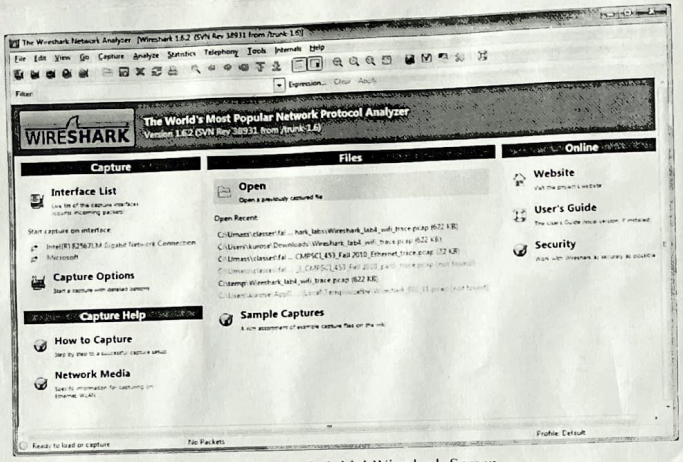


Figure 2: Initial Wireshark Screen

05/01/22 24

Experiment No: 10

FAMILIARIZATION OF WIRESHARK

AIM

a. Using wireshark capture packets transferred while browsing a selected website. Investigate the protocols used in each packet, the values of the header fields and the size of the packets.

b. Using wireshark, observe three way handshaking connection establishment, three way handshaking connection termination and data transfer in client server communication using TCP.

c. Explore atleast the following features of wireshark. Filters, Flow Graphs (TCP), Statistics and protocol hierarchies.

THEORY

The basic tool for observing the messages exchanged between executing Protocol entities is called a packet sniffer. It consists of two parts - packets capture library (receives copy of every link layer frame that is sent from or received by your computer) and packet analyzer displays contents of all fields within a protocol message).

Wireshark is a free network protocol that runs on windows, Linux/Unix and Mac computers, its an ideal packet analyzer for our labs. It is ~~style~~, has a large user base and well documented support

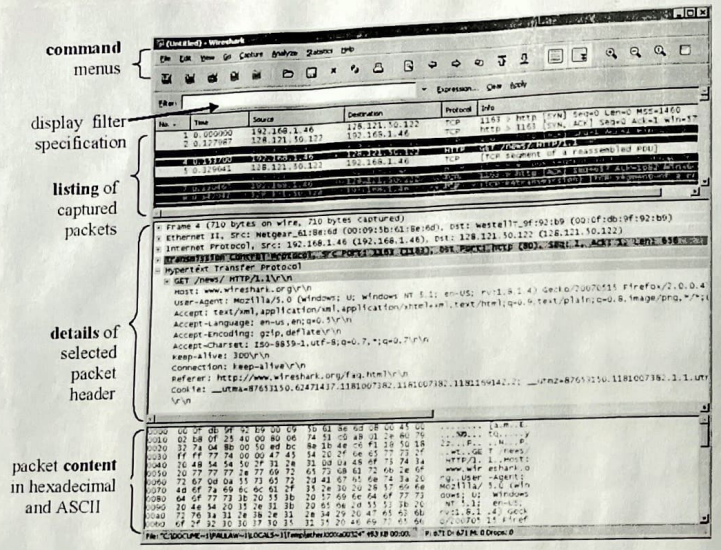


Figure 3: Wireshark Graphical User Interface. during packet capture and analysis

that includes user guide, man pages and detailed FAQ, rich functionality that includes the capability to analyze hundreds of protocols and a well designed user interface

when you run wireshark, you'll get a startup screen, as shown in the figure.

The wireshark interface has five components:

- the command menus are standard pulldown menus located at the top of the window. The file menus allows you to save captured packet data or open a file containing previously captured packet data, and exit the wireshark application. the capture menu allows you to begin packet capture
- the packet-listing window displays a one line summary for each packet captured, including the packet number (assigned by wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type and protocol specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest level protocol that sent or received this packet, ie the protocol that is the source or ultimate sink for this packet.
- The packet header details window provides details about the packet

selected (highlighted) in the packet-listing window. these details include information about the Ethernet frame (assumption - packet was sent/recieved over an Ethernet interface) and IP datagram that contains this packet. the amount of Ethernet and IP layer detail displayed can be expanded or minimized by clicking on the plus minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window.

- The packet contents window displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
- Towards the top of the wireshark graphical user interface, is the packet display filter field, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window.

