REVA UNIVERSITY
Bengaluru, India

SCHOOL OF COMPUTING AND INFORMATIONTECHNOLOGY

COMPUTER NETWORKS LABORATORY MANUAL
[B20CI0506]

B.TECH III YEAR – V SEM
[A.Y:2024-2025]

B. Tech in Artificial Intelligence and Machine Learning

INDEX

REVA UNIVERSITY

VISION OF THE UNIVERSITY

- "REVA University aspires to become an innovative university by developing excellent human resources with leadership qualities, ethical and moral values, research culture and innovative skills through higher education of global standards".

MISSION OF THE UNIVERSITY

- To create excellent infrastructure facilities and state-of-the-art laboratories and incubation centers
- To provide student-centric learning environment through innovative pedagogy and education reforms
- To encourage research and entrepreneurship through collaborations and extension activities
- To promote industry-institute partnerships and share knowledge for innovation and development
- To organize society development programs for knowledge enhancement in thrust areas
- To enhance leadership qualities among the youth and enrich personality traits, promote patriotism and moral values.

Program: B.Tech in CSE [Artificial Intelligence and Machine Learning]

VISION OF THE SCHOOL

To produce excellent quality technologists and researchers of global standards in computing and Information technology who have potential to contribute to the development of the nation and the society with their expertise, skills, innovative problem-solving abilities, strong moral and ethical values.

MSSION OF THE SCHOOL

- To create state of the art computing labs infrastructure and research facilities in information technology.
- To provide student-centric learning environment in Computing and Information technology through innovative pedagogy and education reforms.
- To encourage research, innovation and entrepreneurship in computing and information technology through industry/academia collaborations and extension activities
- Organize programs through club activities for knowledge enhancement in thrust areas of information technology.
- To enhance leadership qualities among the youth and enrich personality traits, promote patriotism, moral and ethical values.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

After few years of graduation, the graduates of B. Tech Computer Science & Engineering (AI & ML) will:

PEO-1: Demonstrate technical skills, competency in AI & ML and exhibit team management capability with proper communication in a job environment.

PEO-2: Support the growth of economy of a country by starting enterprise with a lifelong learning attitude

PEO-3: Carry out research in the advanced areas of AI & ML and address the basic needs of the society.

PROGRAM SPECIFIC OUTCOMES (PSO's)

On successful completion of the program, the graduates of B. Tech CSE(AIML) program will be able to:

- **PSO-1:** Demonstrate the knowledge of human cognition, Artificial Intelligence, Machine Learning and data engineering for designing intelligent systems.
- **PSO-2:** Apply computational knowledge and project development skills to provide innovative solutions.
- **PSO-3:** Use tools and techniques to solve problems in AI & ML.

PROGRAM OUTCOMES (PO'S)

On successful completion of the program, the graduates of B. Tech CSE (AIML) program will be able to:

- **PO-1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals for the solution of complex problems in Computer Science and Engineering.
- **PO-2: Problem analysis:** Identify, formulate, research literature, and analyze engineering problems
- To arrive at substantiated conclusions using first principles of mathematics, natural, and engineering sciences.
- **PO-3: Design/development of solutions:** Design solutions for complex engineering problems and design system components, processes to meet the specifications with consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **PO-4: Conduct investigations of complex problems:** Use research-based knowledge including design of Experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

- **PO-5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- **PO-6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **PO-7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- **PO-8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
- **PO-9: Individual and team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
- **PO-10: Communication:** Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.
- **PO-11: Project management and finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.
- **PO-12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Course Details

1. Course Objectives:

The main objectives of this course are:

1. Explain the protocol stacks (OSI and TCP/IP) for data communication
2. Discuss the error detection & correction strategies for data transmission.
3. Design the connection establishment of network computing devices.
4. Illustrate the TCP, UDP protocols and explain Domain Name System.
5. Emphasis the management of local area networks
6. Learning about computer network organization and implementation

2. Lab Requirements:

Following are the required hardware and software for this lab, which is availablein

the laboratory.

