Don Bosco Institute of Technology
Department of Information Technology
Wireless Technology

BE-IT **SEM 7**

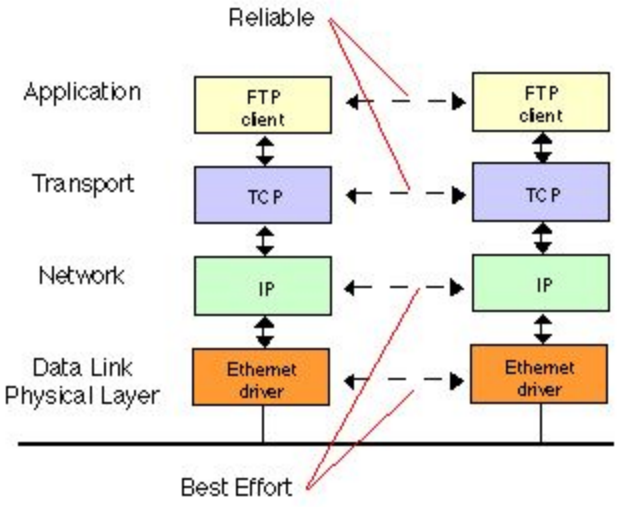
Name: Christine Polly **Roll no: 17**

Experiment No. 14

Title:
Simulation on determining the number of packets sent by TCP/UDP in a point to point network

Theory :
TCP-Transmission control protocol

Transmission Control Protocol (TCP) is a core protocol of the Internet protocol suite. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP.



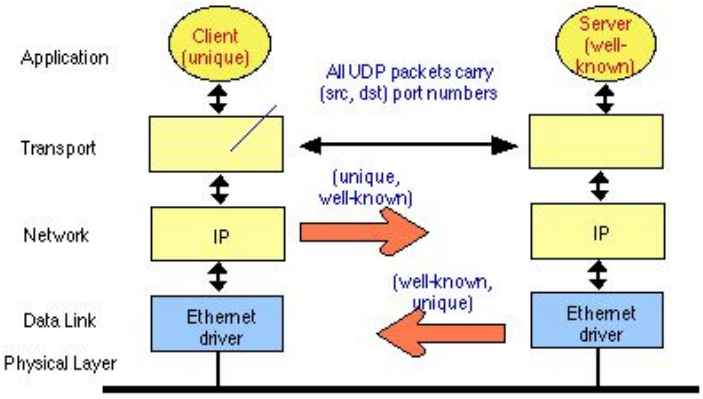
Transmission Control Protocol is one of the most used protocols in digital network communications and is part of the Internet protocol suite, commonly known as the TCP/IP suite. Primarily, TCP ensures end-to-end delivery of data between distinct nodes. TCP works in collaboration with Internet Protocol, which defines the logical location of the remote node, whereas TCP transports and ensures that the data is delivered to the correct destination.Before transmitting data, TCP creates a connection between the source and destination node and keeps it live until the communication is active. TCP breaks large data into smaller packets and also ensures that the data integrity is intact once it is reassembled at the destination node.

UDP-User datagram protocol

The User Datagram Protocol (UDP) is a transport layer protocol defined for use with the IP network layer protocol. It is defined by RFC 768 written by John Postel. It provides a best-effort datagram service to an end-system.
UDP does not provide any communications security. Applications that need to protect their communications against eavesdropping, tampering, or message forgery therefore need to separately provide security services using additional protocol mechanisms.UDP is widely used in video conferencing and real-time computer games.
The protocol permits individual packets to be dropped and UDP packets to be received in a different order than that in which they were sent, allowing for better performance. UDP network

traffic is organized in the form of datagrams, which comprise one message units. The first eight bytes of a datagram contain header information, while the remaining bytes contain message data. A UDP datagram header contains four fields of two bytes each:

- Source port number
- Destination port number
- Datagram size
- Checksum

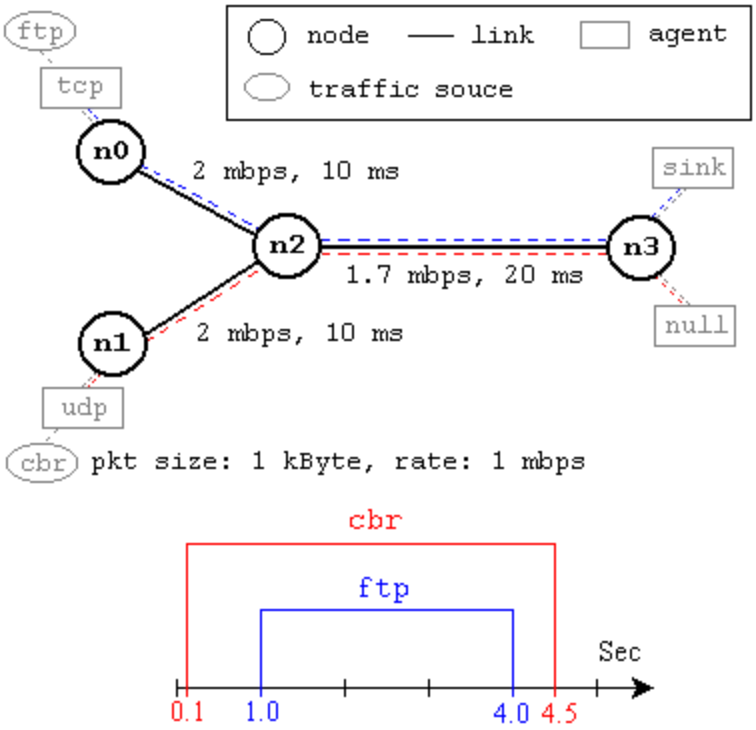


Duplex link

A duplex communication system is a point to point system composed of two connected parties or devices that can communicate with one another in both directions. "Duplex" comes from "duo" that means "double", and "plex" that means "structure" or "parts of".

