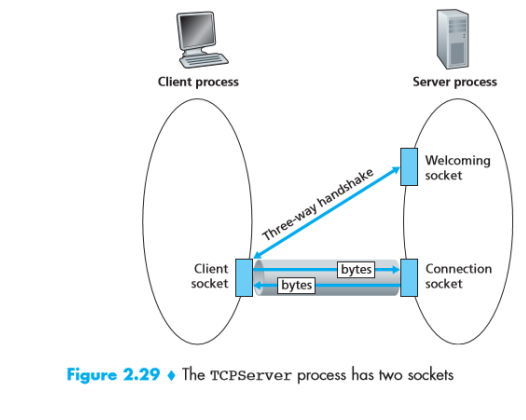
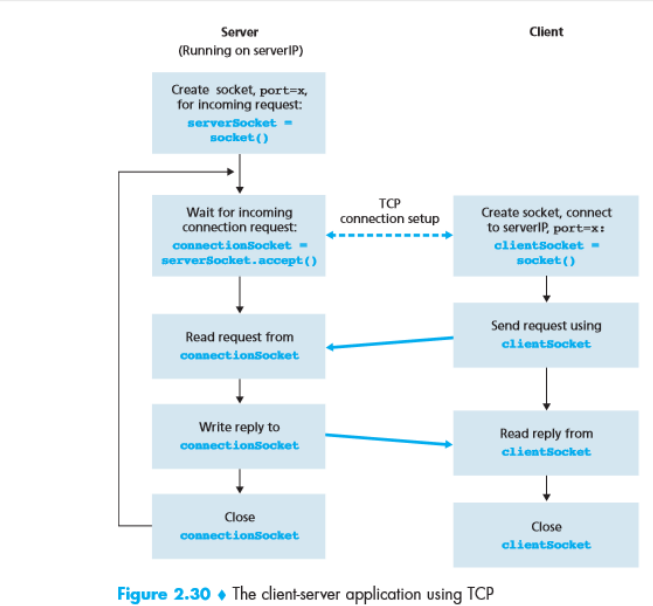
**AIM:** Implement TCP (Transmission Control Protocol) and UDP (User Datagram Protocol)on different pairs of systems using NS2 Simulation Tool.

**DESCRIPTION:**

TCP is a connection-oriented protocol. This means that before the client and server can start to send data to each other, they first need to handshake and establish a TCP connection. One end of the TCP connection is attached to the client socket and the other end is attached to a server socket. When creating the TCP connection, we associate with it the client socket address (IP address and port number) and the server socket address (IP address and port number). With the TCP connection established, when one side wants to send data to the other side, it just drops the data into the TCP connection via its socket.

With the server process running, the client process can initiate a TCP connection to the server. This is done in the client program by creating a TCP socket. When the client creates its TCP socket, it specifies the address of the welcoming socket in the server, namely, the IP address of the server host and the port number of the socket. After creating its socket, the client initiates a three-way handshake and establishes a TCP connection with the server.





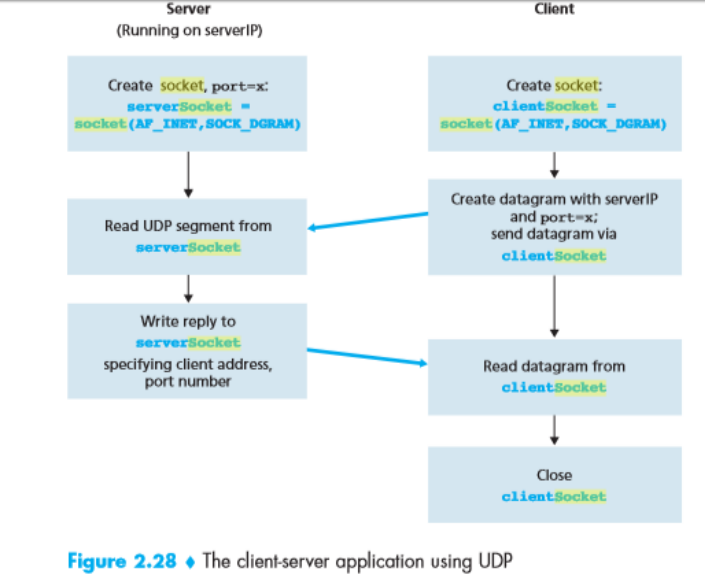
**UDP:-**

Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.