Minimum System requirements:

- Processors: Intel® Core™ i5 Core processor.
- Disk space: 1 GB.
- Operating systems: Windows and Linux.
- CISCO Packet Tracer, GCC complier (C/C++)

About the Lab:

The main emphasis of this course is on the organization and management of local area networks(LANs).The course description include learning about computer network organization and implementation, obtaining a the or etical understanding of data communication and computer networks, and about Open Systems Interconnection(OSI) communication model with TCP/IP protocol; This course provides knowledge of error detection and recovery; local area networks; bridges, routers and gateways; network naming and addressing; and local and remote procedures. This course also emphasis on User Datagram Protocol, TCP Congestion Control; DNS Message Formatting and Remote Login. Protocols

3. Guidelines to Students

- Equipment in the lab for the use of student community. Students need to maintaina proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.
- Students are required to carry their observation / programs book with completed exercises while entering the lab.
- Students are supposed to occupy the machines allotted to them and are not supposed to talk or make noise in the lab. The allocation is put up on the lab noticeboard.
- Lab can be used in free time / lunch hours by the students who need to use the systemsshould take prior permission from the lab in-charge.
- Lab records need to be submitted on or before date of submission.
- Students are not supposed to use flash drives.

Instructions to maintain the record

- Before start of the first lab they have to buy the record and bring the record to the lab.
- Regularly (Weekly) update the record after completion of the experiment and get itcorrected with concerned lab in-charge for continuous evaluation.

- In case the record is lost inform the same day to the faculty in charge and get the new record within 2 days the record has to be submitted and get it corrected by the faculty.
- If record is not submitted in time or record is not written properly, the evaluation marks(5M) will be deducted.

General laboratory instructions

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with: a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session. b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab. c. Proper Dress code and Identity card.
4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty. 5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
5. All the students should be polite and cooperative with the laboratory staff, must maintainthe discipline and decency in the laboratory.\
6. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
7. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
8. Students must take the permission of the faculty in case of any urgency to go out; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.

Course Outcomes (COs)

After the completion of the course, the student will be able to:

COs	Course Outcomes
B20CI0506.1	Make use of the architectural principles of computer networking and compare different approaches to organizing networks.
B20CI0506.2	Identify the good network design with simplicity, scalability, performance and the end-to-end principle.
B20CI0506.3	Appraise the working principles of Internet.
B20CI0506.4	Develop applications using network protocols.

B20CI0506.5	Emphasis the management of local area networks
B20CI0506.6	Learning about computer network organization and implementation

Conduction of Practical Examination:

1. All laboratory experiments (Part A No. 1 to 4 and Part B No. 1 to 5) are included for the syllabus of practicalexamination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution: Procedure + Conduction + Viva: 08 + 35 +07 = 50 Marks

Change of experiment is allowed only once and marks allotted to the procedure part tobe made zero.

CO-PO-PSO MAPPING

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
B20CI0506.1	3	3	3	3	2	1						1	3		
B20CI0506.2	3	3	3	3	2	2						1	3		
B20CI0506.3	3	3	3	3	2	1						2	3		
B20CI0506.4	3	3	3	2	2	1						1	3	3	3
B20CI0506.5	3	3	3	2	2	1						2		3	3
B20CI0506.6	3	3	3	2	2	1						2	2	3	3

Week wise Evaluation of Each Program

ACTIVITY	MARKS
Observation +Viva	20
Record	10
TOTAL	30

INTERNAL ASSESSMENT EVALUATION

(End of Semester)		
Sl No	**ACTIVITY**	**MARKS**
01	Procedure	7
02	Conduction	8
03	Viva Voce	5
	Total	20

FINAL INTERNAL ASSESSMENT CALCULATION		
Sl. No	**ACTIVITY**	**MARKS**
01	Average of weekly Entries	30
02	Internal Assessment Reduced to	20
	Total	50

PART A

Program 1:

Write a program for error detecting code using CRC-CCITT (16- bits).

Whenever digital data is stored or interfaced, data corruption might occur. Since the beginning of computer science, developers have been thinking of ways to deal with this type of problem. For serial data they came up with the solution to attach a parity bit to each sent byte. This simple detection mechanism works if an odd number of bits in a byte changes, but an even number of false bits in one byte will not be detected by the parity check. To overcome this problem developers have searched for mathematical sound mechanisms todetect multiple false bits. The CRC calculation or cyclic redundancy check was the result of this. Nowadays CRC calculations are used in all types of communications. All packets sent over a network connection are checked with a CRC. Also each data block on your hard disk has a CRC value attached to it. Modern computer world cannot do without these CRC calculations. So let's see why they are so widely used. The answer is simple; they are powerful, detect many types of errors and are extremely fast to calculate especially when dedicated hardware chips are used.The idea behind CRC calculation is to look at the data as one large binary number. This number is divided by a certain value and the remainder of the calculation is called the CRC. Dividing in the CRC calculation at first looks to cost a lot of computing power, but it can be performed very quickly if we use a method similar to the one learned at school. We will as an example calculate the remainder for the character 'm'—which is 1101101 in binary notation— by dividing it by 19 or 10011. Please note that 19 is an odd number. This is necessary as we will see further on. Please refer to your schoolbooks as the binary calculation method here is not very different from the decimal method you learned when you were young. It might only look a little bit strange. Also notations differ between countries, but the method is similar.

```
1 0 1
----------------
10011 / 1 1 0 1 1 0 1
1 0 0 1 1 | |
--------- –---| |
1 0 0 0 0 |
0 0 0 0 0 |
---------- |
1 0 0 0 0 1
1 0 0 1 1
----------------
1 1 1 0 = 14 remainder
```