Prepare wireshark to capture packets

Step 1 : Start wireshark

2 : Select an interface to use for capturing packets.

a : From the capture menu, choose interfaces.

b. choose the local network Ethernet interface adapter for capturing network traffic

3 : Start a network capture

a. click the start button of the chosen interface

b. write down the IP address associated with the selected

Ethernet adapter because that is the source IP to look for when examining captured packets.

Generate and Analyze Captured Packets

1. Open a browser and open a website
2. Goto www.google.com. Minimize the google window, and return to wireshark. You should see captured traffic similar to that shown below
3. Captured windows are now active. Locate the source, Destination, and protocol columns on the wireshark display screen. The HTTP data that carries webpage text and graphics uses TCP for reliability
4. Stop the capture by choosing stop from capture menu
5. Analyze the captured output.
   a. In the screen capture, the packet that starts with frame 1, which is an ARP broadcast from the source computer to determine the MAC address of the router default gateway. The computer needs to resolve the default gateway IP address to the interface MAC address. before it can send the first frame or packet to the router.
   b. The second frame is reply from router telling the computer that MAC address of its Fast Ethernet Interface
   c. the third frame is a DNS query from computer to the configured DNS server attempting to resolve the domain name www.google.com to the IP address of the webserver. The computer must have the IP

address before it can send the first frame to the web server.

d. The fourth frame is response from DNS server with the IP address of www.google.com

e. The fifth frame is the start of the TCP three way handshake (SYN)

4. Filter the capture to view only TCP packets.

a. From Analyze menu option, click Display Filters.

b. In the display filter window, click TCP only and then click OK

c. In wireshark window, scroll to first captured TCP packet. This should be the first packet in the flow

d. In the No column, look for 3 packets similar to the first three as shown in image - [SYN], [SYN, ACK], [ACK]

5. Inspect the TCP initialization sequence

a. In the top wireshark window, click on the line containing the first packet identified in step 4.

b. Click the + icon to expand view of TCP information, to contract the view, click the - icon.

c. Notice on first TCP packet that the relative sequence number is set to 0, and the SYN bit is set to 1 in the flags field.

d. Notice in the second TCP packet of handshake that the relative sequence number is set 0, and the SYN bit and the ACK bit are set to 1 in the flags field.

e In the 3rd and final frame of handshake, only ACK bit is set. and the sequence number is set to the starting point of 1. The acknowledgement number is also set to 1 as a starting point. the TCP connection is now established, and communication between the source computer and the web server can begin

f close wireshark

6. observe the data transfer packets in client server communication using TCP.

7. Inspect the TCP connection termination sequence which includes FIN from the client, Ack from server, FIN from server and Ack from client.

RESULT

Successfully ~~famiwarised~~ the basics of wireshark network protocol analyzer

12/07/22

Experiment NO:11

FAMILIARIZATION AND INTRODUCTION TO NS-2

AIM

To familiarise Network simulator-2 (NS2)

THEORY

NS2 is a discrete event driven and object-oriented network simulator, developed by UC Berkeley. It is useful in studying the dynamic nature of communication networks which is used in networking researches. simulation is the process of designing the model of a real system and conducting experiments with this model for the purpose of understanding the behavior of the system and/or evaluating various strategies for the operation of the system A simulator maintains discrete events and execute one event after the other. As there is only single thread of control there won't be any problem of race condition or locking

NS2 contains modules for numerous components such as routing transport layer protocol, application etc. to investigate network performance researches can use an easy to use scripting language to configure a network observe results generated by NS2 NS2, which works at packet level, has become the most widely used open source network simulator simulation of wired as well as wireless network functions and protocols (eg. routing algorithms, TCP, UDP) can be done using NS2 It is primarily

UNIX based. It is a standard experiment environment in research community.

The basic architecture of NS2 is given in fogure. NS uses two languages - C++ (fast to run, slower to change) suitable for detailed protocol implementation and OTCl (object oriented tool command language) which runs much slower but can be changed very quickly making it ideal for simulation configuration. NS (via TclCL) provides glue to make objects and variables appear on both languages. The user gives OTCl script, which makes use of the object compiled in C++ through an OTCl linkage (TclCl) that creates a matching of OTCl object for each of the C++.

The backend for NS2 is C++ event scheduler which consits of many protocols and defines internal mechanism of simulation objects. Frontend is OTCl which will setup simulation by assembling and configuring the objects and schedule discrete events. Mapped to a C++ object, variables in the OTCl domains are sometimes referred to handles. Conceptually, a handle is just a string in OTCl domain and does not contain any functionality. Instead of functionality (eg. receiving a packet) is defined in the mapped C++ object. The member procedures and variables in the OTCl domain are called instance procedures and instance variables respectively

NS2 provides users with an executable command ns which takes an input argument, the name of a Tcl simulation scripting file. Users are feeding the name of a Tcl simulation script (which setup a simulation) as an input argument of an NS2 executable command ns. In most cases

a simulation trace files created and is used to plot graph and or to create animation.

