

## NETWORKS ASSIGNMENT-4

---

# Packet Capturing Using Wireshark

---

*Prepared by:*

Nikhil Agarwal  
11012323

Himanshu Upreti  
11012315

*Instructors:*

Dr. Sukumar Nandi  
T. Venkatesh

April 13, 2014

## PartA: Initials

1. NS-3 ( [www.nsnam.org](http://www.nsnam.org)) is a discrete event, packet level network simulator for Internet systems. Download and install NS-3. Create a simple topology of two nodes – Node1 and Node2, separated by a point-to-point link. Setup a UdpClient on Node1 and UdpServer on Node2. Start the client application, and measure end to end throughput while varying the latency of the link. Now add another client application to Node1 and a server instance to Node2. What you need to configure to ensure that there is no conflict? Measure end-to-end throughput with the extra client and server application instances. Show screenshots of pcap traces which indicate that delivery is made to the appropriate server instance.

2. Make the following simulations in NS-3

Make a topology of 4 nodes in the following way.

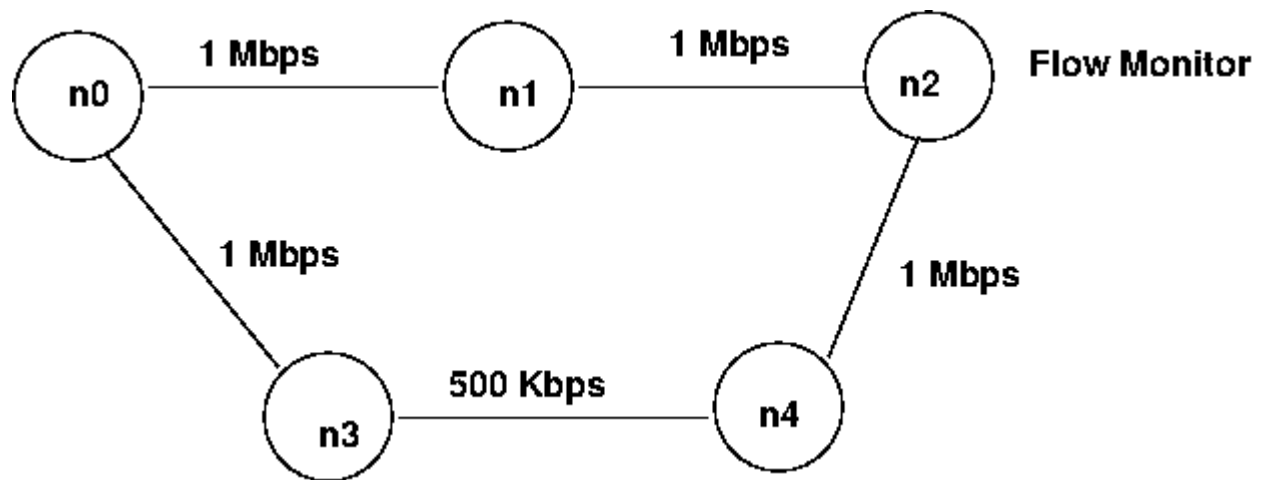
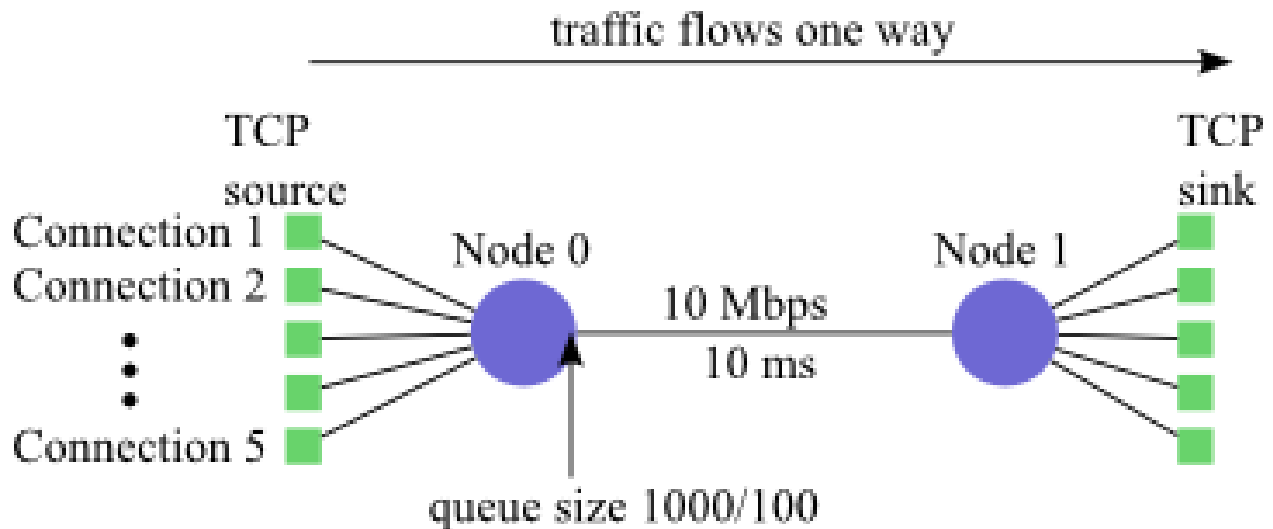