• The message bits are appended with c zero bits; this augmented message is the dividend
• A predetermined c+1-bit binary sequence, called the generator polynomial, is the divisor
• The checksum is the c-bit remainder that results from the division operation

With decimal calculations you can quickly check that 109 divided by 19 gives a quotient of 5 with 14 as the remainder. But what we also see in the scheme is that every bit extra to check only costs one binary comparison and in 50% of the cases one binary subtraction. You can easily increase the number of bits of the test data string—for example to 56 bits if we use our example value "Lammert"—and the result can be calculated with 56 binary comparisons and an average of 28 binary subtractions. This can be implemented in hardware directly with only very few transistors involved. Also software algorithms can be very efficient. All of the CRC formulas you will encounter are simply checksum algorithms based on modulo-2 binary division where we ignore carry bits and in effect the subtraction will be equal to an exclusive or operation. Though some differences exist in the specifics across different CRC formulas, the basic mathematical process is always the same:

Table 1 lists some of the most commonly used generator polynomials for 16- and 32-bit CRCs. Remember that the width of the divisor is always one bit wider than the remainder.
So, for example, you'd use a 17-bit generator polynomial whenever a 16-bit checksum is required.

	CRC - CCITT	**CRC – 16**	**CRC - 32**
Checksum Width	16 bits	16 bits	32 bits
Generator Polynomial	10001000000100001	11000000000000101	100000100110000010001110110111011

International Standard CRC Polynomials

Program:

```
#include<stdio.h>
#include<string.h>

char data[100],concatdata[117],src_crc[17],dest_crc[17],frame[120],divident[18];
char divisor[18];
char res[17]="0000000000000000";

void crc_cal(int node)
{
int i,j;
for(j=17;j<=strlen(concatdata);j++)
{
```

```
        if(divident[0]=='1')
        {
        for(i=1;i<=16;i++)
        if(divident[i]!=divisor[i])
        divident[i-1]='1';
        else
        divident[i-1]='0';
        }

        else
        {
        for(i=1;i<=16;i++)
        divident[i-1]=divident[i];
        }
        if(node==0)
        divident[i-1]=concatdata[j];
        else
        divident[i-1]=frame[j];
}
divident[i]='\0';
printf("\ncrc is %s\n",divident);
if(node==0)
{
strcpy(src_crc,divident);
}
else
strcpy(dest_crc,divident);
}
int main()
{
int i;
printf("enter the generator bits\n");
gets(divisor);
if(strlen(divisor)<17 || strlen(divisor)>17)
{
printf("please enter the geneartor length min of 17 bits\n");
exit(0);
}
printf("\n At src node :\n Enter the msg to be sent :");
gets(data);
strcpy(concatdata,data);
strcat(concatdata,"0000000000000000");
for(i=0;i<=16;i++)
divident[i]=concatdata[i];
divident[i]='\0';
crc_cal(0);
printf("\ndata is:\t");
```

```
puts(data);
printf("\n The frame transmitted is :\t");
printf("\n%s%s",data,src_crc);
printf("\n\t\tSOURCE NODE TRANSMITTED THE FRAME---->");
printf("\n\n\n\n\t\t\tAT DESTINATION NODE\nenter the recived frame:\t");
gets(frame);
for(i=0;i<=16;i++)
divident[i]=frame[i];
divident[i]='\0';
crc_cal(1);
if((strcmp(dest_crc,res))==0)
printf("\nRecived frame is error free .\n ");
else
printf("\nRecived frame containes one or more error ");
return 1;
}
```

Output:

