

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

# COMPUTER NETWORKS

*Submitted by*

**JYOTIRADITYA SINGH BAGHEL**

**1BM20CS064**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**

(Autonomous Institution under VTU)

**BENGALURU-560019**

**October-2022 to Feb-2023**

**B. M. S. College of Engineering,  
Bull Temple Road, Bangalore 560019**  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “Lab course” carried out by **Jyotiraditya singh baghel (1BM20CS064)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks** workprescribed for the said degree.

Nandhini Vineeth  
Designation  
Department of CSE  
BMSCE, Bengaluru

**Dr. Jyothi S Nayak**  
Professor and Head  
Department of CSE  
BMSCE, Bengaluru

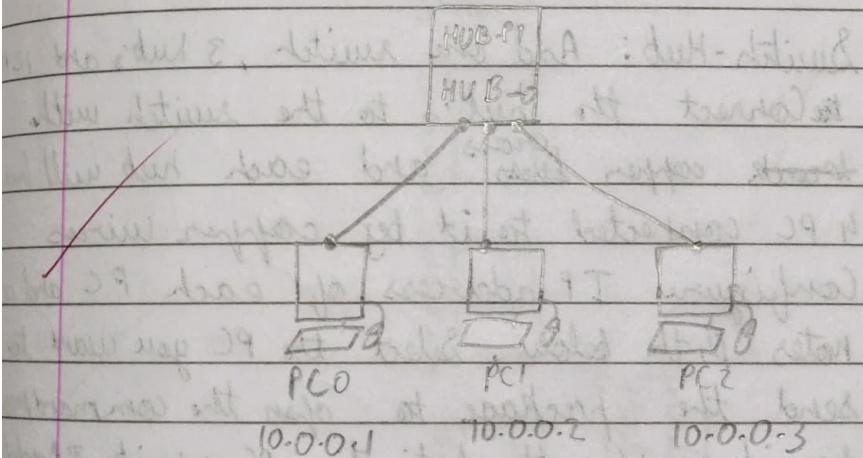
# Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>
<b>1</b>	<b>10-11-22</b>	Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.
<b>2</b>	<b>7-11-22</b>	Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply
<b>3</b>	<b>24-11-22</b>	Configuring default route to the Route
<b>4</b>	<b>1-12-22</b>	Configuring DHCP within a LAN in a packet Tracer
<b>5</b>	<b>8-12-22</b>	Configuring RIP Routing Protocol in Routers
<b>6</b>	<b>15-12-22</b>	Demonstration of WEB server and DNS using Packet Tracer
<b>7</b>	<b>29-12-22</b>	Write a program for error detecting code using CRC-CCITT (16-bits)
<b>8</b>	<b>12-1-23</b>	Write a program for distance vector algorithm to find suitable path for transmission
<b>9</b>	<b>12-1-23</b>	Implement Dijkstra's algorithm to compute the shortest path for a given topology.
<b>10</b>	<b>5-1-23</b>	Write a program for congestion control using Leaky bucket algorithm.
<b>11</b>	<b>28-1-23</b>	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.
<b>12</b>	<b>28-1-23</b>	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

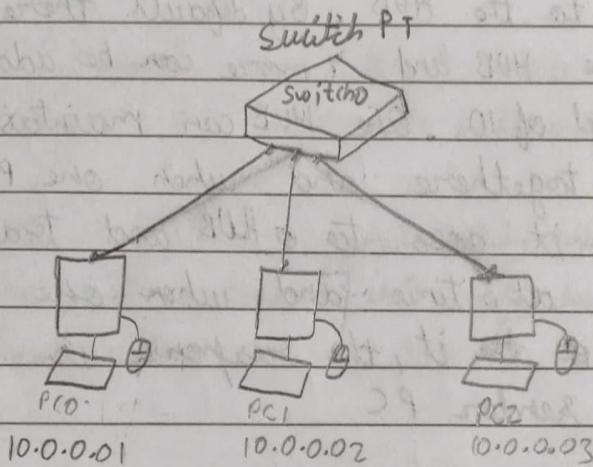
AIM: Creating a Topology and simulate sending a single PDU from source to destination using hub and switch as connecting devices

Topology:

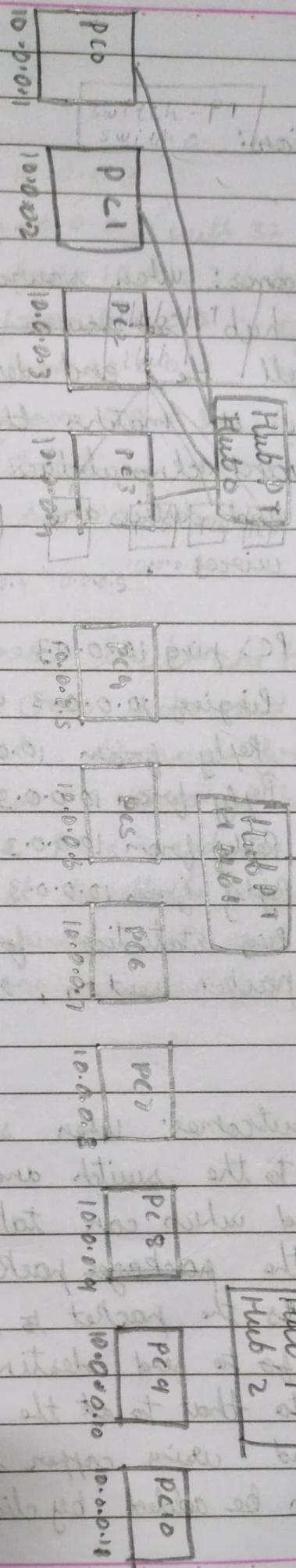
HUB: All the PC's are connected with a copper wire to the HUB. By default there are 6 ports in the HUB and 4 more can be added making a total of 10. One HUB can maintain upto 10 PC's together and when one PC sends a PDU it goes to the HUB and travel to other PC's at a time and when one PC acknowledges it, the response is sent back to the sender PC.



Switch : It can also connect with multiple PC's and they are also connected with copper wires. Then connection here is between ethernet and fast ethernet. The PDU is sent from one PC to another. It acknowledges it and then send it back. Then the switch sends PDU to each of the system.



Switch-Hub: Add one switch, 3 hubs and 12 PCs.  
Connect the hub's to the switch with ~~copper~~ <sup>cross</sup> wires and each hub will have 4 PC connected to it by copper wires.  
Configure IP addresses of each PC and add notes in the below. Select the PC you want to send the package to open the command prompt for it specify the destination PC and its IP address and Add a simple PDU by selecting the 2 PCs you want to send.



## Observation:

### Hub:

**Scanning outcomes:** when source sends a packet in network hub ~~sends~~ source the packet it sends it to all the end devices in the network and where it matches the specified address it accepts and acknowledges and rest ignore it. Hub and end devices are connected using ~~copper~~ straight wires.

**Result:** PC > ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Ping statistics for 10.0.0.3

packet sent = 4, received = 4, lost = 0

### Switches

**Scanning outcomes:** when source device sends a message to the switch and a connection is established which can take some time then it receives the ~~packaged~~ packet. It initially ~~sends~~ broadcasts the packet to all the connected devices to find destination. Then the message is sent to that to ~~at~~ the device. Connection is established using copper straight wires. No. of ports can be added by clicking the devices if needed.