Figure 1: Topology of four nodes

n0 starts CBR traffic at time 1.0 of rate 900 Kbps destined for n2. n0 starts another CBR traffic at time 1.5 of rate 300 Kbps destined for node n1. At time 2.0, link from n0 to n1 goes down. Use a dynamic routing protocol so that path n0-n3-n2 is used now. At time 2.7, link n0-n1 comes up again. At time 3.0, CBR traffic destined for node n1 stops. CBR destined for n2 stops at time 3.5. Use a Flow monitor to monitor losses at n2. Draw a graph of percentage loss as a function of time for the duration of simulation. Give an explanation for results you find.

3. Consider following topology



Connection 1 starts at time 0, Connection 2 at time 5, Connection 3 at time 10, Connection 4 at time 15, and Connection 5 at time 20. End Connection 1 at time 50.1, Connection 2 at time 45, Connection 3 at time 40, Connection 4 at time 35, and Connection 5 at time 30.

Using the generated trace files, produce graphs for the following:

a) The receivers' rates over time. Calculate the rate for each TCP connection separately. Plot the rate for all of the receivers on the same graph. To plot a smooth rate, calculate the rate over a moving window of 10 packets. You can do this by plotting a point every time a packet is received. if  $packet_i$  is received at time  $t_i$ , plot a point at  $t_i$  equal to the sum of the lengths of  $packet_{i-10}$  through  $packet_i$ , divided by the  $t_i - t_{i-10}$ . You will need to modify this slightly to handle the start of the connection, when there are fewer than 10 previous packets.

b) The queue size over time and each packet drop event. You can calculate the queue size at any time by observing all packet enqueue, dequeue, and drop events. Plot each drop event at the maximum queue size when the drop occurs using an "X" symbol.

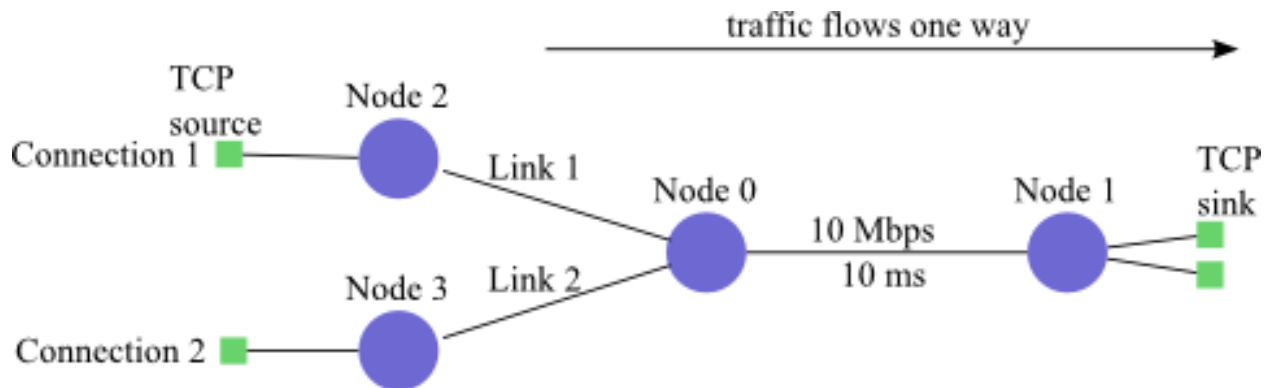
c) The Congestion Window over time for each TCP connection.