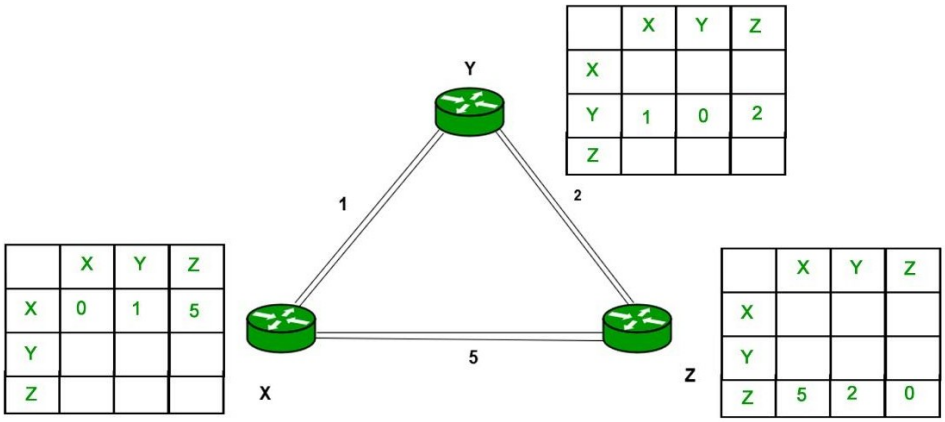
```
enter the generator bits
10001000000100001

 At src node :
 Enter the msg to be sent :0101

crc is 0101000010100101

data is:        0101

 The frame transmitted is :
0101010100001010010l
                SOURCE NODE TRANSMITTED THE FRAME----->



                        AT DESTINATION NODE
enter the recived frame:        0101010100001010010l

crc is 0000000000000000

Recived frame is error free .


...Program finished with exit code 1
Press ENTER to exit console.
```

```
enter the generator bits
10001000000100001

 At src node :
 Enter the msg to be sent :0101

crc is 0101000010100101

data is:        0101

 The frame transmitted is :
0101010100001010010l
                SOURCE NODE TRANSMITTED THE FRAME---->



                        AT DESTINATION NODE
enter the recived frame:        0101010100001010010O

crc is 0000000000000001

Recived frame containes one or more error

...Program finished with exit code 1
Press ENTER to exit console.
```

Program 2

Write a program to find the shortest path between vertices using Distance vector algorithm.

Consider the below network:



Algorithm:

At each node x,

Initialization

for all destinations y in N:

$D_x(y) = c(x,y)$ // If y is not a neighbor then $c(x,y) = \infty$

for each neighbor w

$D_w(y) = ?$ for all destination y in N.for each neighbor w

send distance vector $D_x = [D_x(y) : y \text{ in } N]$ to w

loop

wait(until I receive any distance vector from some neighbor w)for each y in N:

$D_x(y) = \min_v \{c(x,v)+D_v(y)\}$

If $D_x(y)$ is changed for any destination y

Send distance vector $D_x = [D_x(y) : y \text{ in } N]$ to all neighbors

Program

```
#include<stdio.h>

struct rtable
{
int dist[20],nextnode[20];
}table[20];
 int cost[10][10],n;
 void distvector()
{
int i,j,k,count=0;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
table[i].dist[j]=cost[i][j];
table[i].nextnode[j]=j;
}       }
do
{
count=0;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
for(k=0;k<n;k++)
{
if(table[i].dist[j]>cost[i][k]+table[k].dist[j])
{
table[i].dist[j]=table[i].dist[k]+table[k].dist[j];
table[i].nextnode[j]=k;
count++;
}
}
}
}
}while(count!=0);
}
int main()
{
int i,j;
printf("\nenter the no of vertices:\t");
scanf("%d",&n);
```

```
printf("\nenter the cost matrix\n");
for(i=0;i<n;i++)
for(j=0;j<n;j++)
scanf("%d",&cost[i][j]);
distvector();
for(i=0;i<n;i++)
{
printf("\nstate value for router %c \n",i+65);
printf("\ndestnode\tnextnode\tdistance\n");
for(j=0;j<n;j++)
{
if(table[i].dist[j]==99)
printf("%c\t\t-\t\t infinite\n",j+65);
else
printf("%c\t\t%c\t\t%d\n",j+65,table[i].nextnode[j]+65,table[i].dist[j]);
}
}
return 0;

 }
```

Output:

```
enter the no of vertices:        3

enter the cost matrix
0 1 5
1 0 2
5 2 0

state value for router A

destnode          nextnode          distance
A                 A                 0
B                 B                 1
C                 B                 3

state value for router B

destnode          nextnode          distance
A                 A                 1
B                 B                 0
C                 C                 2

state value for router C

destnode          nextnode          distance
A                 B                 3
B                 B                 2
C                 C                 0


...Program finished with exit code 0
Press ENTER to exit console.
```

Program 3

Using TCP sockets, write a client – server program to make the interaction between the client and theserver.