Result:

PC > ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Ping statistics for 10.0.0.3

Packet sent = 4 received = 4, lost = 0

### Switch-Hub :

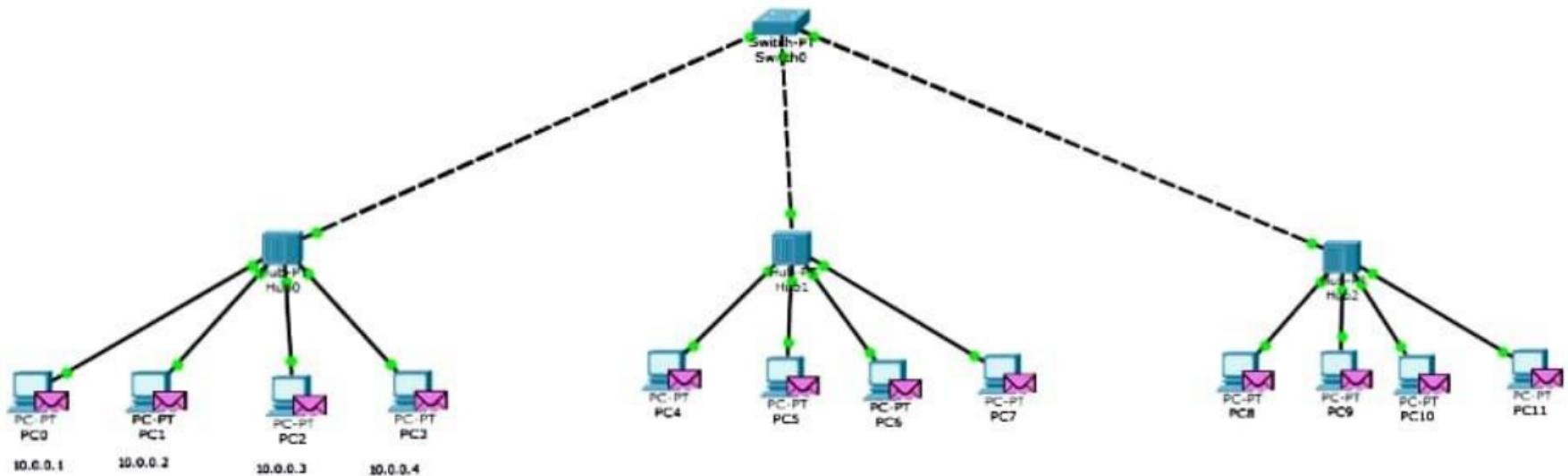
Learning outcomes: Switch and Hub are connected through copper cross wire but PC is connected with copper straight. Message from ~~from~~ source PC to destination PC through ~~hub~~ hub and it is sent to all the PC's and the switch. Then switch sends it to all the connected devices. The dist PC acknowledges it and send it back.

No. of ports can be added by clicking the ~~to~~ switch.

✓  
17/1/22



**SWITCH-HUB**  
IBM20CS064



Net View Background

viewport

## Simulation Panel

## Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
█	4.105	Hub0	PC3	STP	
█	4.105	Hub1	PC4	STP	
█	4.105	Hub1	PC5	STP	
█	4.105	Hub1	PC6	STP	
█	4.105	Hub1	PC7	STP	
█	4.105	Hub2	PC8	STP	
█	4.105	Hub2	PC9	STP	
█	4.105	Hub2	PC10	STP	
█	4.105	Hub2	PC11	STP	

Reset Simulation  Constant DelayCaptured to:  
4.105 s

## Play Controls

Back Auto Capture / Play Capture / Forward

## Event List Filters - Visible Events

ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISMMP, LACP, NDR, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RDP, RDPng, RTP, SCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

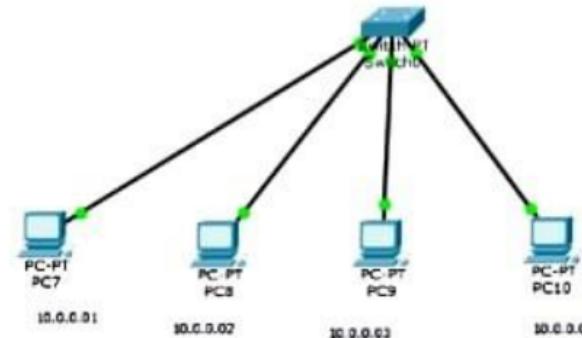
Edit Filters

Show All/None



10.0.0.2

SWITCH REALTIME  
IBM20CS064



PC7

Physical Config Desktop Custom Interface

Command Prompt

```
Packet Tracer PC Command Line 1.0
PCping 10.0.0.01
Invalid Command.

PCping 10.0.0.01

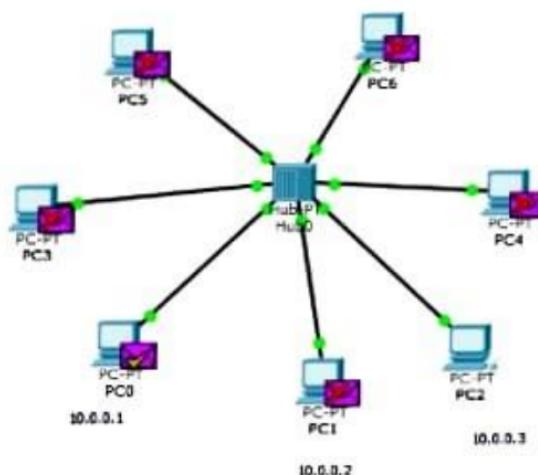
$pinging 10.0.0.01 with 32 bytes of data

Reply from 10.0.0.01: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.01
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).
    Approximate round trip times in milli-seconds.
        Minimum = 1ms, Maximum = 4ms, Average = 1ms

PC-
```

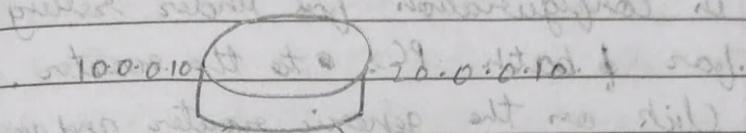
HUB  
1BM20CS064



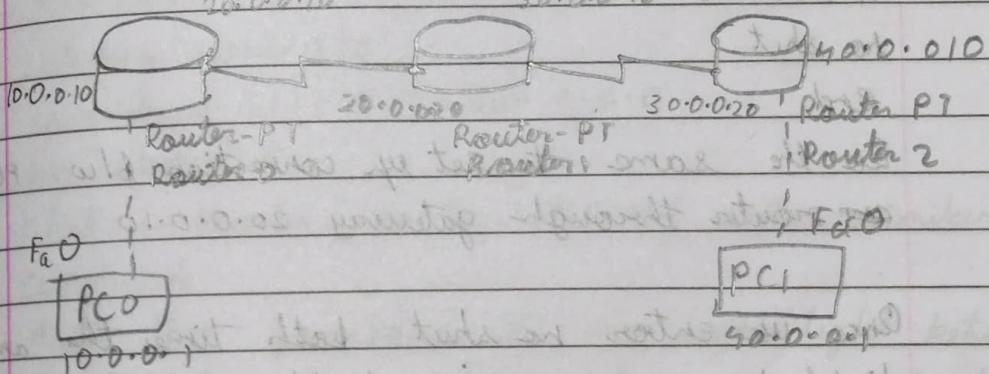
**AIM:** Configuring IP addresses to routers in packet tracer to explore the following messages Ping responses: destination unreachable: Request timed out: reply from software interrupt call? 0

**Topology** at certain time there will be 0, 2, 3, 4 hosts either

**Using single router:** two PC's connected



**Using 3 router 20 PC's connected to network**



Procedure

- Using single router, 2 PC's
- ① Use generic router & add generic PC's in the workspace
- ② Use copper cross over wires to connect the router and PC's.
- ③ Configure IP address for each PC and in configuration box under setting set gateway for both PCs to the router.
- ④ Click on the generic router and go to the CLI. Enter commands to set up a connection between PC1 and generic router through gateway 10.0.0.10

No

#enable

#config

```
interface fastethernet 0/0
ip address 10.0.0.10 255.0.0.0
```

no shut

end

Do the same to set up connection b/w PC2 and router through gateway 20.0.0.10

Once we enter no shut both times the amber light turns green immediately indicating that the 2 devices are ready for use.

- Simulation mode : Add a simple PDU by selecting the PC's and click on auto capture
- Real time mode : Select the PC you want to send the packet from which is PC0 and open command prompt from desktop tab. Specify the destination tab address. A response is sent from destination PC to source PC.

- Using 3 routers 2 PC's
- ① Place 3 gen routers and 2 gen PCs in the workspace
- ② Place notes for all devices and mention their IP address.
- ③ Use copper cross wires to connect PC and router.
- ④ Use serial DCE to connect routers.
- ⑤ Set IP address and subnet mask for PCs.
- ⑥ Set the gateway as the IP address of the next router.
- ⑦ IP address of PC and its gateway address should belong to the same network.

