delays: 5 ms
    - Qlimit: 3
  - Link 2 Parameters
    - Bandwidth (both): 1 Mbps
    - Propagation delays: 20 ms
    - Qlimit: 2
  - Compute packet loss count and
    - verify the same from packet trace.

Animation: experiment 1a



Experiment 1a: Analysis

- Packet size: 1000 bytes = 8000bits
- Packet delivery frequency from node n1: every 5ms
- Total delay on link 1: 2ms + 5ms = 7ms
  - Transmission delay=8000/4*10^6=2ms
  - Propagation delay = 5ms
- Total delay on link 2: 2ms + 5ms = 28ms
  - Transmission delay=8000/1*10^6=8ms
  - Propagation delay = 20ms
- Pkts receive times at node n2 (from n1)
  - 7ms, 12ms, 17ms, 22ms, 27ms, 32ms, ….
- Pkts transmit time at node n2 (to node n3), and Q size
  - 7ms, 15ms, 23ms, 31ms, 39ms, 47ms, …
  - 7ms:1, 12ms: 2, 17ms:2, 22ms:3 (exceeds 2) dropped

Experiment 1b

- Topology 1: A⇔B⇔C
  - Generate Ping traffic from A to C
  - Generate ping traffic from C to A

Experiment 1c

- Topology 1: A⇔B⇔C
  - Generate UDP traffic from A to C
    - 1000 bytes packets every 2ms
      - Total of 10 packets
  - Generate UDP traffic from C to A
    - 2000 bytes packets every 1ms
      - Total of 10 packets
- Link A—B bandwidth: 4Mbps
- Link B—C bandwidth: 2Mbps
- Propagation delays (both links): 1ms
- Q size of both links: 5
- Compute packet loss from above and verify the same from packet trace.

Experiment 1d

- Topology 2: A⇔R⇔B and R⇔C
  - Generate UDP traffic from A to C
    - 1000 bytes packets every 2ms
      - Total of 20 packets
  - Generate UDP traffic from B to C
    - 2000 bytes packets every 1ms
      - Total of 10 packets
- Link A—R bandwidth: 4Mbps
- Link B—R bandwidth: 4Mbps
- Link C—R bandwidth: 2Mbps
- Propagation delays (both links): 1ms
- Q size of all links: 5
- Compute packet loss from above and verify the same from packet trace.

Summary

- Topology of 3 nodes
- Links with different parameters
  – Bandwidth, Propagation delay, Queue size
- Packet drops on exceeding Q size
- Analysis of trace files
  – Packet receive and its subsequent transmission.