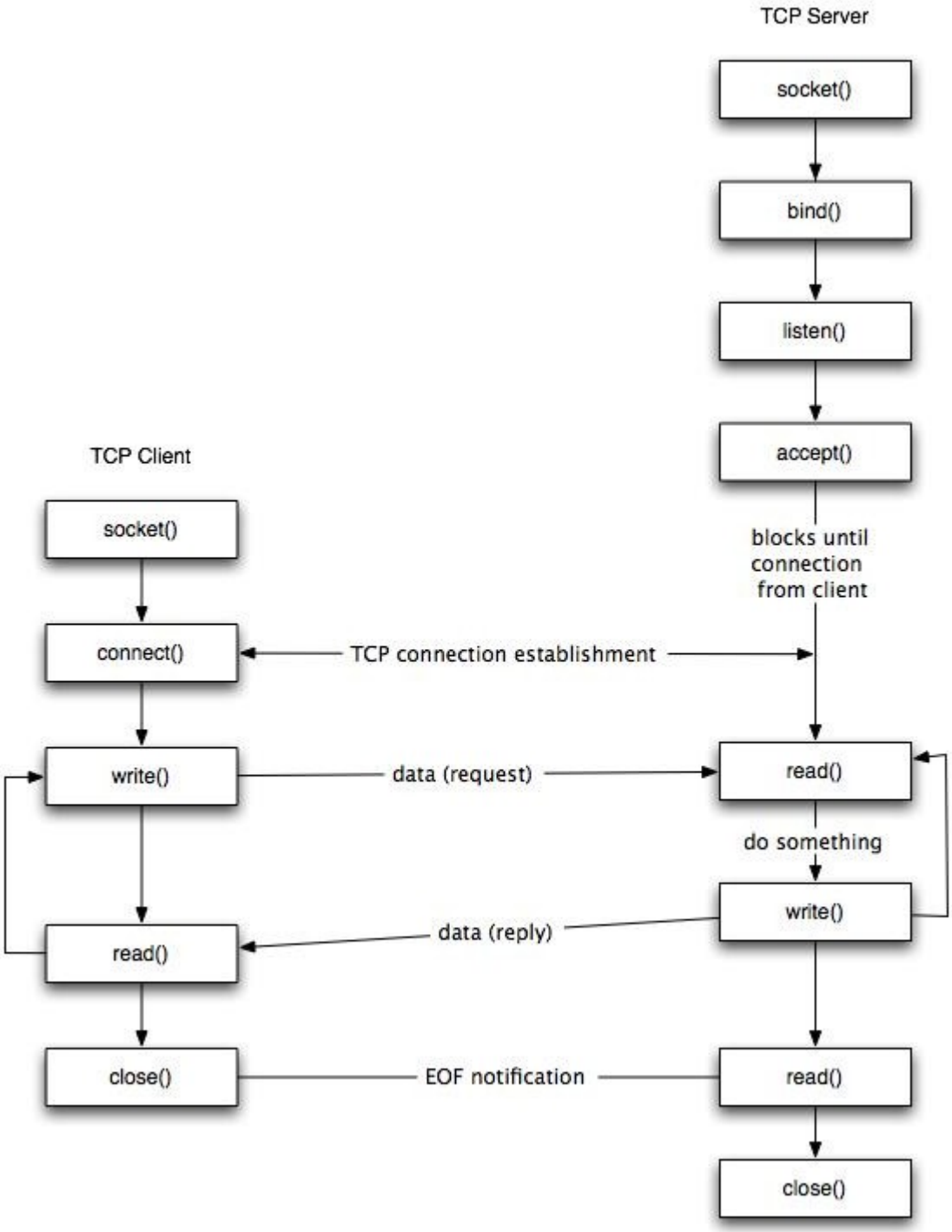


Fig: TCP Socket lifecycle diagram.

Stages for server

1. **Socket creation:**
   int sockfd = socket(domain, type, protocol)

- **sockfd:** socket descriptor, an integer (like a file-handle)
- **domain:** integer, specifies communication domain. We use AF_ LOCAL as defined in the POSIX standard for communication between processes on the same host. For communicating between processes on different hosts connected by IPV4, we use AF_INET and AF_I NET 6for processes connected by IPV6.
- **type:** communication type
  SOCK_STREAM: TCP(reliable, connection oriented) SOCK_DGRAM: UDP(unreliable, connectionless)

- **protocol:** Protocol value for Internet Protocol(IP), which is 0. This is the same number which appears on protocol field in the IP header of a packet.(man protocols for more details)

2. Setsockopt: This helps in manipulating options for the socket referred by the file descriptor sockfd. This is completely optional, but it helps in reuse of address and port. Prevents error such as: "address already in use".
*int setsockopt(int sockfd, int level, int optname, const void *optval, socklen_t optlen);*

3. **Bind:**
   *int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);*

   After creation of the socket, bind function binds the socket to the address and port number specified in addr(custom data structure). In the example code, we bind the server to the localhost, hence we use INADDR_ANY to specify the IP address.