For connecting 2 routers

Click on router 0 go to CLI and enter the commands

No

enable

#config t

interface serial 2/0

ip address 20.0.0.10 255.0.0.0

no shutdown

click on router 1 go to CLI and enter similar commands.

After this procedure, the red light between the two routers will now turn green indicating that they are ready for communication.

P.T.O

For connecting 2 devices

Since IP address of PC0 is already configured, go to route no

enable

config t

interface fast ethernet 0/0

ip address 10.0.0.10 255.0.0.0

no shutdown

Red light turns green indicating the router is ready for communication

Teaching router 0 of network 30

no

enable

config t

interface serial 2/0

ip route 30.0.0.0 255.0.0.0 20.0.0.20

exit

show ip routes

Repeat the same for network 40 as well as router 1 and router 2.

Observation:

1 router

When PC0 pings PC1 for the first time we get the request <sup>time out</sup> for 1<sup>st</sup> packet. If we ping PC1 from PC0 again we get all 4 packets without any loss. Reverse pinging will also not lead to any loss.

3 routers

Before training the routers regard the result as destination is unreachable. After training the routers we get clear statistics on the result.

Results:

Using single router

Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Request timed out

Reply from 20.0.0.1 bytes = 32 time < 1ms

Reply from 20.0.0.1 bytes = 32 time < 1ms

Reply from 20.0.0.1 bytes = 32 time < 1ms

Reply from 20.0.0.1 bytes = 32 time < 1ms

Ping statistics from 20.0.0.1

packets = 4 received = 4 lost = 0

Using 3 routers, 2 PCs

Ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data

Reply from 10.0.0.10 bytes = 32 time = 0 ms

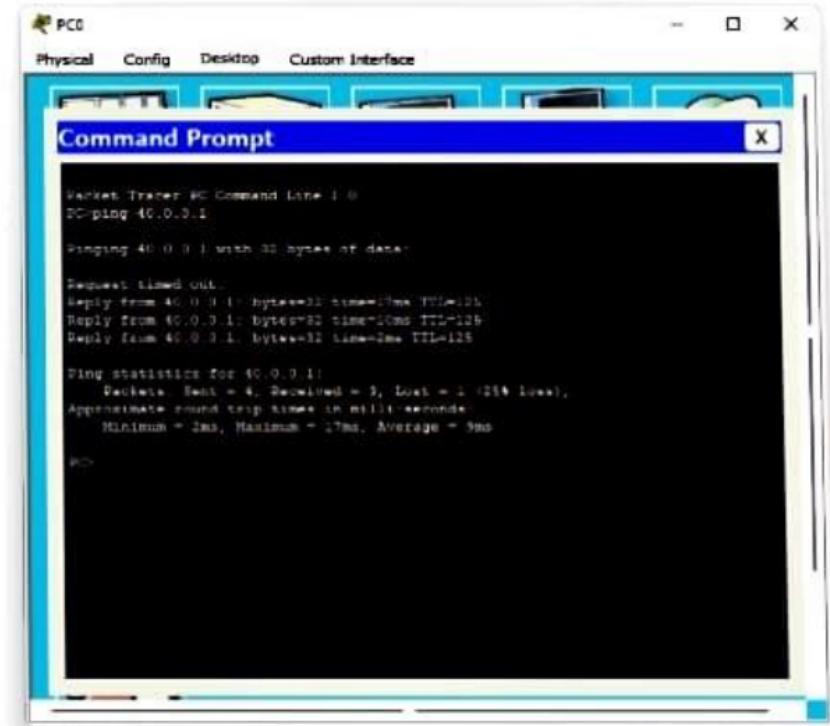
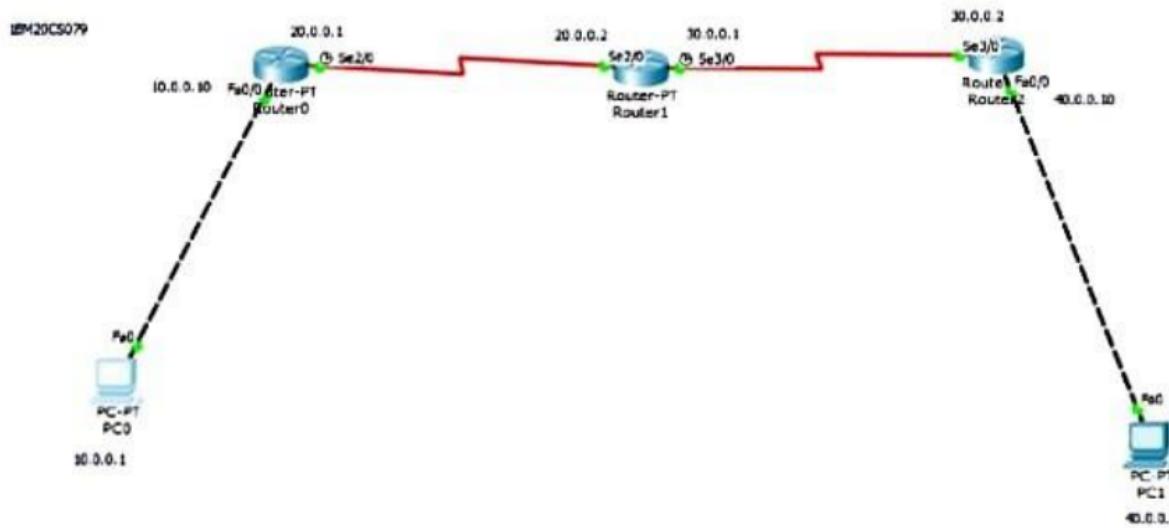
Reply from 10.0.0.10 bytes = 32 time = 0 ms

Reply from 10.0.0.10 bytes = 32 time = 0 ms

Reply from 10.0.0.10 bytes = 32 time = 0 ms

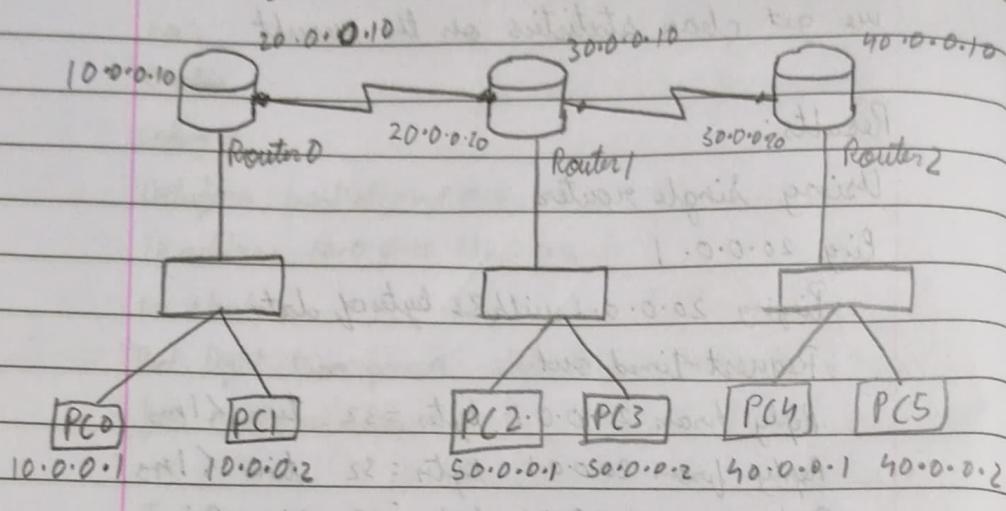
Ping statistics for 40.0.0.1

packet sent = 5 received = 5 lost = 0



AIM: Configuring default route to the router

### Topology



### Procedure:

- ① Use 3 gen routers and 6 gen PCs in the workspace along with 3 switches.
- ② Place a rat for each device and specify the IP addresses.
- ③ ~~Connect~~ Use copper straight wires to connect router and switch.
- ④ Use copper straight wires to connect switch and PC.
- ⑤ Click on a PC to set attributes for a PC and each PC has 3 attributes subnet mask, IP address and gateway. This needs to be done for all 6 PCs.
- ⑥ For router 1 the config is done in the CLI. The IP address and subnet mask is set. Router 2 is a default router for router 1 and this is done by command `ip route 0.0.0.0 0.0.0.0 40.0.0.2`
- ⑦ Router 2 IP address and subnet mask are set for all 3 interfaces. It has static routing done by commands.