After simulation, NS2 outputs either text-based or animation based simulation result. To interpret these results graphically and interactively, tools such as NAM (Network Animator) and x Graph are used. To analyze a particular behavior of the network, users can extract a relevant subset of text based data and transform it to a more conceivable presentation.

Protocols implementation in NS2 - Transport layer (Traffic Agent) - TCP, UDP. Network layer (Routing Agent); Interface queue - FIFO queue, Drop Tail queue, priority queue; Logic link control layer - LEEE 802.2, ARP

Accompanying tools

- NAM - tool that allows visualizing motion of packets through nodes and links of networks
- xgraph - tool that allows plotting the results of the simulation in the form of curves.

NS programming structure

- Every NS2 script starts with creating simulator object set ns [new simulator]
- Create the event scheduler
- Turn on tracing
- Create network topology
- Create transport connections
- Generate traffic
- Insert errors

Creating event scheduler

Set ns [new simulator]

schedule an event : $ns at <time> <event>

event is any legitimate ns/tcl function

eg: $ns at 5.0 "finish"

```
proc finish {} {
    global ns nf
    close $nf
    exec nam out.nam &
    exit 0
}
```

start scheduler

$ns run

Tracing

All packet trace

$ns trace_all [open out.tr w]

Variable trace

Set par [open output/param.tr w]

$tcp attach $par

$tcp trace cwnd_

$tcp trace rcvseq_

$tcp trace rtt_

Tracing and Animation

Network Animator

Set nf [open out.mean w]

$ns namtrace_all $nf

```
proc finish {} {
    global ns nf
    close $nf
    exec nam out.nam &
    exit 0
}
```

Creating Topology

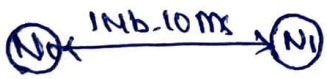
Creating nodes

```
set n0 [$ns node]
set n1 [$ns node]
```

Creating link between nodes

$ns <link type> $n0 $n1 <bandwidth> <delay> <queue-type>$

ns duplex-link $n0 $n1 1Mb 10ms DropTail



Send data

Create UDP portal

```
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
```

Create CBR traffic source for feeding into UDP

```
Agent set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetsize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
```

Create traffic sink

```
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
```

Connect two agents

```
$ns connect $udp0 $null0
```

Start and stop of data

```
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

TCP/IP model

Create TCP agent and attach it to the node

```
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
```

Create a Null Agent & Attach it to the node

```
set null0 [new Agent/TCPSink]
$ns attach-agent $n1 $null0)
```

Connect the agents

```
$ns connect $tcp0 $null0
```

Traffic on top of TCP

FTP

```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
```

Telnet

Set telnet [New Application / Telnet]

$ telnet attach

RESULT

Successfully familiarised the basics of Network simulator 2

EXPERIMENT NO : 12

AUTHOR NAME : NIDHA KAREEM

DATE : 25/07/2022

PROGRAM:

```
set ns [new Simulator] set nr [open tr1.tr w]
$ns trace-all $nr set nf [open tr1.nam w]
$ns namtrace-all $nf
set n0 [$ns node] set n1 [$ns node] set n2 [$ns node] set n3 [$ns node] set n4 [$ns node] set n5 [$ns node] set n6 [$ns node]  set n7 [$ns node]
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n6 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 3Mb 10ms DropTail  $ns
duplex-link $n3 $n7 2Mb 10ms DropTail
$ns duplex-link $n3 $n4 2Mb 10ms DropTail  $ns
duplex-link $n3 $n5 2Mb 10ms DropTail
$ns duplex-link-op $n0 $n2 orient rightdown $ns duplex-link-op $n6 $n2 orient right
$ns duplex-link-op $n1 $n2 orient rightup $ns duplex-link-op $n2 $n3 orient right $ns duplex-link-op $n3 $n5 orient rightup $ns duplex-link-op $n3 $n7 orient right