4. **Listen:**
   int listen(int sockfd, int backlog);

   It puts the server socket in a passive mode, where it waits for the client to approach the server to make a connection. The backlog, defines the maximum length to which the queue of pending connections for sockfd may grow. If a connection request arrives when the queue is full, the client may receive an error with an indication of ECONNREFUSED.

5. **Accept:**
   *int new_socket= accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);*

   It extracts the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket. At this point, connection is established between client and server, and they are ready to transfer data.

Code for TCP socket Server

Program No: 3(a) Server

```
#include<stdio.h>
#include<sys/types.h>
```

```
#include<sys/socket.h>
#include<netinet/in.h>
#include<sys/fcntl.h>
#include<stdlib.h>
int main(int argc,char *argv[])
{
int fd,sockfd,newsockfd,clilen,portno,n;
struct sockaddr_in seradd,cliadd;
char buffer[4096];
if(argc<2)
{
fprintf(stderr,"\n\n No port\n");
exit(1);
}
portno=atoi(argv[1]);
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd<0)
error("\n error opening socket.\n");
bzero((char *)&seradd,sizeof(seradd));
seradd.sin_family=AF_INET;
seradd.sin_addr.s_addr=(htonl)INADDR_ANY;
seradd.sin_port=htons(portno);
if(bind(sockfd,(struct sockaddr *)&seradd,sizeof(seradd))<0)
perror("\n IP addr cannt bind");
listen(sockfd,5);
clilen=sizeof(cliadd);
printf("\n Server waiting for clint request\n");
while(1)
{
newsockfd=accept(sockfd,(struct sockaddr *)&cliadd,&clilen);
if(newsockfd<0)
perror("\n Server cannot accept connection request ");
bzero(buffer,4096);
read(newsockfd,buffer,4096);
fd=open(buffer,O_RDONLY);
if(fd<0)
perror("\n File  doesnot exist");
while(1)
{
n=read(fd,buffer,4096);
if(n<=0)
exit(0);
write(newsockfd,buffer,n);
printf("\n File transfer completet\n");
}
close(fd);
close(newsockfd);
}
return 0;}
```

Stages for Client

- **Socket connection:** Exactly same as that of server's socket creation
- **Connect:** The connect() system call connects the socket referred to by the file
  Descriptor sockfd to the address specified by addr. Server's address and port is specified in
  addr.

Int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);

Program No: 3(b) Client

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<sys/fcntl.h>
#include<stdlib.h>
#include<string.h>
#include<arpa/inet.h>

int main(int argc,char *argv[])
{
int sockfd,portno,n;
struct sockaddr_in seradd;
char buffer[4096],*serip;
if(argc<4)
{
fprintf(stderr,"usage %s serverip filename port",argv[0]);
exit(0);
}
serip=argv[1];
portno=atoi(argv[3]);
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd<0)
perror("\n Error in creating socket.\n");
perror("\n Client on line.");
bzero((char *)&seradd,sizeof(seradd));
seradd.sin_family=AF_INET;
seradd.sin_addr.s_addr=inet_addr(serip);
seradd.sin_port=htons(portno);
if(connect(sockfd,(struct sockaddr *)&seradd,sizeof(seradd))<0)
perror("\n Error in connection setup \n");
write(sockfd,argv[2],strlen(argv[2])+1);
bzero(buffer,4096);
n=read(sockfd,buffer,4096);
if(n<=0)
```