- ④ Router 3 is config us both ~~to~~ interface with IP address and subnet mask. The default router for router 3 is router 2
- ⑤ ping command is executed from 10.0.0.1 to 20.0.0.1 and from 10.0.0.1 to 30.0.0.2

### Observation

- 1 router can have 2 default router
- Default router for router 3 is also middle router
- Middle router gets no default router because if one router is made default then there are chances of packets going to switch are sent to the router.

### Result

Ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data

Request timed out after 3 sec

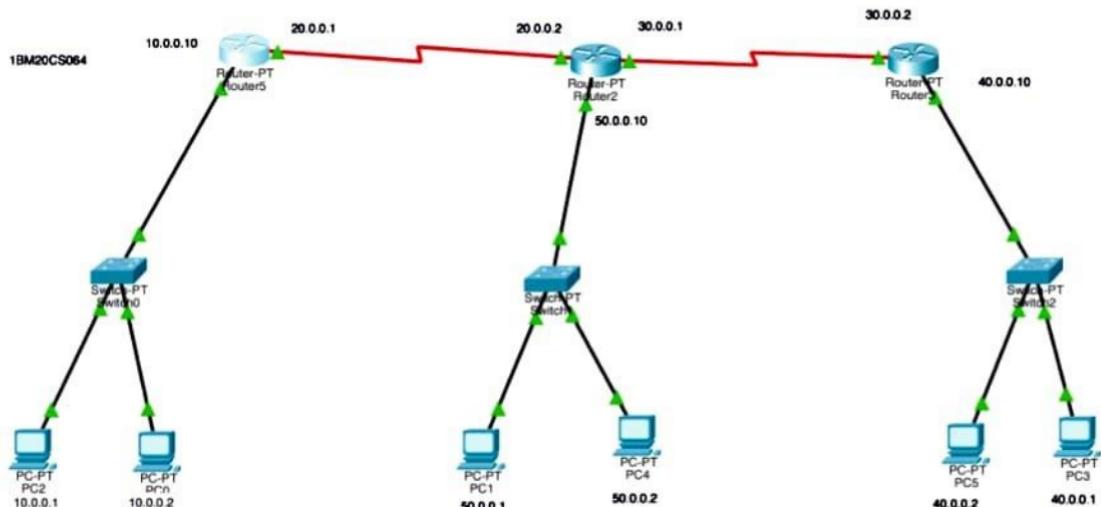
Reply from 30.0.0.2 bytes = 32, time = 4 ms TTL = 123

Reply from 30.0.0.2 bytes = 32, time = 4 ms TTL = 125

Reply from 30.0.0.2 bytes = 32, time = 4 ms TTL = 123

Request timed out after 3 sec

## Cisco Packet Tracer - /Users/joanimegod/Downloads/3 routers default.pkt



Time: 00:01:39

Realtime Simulation



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
Scenario 0	-	PC2	PC1	IC...	Black	0.000	N	0	(...)	(delete)

Toggle PDU List Window

(Select a Device to Drag and Drop to the Workspace)





```

Readonly ROMMON initialized
Self decompressing the Image ...
***** (OK)

Restricted Rights Legend

The duplication or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (ii) (iii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco Internetwork Operating System Software
IOS (TM) PT1000 Software (PT1000-I-M), Version 12.2(28), RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:01 by miwang

PT 1001 (PTSC2005) processor (revision 6a200) with 60416K/5120K bytes of memory

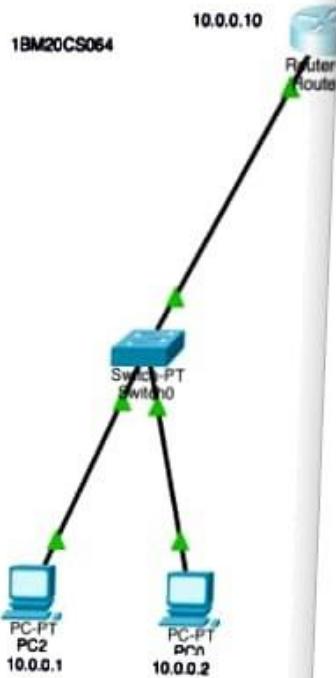
Processor board ID PT0123 (0123)
PT2005 processor part number 0, mask 31
Bringing software.
X.25 software, Version 3.0.0.
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial (sync/asynch) network interface(s)
128 bytes of non-volatile configuration memory.
43488K bytes of ATA Compaqflash (Read/Write)

Press RETURN to get started!

*LINE-0-CHANGED: Interface Serial2/0, changed state to up
*LINEPROTO-0-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*LINEPROTO-0-UPDOWN: Line protocol on interface Serial2/0, changed state to up

```





```

Readonly ROMMON initialized
Self decompressing the Image ...
***** (OK)

Restricted Rights Legend

The duplication or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (ii) (iii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco Internetwork Operating System Software
IOS (TM) PT1000 Software (PT1000-I-M), Version 12.2(28), RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:01 by miwang

PT 1001 (PTSC2005) processor (revision 6a200) with 60416K/5120K bytes of memory

Processor board ID PT0123 (0123)
PT2005 processor part number 0, mask 31
Bringing software.
X.25 software, Version 3.0.0.
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial (sync/asynch) network interface(s)
128 bytes of non-volatile configuration memory.
43488K bytes of ATA Compaqflash (Read/Write)

Press RETURN to get started!

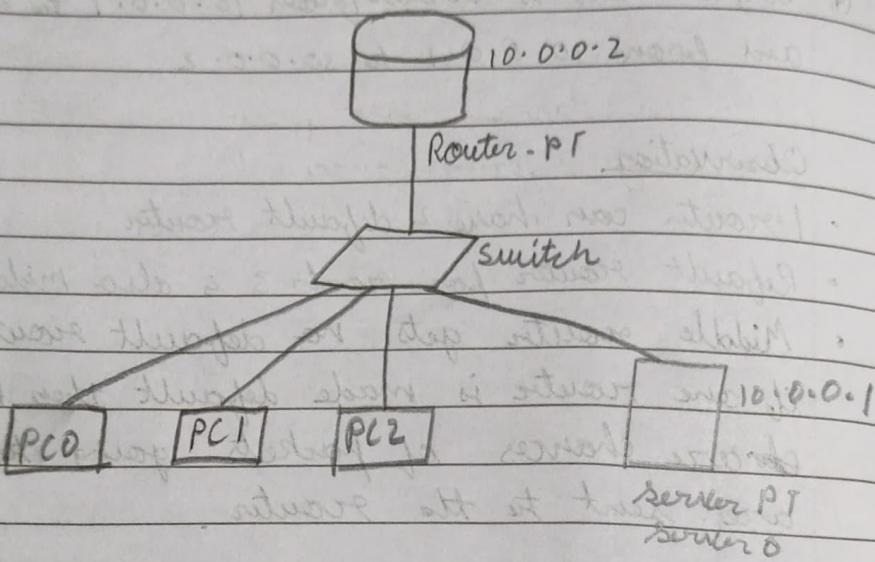
*LINE-0-CHANGED: Interface Serial2/0, changed state to up
*LINEPROTO-0-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*LINEPROTO-0-UPDOWN: Line protocol on interface Serial2/0, changed state to up

```



Aim: Configuring DHCP within a LAN in a packet tracer

### Topology



### Procedure

- ① Use one gen router one gen switch one gen server and 3 gen PCs in the workspace.
- ② Use copper straight wire to connect PCs and server.
- ③ Use fibre to connect switch and router.
- ④ Configure the server by adding IP address, subnet mask and gateway.
- ⑤ Config the router by setting IP address, subnet masks and executing this command in CLI  
ip route 10.0.0.2 255.0.0.0
- ⑥ click on server and go to services, select ~~network~~ DHCP switch it on and add the gateway servers, if address and subnet mask.