**DESCRIPTION:**

Using User Datagram Protocol, Applications can send data/message to the other hosts without prior communications or channel or path. This means even if the destination host is not available, application can send data. There is no guarantee that the data is received in the other side. Hence it's not a reliable service.

UDP is appropriate in places where delivery of data doesn't matters during data transition.



**SOURCE CODE:**

**TCP for various networks.**

**1 Simple client-server model using TCP**

set ns [new Simulator]

set nf [open cs.nam w]

$ns namtrace-all $nf

set nd [open cs.tr w]

$ns trace-all $nd

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

$ns duplex-link $n0 $n1 10Mbps 10ms DropTail

$ns duplex-link-op $n0 $n1 orient right

$ns duplex-link $n1 $n2 10Mbps 10ms DropTail

$ns duplex-link-op $n1 $n2 orient right

$ns duplex-link $n2 $n3 10Mbps 10ms DropTail

$ns duplex-link-op $n2 $n3 orient right

$n0 shape box

$n1 shape hexagon

$ns at 0.1 "$n0 label client"

$ns at 0.2 "$n1 label server"

$ns at 0.1 "$n0 color blue"

$ns at 0.2 "$n1 color red"

set tcp0 [new Agent/TCP]

$ns attach-agent $n0 $tcp0

set Sink [new Agent/Null]

$ns attach-agent $n3 $Sink

$ns connect $tcp0 $Sink

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize \_ 500

$cbr0 set interval \_ 0.005

$cbr0 attach-agent $tcp0

$ns at 0.5 "$cbr0 start"

$ns at 2.5 "$cbr0 stop"

proc finish { } {

global ns nf nd

$ns flush-trace

close $nf

exec nam cs.nam &

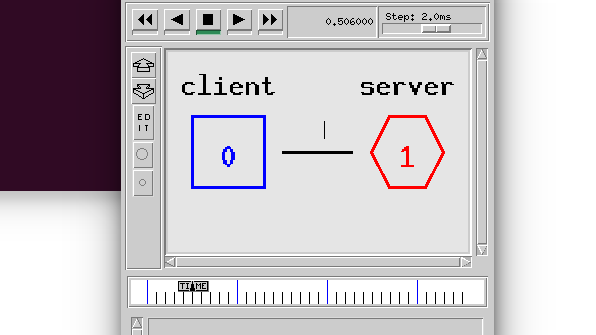
exit 0

}

$ns at 5.0 "finish"

$ns run

**OUTPUT:**



**TCP: Data transfer among set of nodes:**

set ns [new Simulator]

set nf [open cse.nam w]

$ns namtrace-all $nf

set nd [open cse.tr w]

$ns trace-all $nd

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

$ns duplex-link $n0 $n1 10Mbps 10ms DropTail

$ns duplex-link-op $n0 $n1 orient right

$ns duplex-link $n1 $n2 10Mbps 10ms DropTail

$ns duplex-link-op $n1 $n2 orient right

$ns duplex-link $n2 $n3 10Mbps 10ms DropTail

$ns duplex-link-op $n2 $n3 orient right

set udp0 [new Agent/TCP]

$ns attach-agent $n0 $udp0

set Sink [new Agent/Null]

$ns attach-agent $n3 $Sink

$ns connect $udp0 $Sink

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize \_ 1500

$cbr0 set interval \_ 0.005

$cbr0 attach-agent $udp0

$ns at 0.1 "$cbr0 start"