```
{
perror("\n File not found");
exit(0);
}
write (1,buffer,n);
}
```

Output:

cc client.c

./a.out

Client:Hello message sentHello from server

cc server.c

./a.out

Server:Hello from clientHello message sent

Program 4

Implement the above program using as message queues or FIFOs as IPC channels.

Algorithm (Client Side)

Start.

Open well known server FIFO in write mode.

Write the pathname of the file in this FIFO and send the request.

Open the client specified FIFO in read mode and wait for reply.

When the contents of the file are available in FIFO, display it on the terminal

Stop.

Algorithm (Server Side)

Start.

Create a well known FIFO using mkfifo()

Open FIFO in read only mode to accept request from clients.

When client opens the other end of FIFO in write only mode, then read the contents and store it in buffer.

Create another FIFO in write mode to send replies.

Open the requested file by the client and write the contents into the client specified FIFO and terminate the connection.

Stop.

Program No: 4(a) Server

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
#define FIFO1 "fifo1"
#define FIFO2 "fifo2"

int main()
{
```

```
char p[100],c[5000,ch;
int num,fd,fd2,f1;
mknod(FIFO1,S_IFIFO|0666,0);
mknod(FIFO2,S_IFIFO|0666,0);
printf("\n Server online...\n");
fd=open(FIFO1,O_RDONLY);
fd2=open(FIFO2,O_WRONLY);
printf("Server online\n waiting for client \n\n");
if((num=read(fd,p,100))==-1)
perror("\n Read Error ");
else
{
p[num]='\0';
printf("\n File is %s .\n",p);
if((f1=open(p,O_RDONLY))<0)
{
write(fd2,"File not found",15);
return 1;
}
else
{
stdin=fdopen(f1,"r");
num=0;
while((ch=fgetc(stdin))!=EOF)
c[num++]=ch;
c[num]=0;
printf(" Server: Transfering the contents of :%s ",p);
if(num=write(fd2,c,strlen(c))==-1)
printf("\n Error in writting to FIFO\n");
else
printf("\n File transfer completed \n");
}}}
```

Program No: 4(b) Client

```
#include<stdio.h>

#include<stdlib.h>
#include<string.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
#define FIFO1 "fifo1"
#define FIFO2 "fifo2"

int main()
{
char p[100],c[5000];
```

```
int num,fd,fd2,f1;
mknod(FIFO1,S_IFIFO|0666,0);
mknod(FIFO2,S_IFIFO|0666,0);
printf("\n Client online...\n");
fd=open(FIFO1,O_WRONLY);
fd2=open(FIFO2,O_RDONLY);
printf("Client : Enter the filename . \n\n");
scanf("%s",p);
num=write(fd,p,strlen(p));
if(num==-1)
{
perror("\nWrite Error.\n");
return 1;
}
else
{
printf("\n Waiting for reply\n");
if((num=read(fd2,c,5000))==-1)
perror("\nError while reading from fifo \n");
else
{
c[num]=0;
printf("%s",c);
}}
return 1;
}
```

Output:

SERVER OUTPUT

[root@localhost ~]# vi fifo_server.c

[root@localhost ~]# cc -o fifo_server fifo_server.c

[root@localhost ~]# ./fifo_serverWaiting for connection Request...Connection Established...Client has requested file dvr.c

[root@localhost ~]#

CLIENTOUTPUT

[root@localhost ~]# vi fifo_client.c

[root@localhost ~]# cc -o fifo_client fifo_client.c

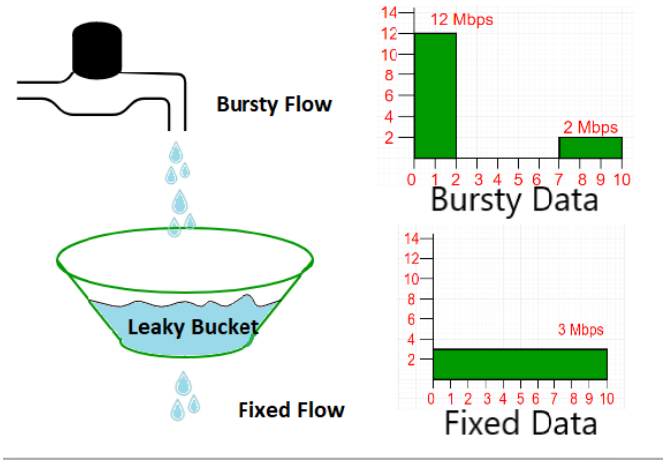
[root@localhost ~]# ./fifo_clientTrying to Connect to Server...Connected...

Enter the filename to request from server: dvr.c

Waiting for Server to reply...

Program 5

Write a program to implement the concept traffic flow controlling using leaky buckets.



Suppose we have a bucket in which we are pouring water, at random points in time, but we have to get water at a fixed rate, to achieve this we will make a hole at the bottom of the bucket. This will ensure that the water coming out is at some fixed rate, and also if the bucket gets full, then we will stop pouring water into it. The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate.