Desktop Tab

- ② Click on the 1<sup>st</sup> PC and go to IP config. Select DHCP  
Repeat this for all.
- ③ Ping command is executed from 10.0.0.3 to 10.0.0.5

Observation

pool of ip addresses exist from which IP addresses can be dynamically allocated.

This is called Dynamic Host Configuration Protocol (DHCP).

Result

ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data

Reply from 10.0.0.5 bytes = 32 time = 2 ms TTL = 125

Reply from 10.0.0.5 bytes = 32 time = 2 ms TTL = 125

Reply from 10.0.0.5 bytes = 32 time = 2 ms TTL = 125

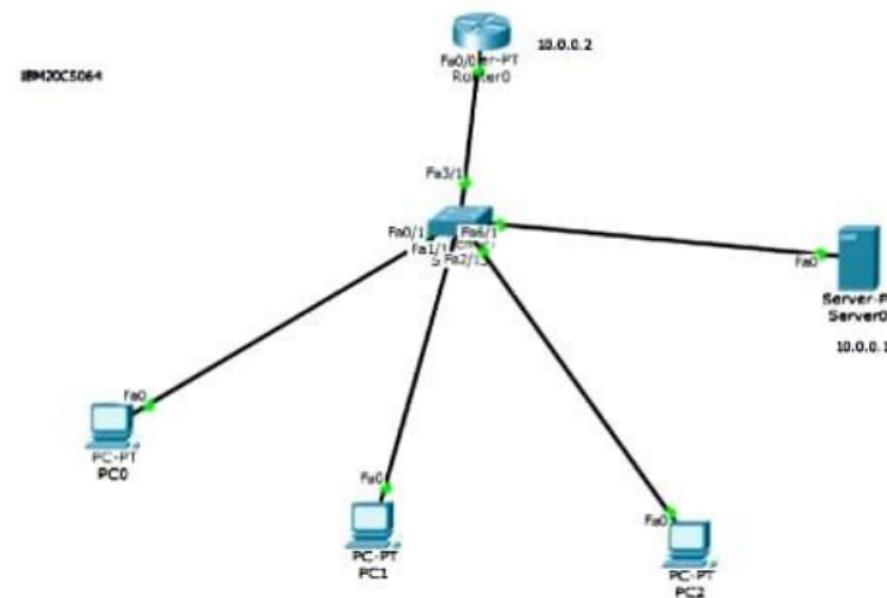
Reply from 10.0.0.5 bytes = 32 time = 2 ms TTL = 125