$ns at 0.4 "$cbr0 stop"

proc finish { } {

global ns nf nd

$ns flush-trace

close $nf

exec nam cse.nam &

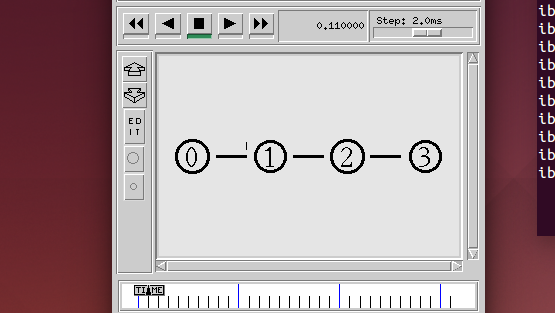
exit 0

}

$ns at 5.0 "finish"

$ns run

**OUTPUT:**



**#UDP: Client-Server**

set ns [new Simulator]

set nf [open cs.nam w]

$ns namtrace-all $nf

set nd [open cs.tr w]

$ns trace-all $nd

set n0 [$ns node]

set n1 [$ns node]

$ns duplex-link $n0 $n1 10Mbps 10ms DropTail

$ns duplex-link-op $n0 $n1 orient right

$n0 shape box

$n1 shape hexagon

$ns at 0.1 "$n0 label client"

$ns at 0.2 "$n1 label server"

$ns at 0.1 "$n0 color blue"

$ns at 0.2 "$n1 color red"

set udp0 [new Agent/UDP]

$ns attach-agent $n0 $udp0

set udp1 [new Agent/Null]

$ns attach-agent $n1 $udp1

$ns connect $udp0 $udp1

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize \_ 500

$cbr0 set interval \_ 0.005

$cbr0 attach-agent $udp0

$ns at 0.5 "$cbr0 start"

$ns at 2.5 "$cbr0 stop"

proc finish { } {

global ns nf nd

$ns flush-trace

close $nf

exec nam cs.nam &

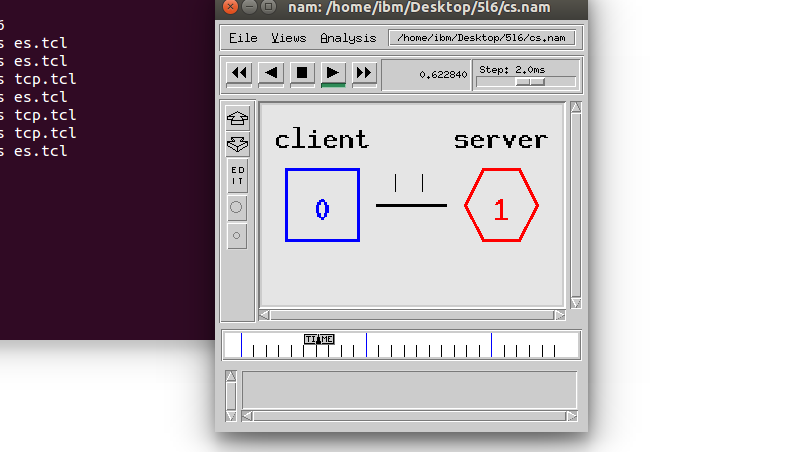
exit 0

}

$ns at 5.0 "finish"

$ns run

**OUTPUT**:



**#UDP: Data Transfer between nodes**

set ns [new Simulator]

set nf [open cse.nam w]

$ns namtrace-all $nf

set nd [open cse.tr w]

$ns trace-all $nd

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

$ns duplex-link $n0 $n1 10Mbps 10ms DropTail

$ns duplex-link-op $n0 $n1 orient right

$ns duplex-link $n1 $n2 10Mbps 10ms DropTail

$ns duplex-link-op $n1 $n2 orient right

$ns duplex-link $n2 $n3 10Mbps 10ms DropTail

$ns duplex-link-op $n2 $n3 orient right

set udp0 [new Agent/UDP]

$ns attach-agent $n0 $udp0

set Sink [new Agent/Null]

$ns attach-agent $n3 $Sink

$ns connect $udp0 $Sink

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize \_ 500

$cbr0 set interval \_ 0.005

$cbr0 attach-agent $udp0

$ns at 0.1 "$cbr0 start"

$ns at 0.4 "$cbr0 stop"

proc finish { } {

global ns nf nd

$ns flush-trace

close $nf

exec nam cse.nam &

exit 0

}

$ns at 5.0 "finish"

$ns run

**OUTPUT:**