Use only TCP NewReno for this experiment. Turn in all of your graphs and then answers to the following questions

a) What do you observe about fairness among the various TCP connections?

b) What do you observe about the queue size and packet drops?

4. Consider the following topology



The queue size limit on all nodes is 10. The bandwidth on Links 1 and 2 is 1.5 Mbps. The propagation delay on Links 1 and 2 is initially 10 ms, but you will be varying the propagation delay on Link 2 for your experiments.

Connection 1 starts at time 0, using Link 1. Connection 2 starts at time 0. Both connections run for 5 seconds.

Plot a graph to show the throughput for each connection over time and observe that they get approximately the same amount. Now run a set of experiments, varying the propagation delay on Link 2 so that Connection 2 has an increasingly longer RTT. Produce the following graph:

The receiver's total rate (averaged over the length of the connection) versus the propagation delay on Link 2. Plot each connection separately.

Use only TCP NewReno for these experiments. What do you observe about the relationship ?

No.	Time	Source	Destination
Protocol Length Info			
9	10.493119000	192.168.169.53	202.141.80.32 HTTP
	372	GET /~sukumar/cs349/HTTP-wireshark-file1.html	HTTP/1.1
Frame 9: 372 bytes on wire (2976 bits), 372 bytes captured (2976 bits) on interface 0			
Ethernet II, Src: HonHaiPr_0a:b0:03 (94:39:e5:0a:b0:03), Dst: Universa_46:1b:10 (e0:2a:82:46:1b:10)			
Internet Protocol Version 4, Src: 192.168.169.53 (192.168.169.53), Dst: 202.141.80.32 (202.141.80.32)			
Transmission Control Protocol, Src Port: 63940 (63940), Dst Port: http (80), Seq: 1, Ack: 1, Len: 318			
Hypertext Transfer Protocol			
GET /~sukumar/cs349/HTTP-wireshark-file1.html HTTP/1.1\r\n			
Host: 202.141.80.32\r\n			

```

User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:5.0) Gecko/20100101
Firefox/5.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
=0.8\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
\r\n
[Full request URI: http://202.141.80.32/~sukumar/cs349/HTTP-wireshark-
file1.html]
[HTTP request 1/1]
[Response in frame: 11]

No.      Time                Source                Destination
Protocol Length Info
  11  10.495353000    202.141.80.32        192.168.169.53      HTTP
      542      HTTP/1.1 200 OK (text/html)

Frame 11: 542 bytes on wire (4336 bits), 542 bytes captured (4336 bits) on
interface 0
Ethernet II, Src: Universa_46:1b:10 (e0:2a:82:46:1b:10), Dst: HonHaiPr_0a:
b0:03 (94:39:e5:0a:b0:03)
Internet Protocol Version 4, Src: 202.141.80.32 (202.141.80.32), Dst:
192.168.169.53 (192.168.169.53)
Transmission Control Protocol, Src Port: http (80), Dst Port: 63940
(63940), Seq: 1, Ack: 319, Len: 488
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
Date: Sat, 29 Mar 2014 12:48:34 GMT\r\n
Server: Apache/2.2.15 (Red Hat)\r\n
Last-Modified: Fri, 14 Mar 2014 08:57:23 GMT\r\n
ETag: "68161c-da-4f48d4220577b"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 218\r\n
Connection: close\r\n
Content-Type: text/html; charset=UTF-8\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.002234000 seconds]
[Request in frame: 9]
Line-based text data: text/html

```

Listing 1: HTTP GET/RESPONSE interaction

Ack Segment no.	Ack Sequence no.	Acknowledged Data
1	493	493
2	1953	1460
3	3413	1460
4	4873	1460
5	6333	1460
6	7793	1460
7	7793	0
8	7793	0
9	7793	0
10	12173	4380
11	13633	1460