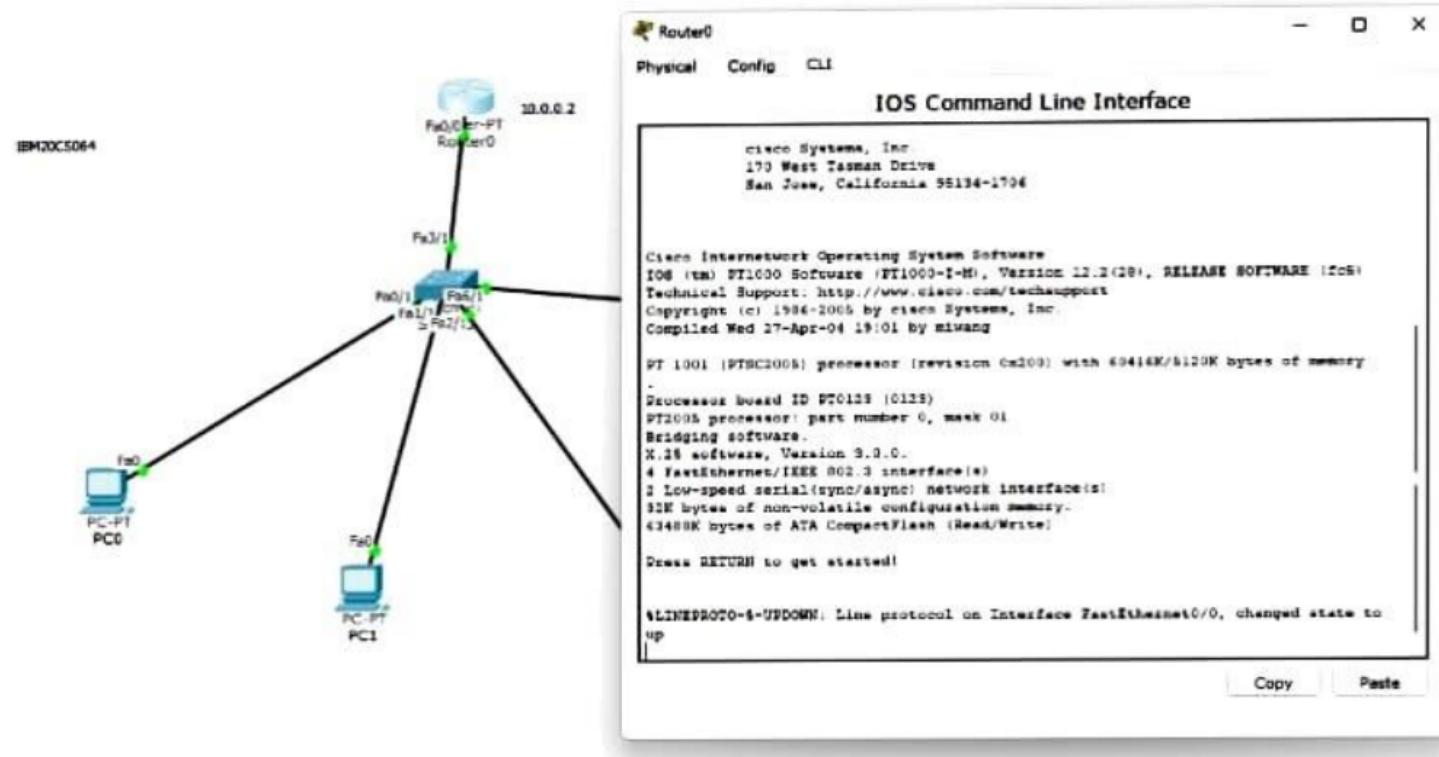
✓ Ping statistics for 10.0.0.5

packets sent = 4, received = 4, lost = 0

✓ 8/12/22

✓





**Command Prompt**

```
Packet tracer PC Command Line 1.0
```

```
PC>ping 10.0.0.5
```

```
Pinging 10.0.0.5 with 32 bytes of data:
```

```
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
```

```
Ping statistics for 10.0.0.5:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

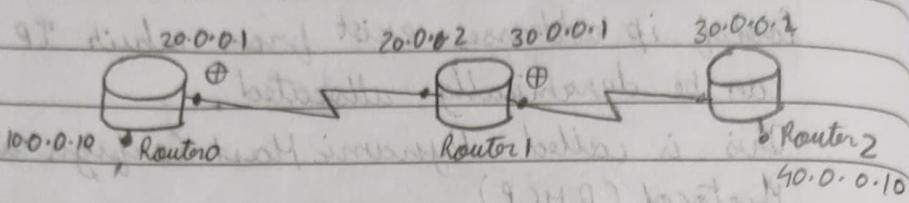
```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
PC>
```

Aim: Configuring RIP Routing Protocol in Routers

Topology:



PC0

PC1

10.0.0.1

40.0.0.10

~~Procedure :-~~

- ✓ Use 3 generic routers, 2 generic PC and place notes to indicate respective IP addresses
- Use serial DCE cable to connect routers and use cross cable to connect PC with router1 and router3
- Set IP addresses, gateway and subnet mask as 10.0.0.10, 255.0.0.0 for PC0 and set 40.0.0.1, 255.0.0.1 for PC1
- interface PC0 and router1
  - interface fastethernet 0/0
  - IP address 20.0.0.10 255.0.0.0
  - no shut

- for interfacing serial 2/0 of router 1 use:
  - interface serial 2/0
  - IP address 20.0.0.1 255.0.0.0
  - encapsulation PPP
  - clock rate 64000
  - no sheet
- Use above commands for interfacing router which has clock symbol in cable hear to it and for other interfaces of routers use same above command except "clock rate 64000".
- Once all the lights are turned green follow the commands below each router
  - router rip
  - network 10.0.0.0
  - network 20.0.0.0
  - exit
- Repeat the same command for router 2 and router 3

#### Observation:

Use RIP routing becomes easy when large number of routers are present

#### Result:

Pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.1 byte = 32

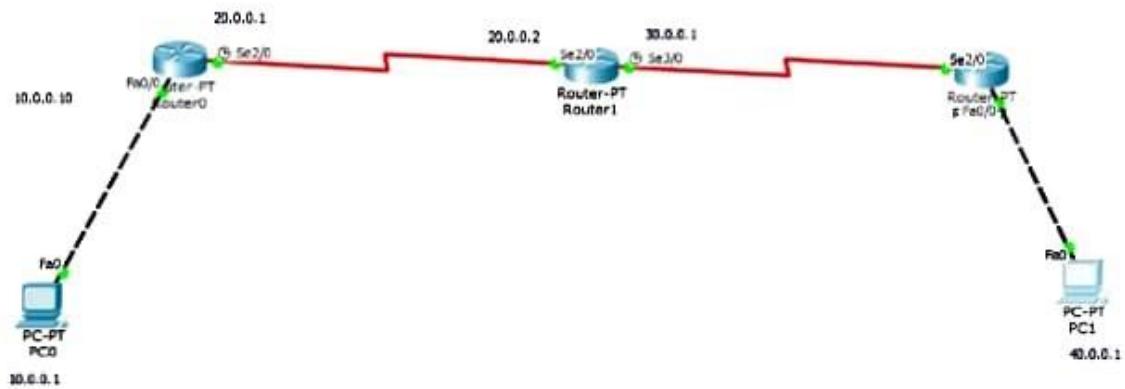
✓ 12/27 reply from 10.0.0.1 byte = 32

reply from 10.0.0.1 byte = 32

reply from 10.0.0.1 byte = 32

ping statistics for 10.0.0.1

packets: sent = 4, received = 4, lost = 0





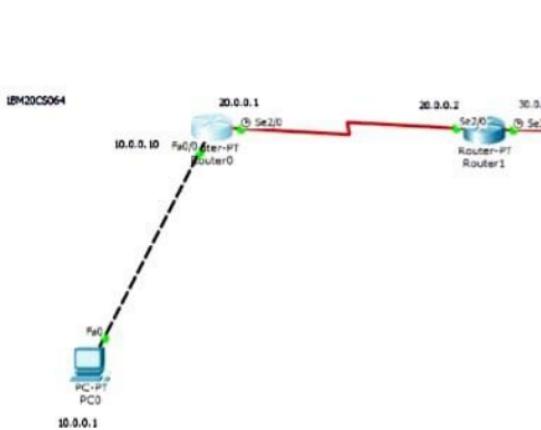
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



**Route0**

Physical Config CLI

### IOS Command Line Interface

```
Cisco Internetwork Operating System Software
IOS (TM) PT1000 Software (W7100S-I-M), Version 12.2(28), RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:31 by mhwang

ZT101 (PTBC200) processor (revision 0a00) with 40416K/5120K bytes of memory

Processor board ID ZT0123 (0123)
ZT2005 processor; part number 0, mask 02
Bridging software
X.25 software, Version 3.0.0.
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial (vnr/vsnc) network interface(s)
32K bytes of non-volatile configuration memory.
41488K bytes of ATA CompactFlash (Read/Write)

Press RETURN to get started!

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINE-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
```

Copy Paste

Time: 00:26:38 | Power Cycle Devices Test Forward Time



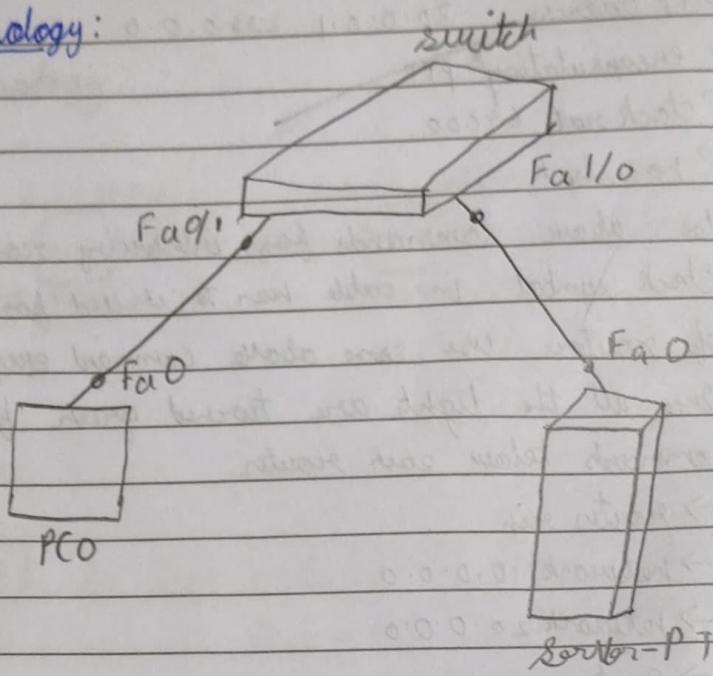
Scenario 0	New	Delete	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete

Toggle PDU List Window

(Select a Device to Drag and Drop to the Workspace)

Aim: Demonstration of Web service and DNS

Topology:



Procedure:

- Place an PC, server and switch and then set the IP addresses of PC and server as 10.0.0.1 and 10.0.0.2.
- Open web browser ~~in~~ from desktop tab of the PC and type "http://10.0.0.2"
- It will display a default page
- Open server and in services enable HTTP and change or add the contents of in the HTTP.
- After save refresh the browser of PC0 to the updated changes.

### Activate DNS:-

- enable DNS on the server to activate it
- Enter the name and the IP address needed to be mapped
- Click on add to add the new mapping.
- Now give the name in the web browser you saved to check if its working.

### Custom page:-

- Create a new page ~~test~~.html and save it in http services
- Change hyperlink in index.html to link the created file.
- Check the output in the web browser of the PC by clicking on hyper link

### Observations:-

- We can view the web page when we type 'csbmsc' in browser because 10.0.0.2 address is mapped to the name 'csbmsc'.  
Mapping is required because its difficult for users to remember IP's. Hence it is mapped.

## Result:

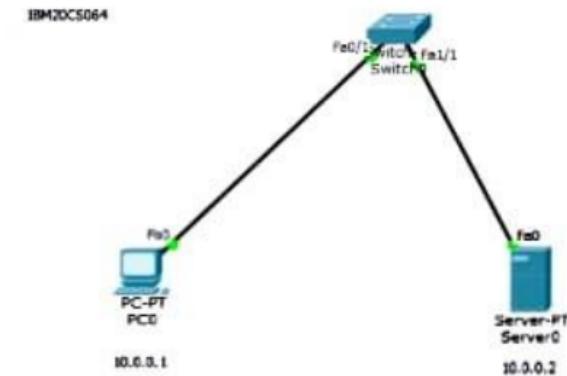
Web Browser

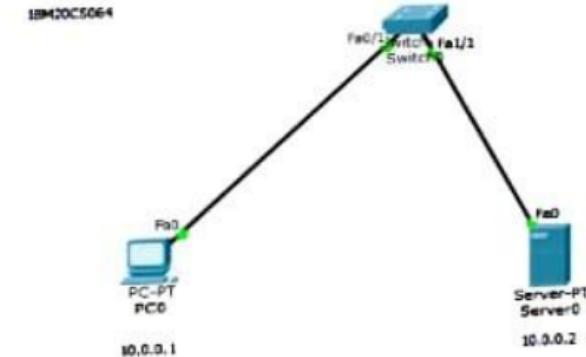
<URL>

http://cn/resume.html

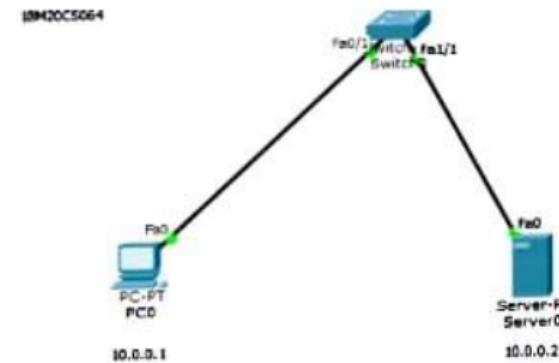
	Name	Yogirajya
	age	20
	College	BMSCE
	Experience	2 years

✓  
29/12/22

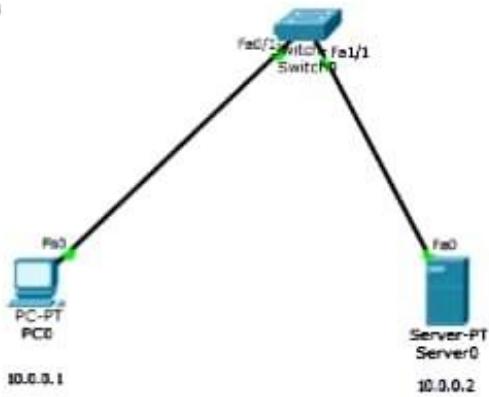




Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
--------	-------------	------	-------	-----------	----------	-----	------	--------



IBM20C5064



# Lab - 7

Date \_\_\_\_\_  
Page \_\_\_\_\_

AIM: Write a program for error detection by CRC

(n: size; p: poly)

(state: initial state)

3. ab