$ns duplex-link-op $n3 $n4 orient right-down
set tcp0 [new Agent/TCP]  set tcpsink5 [new
Agent/TCPSink]
$ns attach-agent $n0 $tcp0 $ns attach-agent $n5 $tcpsink5 $ns connect $tcp0 $tcpsink5 $tcp0  set
fid_ 2
$ns color 2 Black
```

12/07/22

37

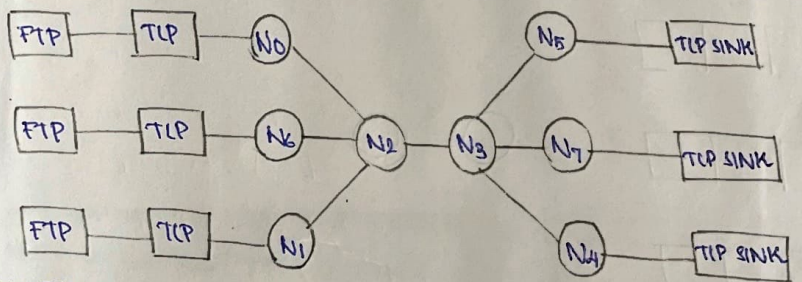
Experiment No: 12

SIMULATION OF TOPOLOGY

AIM

Simulate the given topology

NETWORK TOPOLOGY

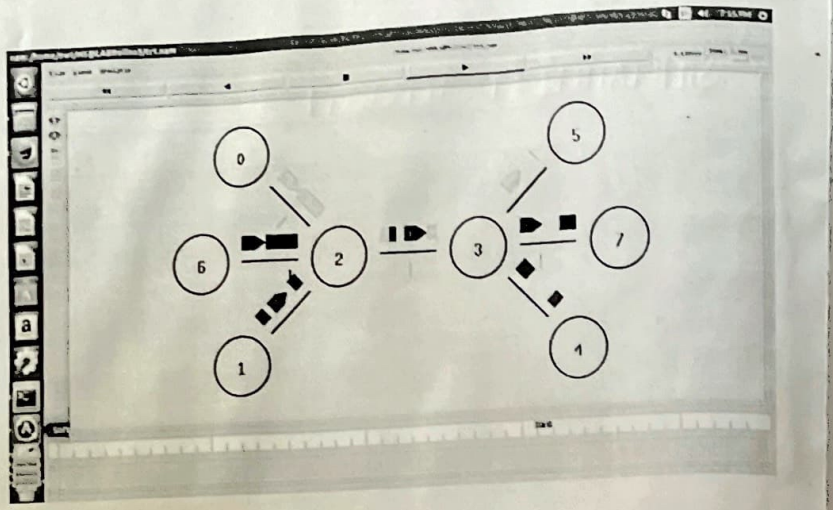


FEATURES

CBR starts from node 1 at 0.1 ms
CBR stop at node 4 at 4.5 ms
FTP0 starts from node 0 at 1.0 ms
FTP0 stop at node 5 at 4.0 ms
FTP6 starts from node 6 at 1.5 ms
FTP6 stop at node 7 at 4.2 ms
Simulation stops at 5.0 ms

ALGORITHM

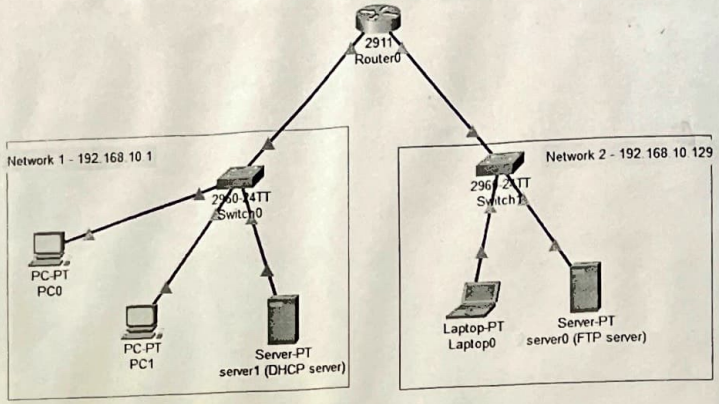
Step 1 : Start
Step 2 : Create simulator objects ns for designing the given simulation.



step 3 : Open trace file and nam in write mode
step 4 : Create node objects
step 5 : Create links to the appropriate nodes using $ns duplex-link command
step 6 : Set orientation of nodes
step 7 : Create two TCP agents and attach it to corresponding nodes.
step 8 : Create two FTP applications and attach it to corresponding nodes
step 9 : Create two TCP sinks agents and attach it to corresponding nodes
step 10 : Connect TCP agent to TCP sink agent.
step 11 : Create UDP agent and attach with corresponding node
step 12 : Create CBR traffic src and attach with UDP agent.
step 13 : Create NULL agent and attach it to corresponding nodes
step 14 : Connect UDP agent to NULL agent
step 15 : Set the start time and stop time of CBR as 0.1 and 4.5 respectively
step 16 : Set the start time and stop time of FTP0 as 1.0 and 4.0 respectively
step 17 : Set the start time and stop time of FTP0 as 1.3 and 4.2 respectively
step 18 : Set finish time as 5.0
step 19 : stop

RESULT

The program has been successfully executed, the network is implemented and the output is obtained and verified.



Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.10.3

Pinging 192.168.10.3 with 32 bytes of data:

Reply from 192.168.10.3: bytes=32 time<1ms TTL=128
Reply from 192.168.10.3: bytes=32 time<1ms TTL=128
Reply from 192.168.10.3: bytes=32 time<1ms TTL=128
Reply from 192.168.10.3: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ftp 192.168.10.130
Trying to connect...192.168.10.130
Connected to 192.168.10.130
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

26/07/22

Experiment No: 13

DESIGN AND CONFIGURE A NETWORK

AIM

To design a network with multiple subnets using required network devices and configure DHCP server, file server in the network.

THEORY

Cisco packet tracer is a tool that provides a network simulation to practice simple and complex networks. The main purpose of cisco packet tracer is to help students learn the principles of networking with hands-on experience as well as develop cisco technology specific skills

Workspace

Logical

Logical workspace shows logical network topology of the network the user has built. It represents the placing, connecting and clustering virtual network devices.

Physical

Physical workspace allows the graphical physical dimension of the logical network. It depicts the size and placement in how network devices routers, switches and hosts should look in a real environment