Full duplex system, both parties can communicate with each other simultaneously. An example of a full-duplex device is a telephone; the parties at both ends of a call can speak and be heard by the other party simultaneously. The earphone reproduces the speech of the remote party as the microphone transmits the speech of the local party, because there is a two-way communication channel between them, or more strictly speaking, because there are two communication paths/channels between them.

In a half-duplex system, there are still two clearly defined paths/channels, and each party can communicate with the other but not simultaneously; the communication is one direction at a time. An example of a half-duplex device is a walkie-talkie two-way radio that has a "push-to-talk" button; when the local user wants to speak to the remote person they push this button, which turns on the transmitter but turns off the receiver, so they cannot hear the remote person.



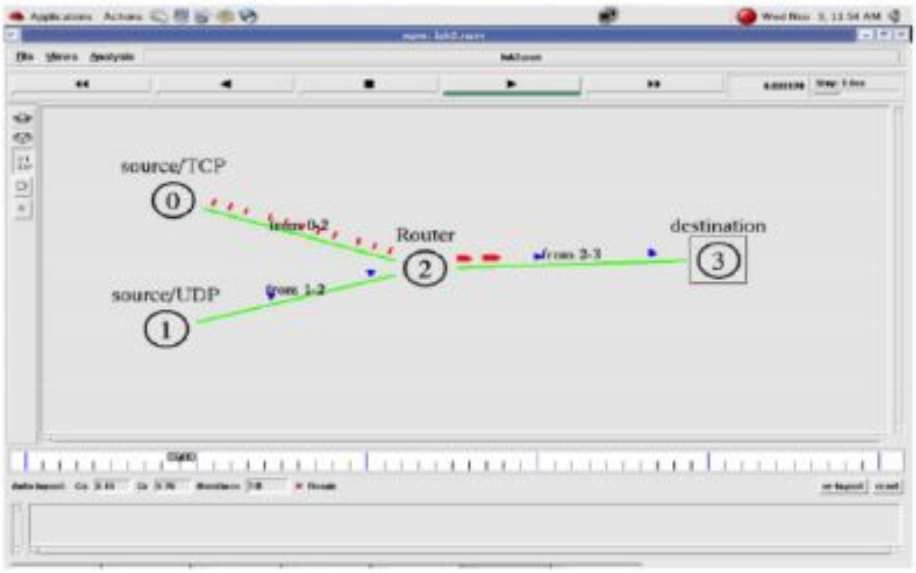
Introduction

This network consists of 4 nodes (n0, n1, n2, n3) as shown in above figure.

The duplex links between n0 and n2, and n1 and n2 have 2 Mbps of bandwidth and 10 ms of delay. The duplex link between n2 and n3 has 1.7 Mbps of bandwidth and 20 ms of delay.
Each node uses a DropTail queue, of which the maximum size is 10. A "tcp" agent is attached to n0, and a connection is established to a tcp "sink" agent attached to n3. As default, the maximum size of a packet that a "tcp" agent can generate is 1KByte. A tcp "sink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets.
A "udp" agent that is attached to n1 is connected to a "null" agent attached to n3. A "null" agent just frees the packets received. A "ftp" and a "cbr" traffic generator are attached to "tcp" and "udp" agents respectively, and the "cbr" is configured to generate 1 KByte packets at the rate of 1 Mbps. The "cbr" is set to start at 0.1 sec and stop at 4.5 sec, and "ftp" is set to start at 1.0 sec and stop at 4.0 sec.

CODE

```
set ns [new Simulator]
set tf [open lab2.tr w]
$ns trace-all $tf
set nf [open lab2.nam w]
$ns namtrace-all $nf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
```




```
# The below code is used to set the color and name's to the nodes.#
$ns color 1 "red"
$ns color 2 "blue"
$n0 label "source/TCP"
$n1 label "source/UDP"
$n2 label "Router"
$n3 label "destination"
$ns duplex-link $n0 $n2 100Mb 1ms DropTail
$ns duplex-link $n1 $n2 100Mb 1ms DropTail
$ns duplex-link $n2 $n3 100Mb 1ms DropTail

# The below code is used to set the color and labels to the links.#
$ns duplex-link-op $n0 $n2 color "green"
$ns duplex-link-op $n0 $n2 label "from 0-2"
$ns duplex-link-op $n1 $n2 color "green"
$ns duplex-link-op $n1 $n2 label "from 1-2"
$ns duplex-link-op $n2 $n3 color "green"
$ns duplex-link-op $n2 $n3 label "from 2-3"
```

```
# The below code is used create TCP and UDP agents and the traffic ftp & cbr respectively.#
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
set null3 [new Agent/Null]
$ns attach-agent $n3 $null3

#The below code is used to set the packet size of ftp and udp.#
$ftp0 set packetSize_ 500
$ftp0 set interval_ 0.001

#The below code is used to increase the data rate(if the #interval is more then the   more number of
packets goes to #destination).# $cbr1 set packetSize_ 500
$cbr1 set interval_ 0.001

#This code is used give a color red->tcp and blue ->udp.#
$tcp0 set class_ 1
$udp1 set class_ 2

# The below code is used connect the agents.
$ns connect $tcp0 $sink3
$ns connect $udp1 $null3
proc finish { } {
global ns nf tf $ns
flush-trace
exec nam lab2.nam &
close $nf
close $tf
exit 0
}
$ns at 0.1 "$cbr1 start"
$ns at 0.2 "$ftp0 start"
$ns at 5.0 "finish"
$ns run
```