```
#include <stdio.h> // header file
#include <string.h> // header file
#define N strlen(gen_poly) // header file
char data[28]; // header file
char check_value[28]; // header file
char gen_poly[10]; // header file
int data_length, i, j; // header file
void XOR() { // function declaration
    for (j = 1; j < n; j++) {
        check_value[j] = ((check_value[j] == gen_poly[j]) ? // condition
            0 : 1); // assignment
    }
}
void receiver() { // function declaration
    printf("Enter the received data: ");
    scanf("%s", data); // input
    printf("\n-----\n");
    printf("Data received: %s", data); // output
    crc(); // function call
    for (i = 0; i < N - 1) && (check_value[i] != '1'); i++)
        if (i < N - 1)
            printf("\n Error Detected \n\n");
        else
            printf("\n No Error Detected \n\n");
}

```

S-Dat

```

void crc() {
    for (i=0; i<N; i++)
        check_value[i] = data[i];
    do {
        if (check_value[0] == '1') {
            xor();
            for (j=0; j<N; j++)
                check_value[j] = check_value[j+1];
            check_value[0] = data[i++];
        }
    } while (i <= data_length+N-1);
}

```

int main()

{

```

printf("Enter data to be transmitted: ");
scanf("%s", data);
printf("Enter the gen polynomial: ");
scanf("%s", gen_poly);
data_length = strlen(data);
for (i=data_length; i<data_length+N-1; i++)
    data[i] = '0';
printf("Data padded with n-1 zeros: %s", data);
printf("CRC: ");
crc();
printf("In CRC or check value is: %s", check_value);
for (i=data_length; i<data_length+N-1; i++)
    data[i] = check_value[i-data_length];
printf("Inferred data to be sent: %s", data);
printf("\n-----\n");
receiver();
return 0;
}

```

Output

Enter the data : 10001000 000100001

Enter poly : 1011101

Data padded with  $n-1$  : 100010000001000000000

CRC or check value : 010011

Final data sent : 10001000000100001010011

Enter the received data : 10001000000100001010011

Data received : 10001000000100001010011

No error Detected

(0<0) ✓

29/12/22  
base of if " >> n2;" knot +/+ / n1" >> two3

\* before while loop in line 2 +/+ / n1" >> two3

( ) many tri

; switch... if tri  
\*: store higher after " >> two3

(++i ; z>i ; i= ; tri) naf

• (01 \* 0 know) naf

Enter data to be transmitted: 10001000000100001

Enter the Generating polynomial: 1011101

---

Data padded with n-1 zeros : 10001000000100001000000

---

CRC or Check value is : 010011

---

Final data to be sent : 10001000000100001010011

---

Enter the received data: 10001000000100001010011

---

Data received: 10001000000100001010011

No error detected

## Leaky bucket

```
#include<iostream>
using namespace std;
int main()
{ cout << "Enter bucket size" << endl;
  int bucketsize;
  int filled = 0;
  int outputrate;
  int inputpacket;
  int choice;
  cin >> bucketsize;
  cout << "Enter outputrate" << endl;
  cin >> outputrate;
  do {
    cout << "Enter packet size" << endl;
    cin >> inputpacket;
    if (inputpacket <= bucketsize)
    { if (filled + inputpacket > bucketsize)
      { cout << "Packets too big for bucket" << endl;
        }
      else
        { filled = filled + inputpacket;
          }
    }
    else
    { cout << "Packets too big for bucket" << endl;
      }
    if (filled <= outputrate)
    { filled = 0;
      }
    else
    { filled = filled - outputrate outputrate;
      }
  } while (choice != 2);
}
```

```
cout << "Amount of bucket filled" << filled << endl;  
cout << "Do you want to enter another packet (1 for yes, 2 for no)"  
<< endl;
```

```
cin >> choice;
```

```
} while(choice == 1);  
}
```

## Output

Enter bucket size : 500

Enter output rate : 100

Enter packet size : 700

Packets too big for bucket

Amount of bucket filled 0

Do you want to enter another packet (1 for yes, 2 for no)

1

Enter packet size

200

Amount of bucket filled 100

Do you want to enter another packet (1 for yes, 2 for no)

1

Enter packet size 300

Amount of bucket filled 300

Do you want to enter another packet (1 for yes, 2 for no)

1

Enter packet size 100

Amount of bucket filled 300

Do you want to enter another packet (1 for yes, 2 for no)

2

```
Enter bucket size  
500  
Enter output rate  
100  
Enter packet size  
700  
Packets too big for bucket  
Amount of bucket filled 0  
Do you want to enter another packet(1 for yes, 2 for no)  
1  
Enter packet size  
200  
Amount of bucket filled 100  
Do you want to enter another packet(1 for yes, 2 for no)  
1  
Enter packet size  
300  
Amount of bucket filled 300  
Do you want to enter another packet(1 for yes, 2 for no)  
1  
Enter packet size  
100  
Amount of bucket filled 300  
Do you want to enter another packet(1 for yes, 2 for no)  
2
```

Bellman

AIM: write a C prog for Bellman Ford algorithm

#include &lt; stdio.h&gt;

#include &lt; stdlib.h&gt;

int Bellman\_Ford (int G[20][20], int V, int E, int edge[20][20])

{ int i, u, v, k, distance[20], parent[20], s, flag = 1;

for (i=0; i&lt;V; i++)

distance[i] = 1000, parent[i] = -1;

printf("Enter source: ");

scanf("%d", &amp;s);

distance[s-1] = 0;

for (i=0; i&lt;V-1; i++)

{ for (k=0; k&lt;E; k++)

{ u = edge[k][0], v = edge[k][1];

if (distance[u] + G[u][v] &lt; distance[v])

distance[v] = distance[u] + G[u][v], parent[v] = u;

}

y

for (k=0; k&lt;E; k++)

{ u = edge[k][0], v = edge[k][1];

if (distance[u] + G[u][v] &lt; distance[v])

flag = 0;

y

if(flag)

for (i=0; i&lt;V; i++)

printf("Vertex %d \rightarrow cost = %d parent = %d\n", i+1, distance[i], parent[i]+1);

return flag;

y

int main()

{ int v, edge[20][20], G[20][20], i, j, k = 0;

printf("BELLMAN FORD\n");

printf("Enter no. of vertices: ");

scanf("%d", &amp;v);

printf("Enter graph in matrix form: |n|");

for ( $i = 0; i < V; i++$ )

    for ( $j = 0; j < V; j++$ )

}

    scanf("%d", & $[i][j]$ );

    if ( $G[i][j] \neq 0$ )

        edge[k][0] =  $i$ , edge[k++][1] =  $j$ ;

}

if (Bellmann\_Ford(G, V, K, edge))

    printf("\nNo negative weight cycle exists |n|");

return 0;

}

### Output

BELLMAN FORD

Enter the no. of vertices : 4

Enter & graph in matrix form:

0 5 4 999

5 0 6 3

999 3 1 6

2 0 1 4

Enter source: 1

Vertex 1  $\rightarrow$  cost = 0 parent = 0

Vertex 2  $\rightarrow$  cost = 5 parent = 1

Vertex 3  $\rightarrow$  cost = 4 parent = 1

Vertex 4  $\rightarrow$  cost = 8 parent = 2

No negative weight cycle

BELLMAN FORD

Enter no. of vertices: 4

Enter graph in matrix form:

0 5 4 999

5 0 6 3

999 3 1 6

2 0 1 4

Enter source: 1

Vertex 1 -> cost = 0 parent = 0

Vertex 2 -> cost = 5 parent = 1

Vertex 3 -> cost = 4 parent = 1

Vertex 4 -> cost = 8 parent = 2

No negative weight cycle

Dijkstra

AIM: Write a C/C++ program to implement Dijkstra's algorithm.