Configuration steps

Step 1 : Configuring Router 0

a. Select a 2911 router from network devices and drag and drop to the network

b. Select Router 0 and goto Config

c. Configure the Gigabit Ethernet 0/0 by assigning IP address as 192.168.10.1 and subnet mask as 255.255.255.128 and turn on the port status.

d. Configure the Gigabit Ethernet 0/1 by assigning IP address as 192.168.10.129 and subnet mask as 255.255.255.128 and turn on port status.

Step 2: Configure Server 0

To configure FTP server

a. Select a server from End devices and drag and drop to the workspace

b. Go to the global settings in config and assign default gateway 192.168.10.129.

c. Go to Fast Ethernet 0 and assign IP address and subnet mask as 192.168.10.130.

d. Go to services and open FTP service

e. Go to user setup and create a username and password

f. Save changes

Step 3: Configure Server 1

To configure DHCP server

a. Select server from end devices and drag and drop to the workspace

b. Go to the global settings ~~and~~ in config and assign default gateway as 192.168.10.1

c. Go to Fast Ethernet 0 and assign IP address and subnet mask as 192.168.10.2, 255.255.255.128

d. Go to services and open DHCP service
e. Turn DHCP service on
f. change the pool name if required
g. Enter the default gateway
h. Save the changes

Step A: configuring switch, PCs, Laptop and making connections

a. Select 2 2960-24 switches switch 0 and switch 1 from the network devices and drag and drop to the workspace
b. select 2 PC-PT type PC and one laptop PT type Laptop from End devices and drag and drop to the workspace.
c. Connect FastEthernet 0 port of PC0 to the FastEthernet 0/1 port of switch 0 using copper straight-through code.
d. Connect for FastEthernet 0 port of PC1 to the FastEthernet 0/2 port of switch 0 using copper straight-through cable
e. Connect FastEthernet 0 port of server 1 (DHCP server) to the FastEthernet 0/3 port of switch 0 using copper straight through cable
f. Connect FastEthernet 0/4 port of switch 0 to the Gigabit Ethernet 0/0 of Router 0 using copper straight through cable
h. Connect FastEthernet 0 port of server 0 (FTP server) to the FastEthernet 0/2 port of switch 1 using copper straight-through cable
i. Connect FastEthernet 0/3 port of switch 1 to the Gigabit Ethernet 0/1 of router 0 using copper-straight-Through cable
j. Select PC0 goto FastEthernet 0 in config and choose DHCP option

to assign IP address for this PC

a. Select PC1, goto FastEthernet 0 in config and choose DHCP option to assign IP address for this PC

b. Select laptop, goto FastEthernet 0 in config and choose DHCP option to assign IP address for this PC. Then goto IP configuration in Desktop section and configure IP address as 192.168.10.131, subnet mask as 255.255.255.128 and default gateway as 192.168.10.129

Step 5: Accessing FTP server from laptop

a. Select laptop, goto command prompt in Desktop section

b. Inside command prompt, type the command "FTP 192.168.10.130"

c. Enter the correct username and password to access FTP server

RESULT

The program has been successfully executed, the network with multiple subnets is designed and output is obtained and verified