```
#include <stdio.h>
#include <conio.h>
#define INFINITY 9999
#define MAX 10
void dijkstra (int G[MAX][MAX], int n, int startnode);
int main(){
    int G[MAX][MAX], i, j, n, u;
    printf ("Enter no. of vertices: ");
    scanf ("%d", &n); printf ("\nEnter the adjacency matrix: ");
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            scanf ("%d", &G[i][j]);
    printf ("\nEnter the starting node: ");
    scanf ("%d", &u); dijkstra (G, n, u); return 0;
}
void dijkstra (int G[MAX][MAX], int n, int startnode){
    int cost[MAX][MAX], distance[MAX], pred[MAX];
    int visited[MAX], count, mindistance, nextnode, i, j;
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            if (G[i][j]==0)
                cost[i][j]=INFINITY;
            else
                cost[i][j]=G[i][j];
    for (i=0; i<n; i++)
        distance[i]=cost[startnode][i];
    pred[i]=startnode; visited[i]=0;
    mindistance=0;
    nextnode=startnode;
    count=1;
    while (count<n) {
        mindistance=distance[nextnode];
        for (j=0; j<n; j++)
            if (cost[nextnode][j]<mindistance)
                mindistance=cost[nextnode][j];
        if (mindistance==INFINITY)
            break;
        visited[mindistance]=1;
        for (i=0; i<n; i++)
            if (cost[mindistance][i]<distance[i])
                distance[i]=cost[mindistance][i];
        pred[i]=nextnode;
        count++;
        mindistance=0;
        for (j=0; j<n; j++)
            if (visited[j]==0)
                if (cost[nextnode][j]<mindistance)
                    mindistance=cost[nextnode][j];
        if (mindistance==INFINITY)
            break;
        visited[mindistance]=1;
        for (i=0; i<n; i++)
            if (cost[mindistance][i]<distance[i])
                distance[i]=cost[mindistance][i];
        pred[i]=nextnode;
        count++;
    }
}
```

```
while (count < n-1){  
    mindistance = INFINITY;  
    for (i=0; i<n; i++)  
        if (distance[i] < mindistance && !visited[i]) {  
            mindistance = distance[i]; nextnode = i;  
        }  
    visited[nextnode] = 1;  
    for (i=0; i<n; i++)  
        if (!visited[i])  
            if (mindistance + cost[nextnode][i] < distance[i])  
                { distance[i] = mindistance + cost[nextnode][i];  
                  pred[i] = nextnode;  
                }  
    count++;  
}  
for (i=0; i<n; i++)  
    if (i != startnode) {  
        printf ("\nDistance of node %d = %.d", i, distance[i]);  
        printf ("\nPath = %.d", i); j = i;  
        do {  
            j = pred[j];  
            printf (" - %.d", j);  
        } while (j != startnode);  
    }  
}
```

## Output

Enter no. of vertices : 4

Enter the adjacency matrix:

0	5	4	9 9 9
5	0	6	3
9 9 9	3	1	6
2	0	1	4

Enter the starting node: 1

Distance of node 0 = 5

path = 0 <= 1

Distance of node 2 = 4

path = 2 <= 3 <= 1

Distance of node 3 = 3

Path = 3 <= 1

Enter no. of vertices:4

Enter the adjacency matrix:

0 5 4 999

5 0 6 3

999 3 1 6

2 0 1 4

Enter the starting node:1

Distance of node0=5

Path=0<-1

Distance of node2=4

Path=2<-3<-1

Distance of node3=3

Path=3<-1

...Program finished with exit code 0

Press ENTER to exit terminal

for Client

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
```

2

```
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('From Server: \n')
print(filecontents)
clientSocket.close()
```

Output

Enter file name: server.py

From server:

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
```

while 1:

```
print("The server is ready to receive")
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file = open(sentence, "r")
l = file.read(1024)
connectionSocket.send(l.encode())
print('\nSend contents of ' + sentence)
file.close()
connectionSocket.close()
```

Server

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('In sent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

Output

The server is ready to receive

Sent contents of server.py

The server is ready to receive

File Edit Shell Debug Options Window Help

Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]  
on win32  
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:/Users/mdsur/Desktop/server.py =====

The server is ready to receive

File Edit Shell Debug Options Window Help

Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]  
on win32  
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:/Users/mdsur/Desktop/client.py =====

Enter file name: server.py

From Server:

```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

>>>

File Edit Shell Debug Options Window Help

Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]  
on win32  
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:/Users/mdsur/Desktop/server.py =====

The server is ready to receive

Sent contents of server.py

The server is ready to receive

Server UDP	AIM: Using UDP sockets. write a client server prog
<code>from socket import *</code>	
<code>serverPort = 12000</code>	
<code>serverSocket = socket (AF_INET, SOCK_DGRAM)</code>	
<code>serverSocket.bind (( "127.0.0.1", serverPort ))</code>	
<code>print ("The server is ready to receive")</code>	
<code>while 1 :</code>	
<code>    sentence, clientAddress = serverSocket.recvfrom (2048)</code>	
<code>    sentence = sentence.decode ("Utf-8")</code>	
<code>    file = open (sentence, "r")</code>	
<code>    l = file.read (2048)</code>	
<code>    serverSocket.sendto (bytes (l, "Utf-8"), clientAddress)</code>	
<code>    print ('\\n sent contents of ', end = ' ')</code>	
<code>    print (sentence)</code>	
<code>    file.close</code>	

## Output

The server is ready to receive

Sent contents of server UDP.py

## Client UDP

`from socket import *`

`serverName = "127.0.0.1"`

`serverPort = 12000`

`clientSocket = socket (AF_INET, SOCK_DGRAM)`

`sentence = input ("\\nEnter file name: ")`

`clientSocket.sendto (bytes (sentence, "Utf-8"), (serverName, serverPort))`

`fileContents, serverAddress = clientSocket.recvfrom (2048)`

`print ('\\n reply from server: \\n')`

```
print(filecontents.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

## Output

Enter file name: Server-UDP.py

Reply from sender:

```
from socket import *
```

```
serverPort = 12000
```

```
serverSocket = socket(AF_INET, SOCK_DGRAM)
```

```
serverSocket.bind(("127.0.0.1", serverPort))
```

```
print("The server is ready to receive")
```

while True:

```
sentence, clientAddress = serverSocket.recvfrom(2048)
```

```
sentence = sentence.decode("utf-8")
```

```
file = open(sentence, "r")
```

```
l = file.read(2048)
```

```
serverSocket.sendto(bytes(l, "utf-8"), clientAddress)
```

```
print('In sent contents of ', end = ' ')
```

```
print(sentence)
```

```
# for i in sentence:
```

```
# print(str(i), end = ' ')
```

```
file.close()
```



C:\Windows\System32\cmd.e



+ ▾



C:\Users\mdsur\Desktop\UDP>python -u serverUDP.py  
The server is ready to receive

Sent contents of serverUDP.py



```
C:\Users\mdsur\Desktop\UDP>python -u clientUDP.py
```

```
Enter file name: serverUDP.py
```

```
Reply from Server:
```

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:      ↗
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence, "r")
    l=file.read(2048)
    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print ('\nSent contents of ', end = ' ')
    print (sentence)
# for i in sentence:
#     print (str(i), end = '')
    file.close()
```

```
C:\Users\mdsur\Desktop\UDP>
```