In the above figure, we assume that the network has committed a bandwidth of 3 Mbps for a host. The use of the leaky bucket shapes the input traffic to make it conform to this commitment. In the above figure, the host sends a burst of data at a rate of 12 Mbps for 2s, for a total of 24 Mbits of data. The host is silent for 5 s and then sends data at a rate of 2 Mbps for 3 s, for a total of 6 Mbits of data. In all, the host has sent 30 Mbits of data in 10 s. The leaky bucket smooths out the traffic by sending out data at a rate of 3 Mbps during the same 10 s.

Without the leaky bucket, the beginning burst may have hurt the network by consuming more bandwidth than is set aside for this host. We can also see that the leaky bucket may prevent congestion. A simple leaky bucket algorithm can be implemented using FIFO queue. A FIFO queue holds the packets. If the traffic consists of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock. If the

traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits.

C Program

```
#include <stdio.h>
#define BUCKET_SIZE 10  // Size of the bucket (maximum capacity)
#define OUTPUT_RATE 4   // Rate at which packets are sent out

int main() {
   int n, i;
   int bucket_content = 0;
   // Collect the number of packets from the user
   printf("Enter the number of packets: ");
   scanf("%d", &n);
   int input_packets[n];
   // Collect the size of each packet from the user
   for (i = 0; i < n; i++) {
      printf("Enter the size of packet %d: ", i + 1);
      scanf("%d", &input_packets[i]);
   }

   // Process each packet
   for (i = 0; i < n; i++) {
      printf("\nPacket %d with size %d arrived.\n", i+1, input_packets[i]);
```

```
        // If incoming packet can be accommodated in the bucket
        if (input_packets[i] <= (BUCKET_SIZE - bucket_content)) {
            bucket_content += input_packets[i];
            printf("Bucket content: %d out of %d\n", bucket_content, BUCKET_SIZE);
        } else {
            printf("Bucket overflow! Packet %d of size %d is discarded.\n", i+1, input_packets[i]);
        }
        // Leaking the packets at the output rate
        if (bucket_content > OUTPUT_RATE) {
            bucket_content -= OUTPUT_RATE;
        } else {
            bucket_content = 0;
        }
        printf("After sending packets, bucket content: %d out of %d\n", bucket_content, BUCKET_SIZE);
    }
    // Handle remaining packets in the bucket after all inputs are processed
    while (bucket_content > 0) {
        printf("\nSending remaining packets...\n");
        if (bucket_content > OUTPUT_RATE) {
            bucket_content -= OUTPUT_RATE;
        } else {
            bucket_content = 0;
        }
```

```
        printf("Bucket content: %d out of %d\n", bucket_content, BUCKET_SIZE);
    }
    return 0;
}
```

Output:

```
/tmp/GLaipJY6Pc.o
Enter the number of packets: 4
Enter the size of packet 1: 5
Enter the size of packet 2: 3
Enter the size of packet 3: 6
Enter the size of packet 4: 6

Packet 1 with size 5 arrived.
Bucket content: 5 out of 10
After sending packets, bucket content: 1 out of 10

Packet 2 with size 3 arrived.
Bucket content: 4 out of 10
After sending packets, bucket content: 0 out of 10

Packet 3 with size 6 arrived.
Bucket content: 6 out of 10
After sending packets, bucket content: 2 out of 10

Packet 4 with size 6 arrived.
Bucket content: 8 out of 10
After sending packets, bucket content: 4 out of 10

Sending remaining packets...
Bucket content: 0 out of 10
```

Output – 2

```
Enter the number of packets: 5
Enter the size of packet 1: 11
Enter the size of packet 2: 4
Enter the size of packet 3: 8
Enter the size of packet 4: 12
Enter the size of packet 5: 3

Packet 1 with size 11 arrived.
Bucket overflow! Packet 1 of size 11 is discarded.
After sending packets, bucket content: 0 out of 10

Packet 2 with size 4 arrived.
Bucket content: 4 out of 10
After sending packets, bucket content: 0 out of 10

Packet 3 with size 8 arrived.
Bucket content: 8 out of 10
After sending packets, bucket content: 4 out of 10

Packet 4 with size 12 arrived.
Bucket overflow! Packet 4 of size 12 is discarded.
After sending packets, bucket content: 0 out of 10

Packet 5 with size 3 arrived.
Bucket content: 3 out of 10
After sending packets, bucket content: 0 out of 10
```

Program 6

Write a program to illustrate the optimization technique to achieve shortest path routing using travelling sales man approach

The Traveling Salesman Problem (TSP) is a classic optimization problem where the goal is to find the shortest possible route that visits a set of cities exactly once and returns to the original city. It's a well-known NP-hard problem in combinatorial optimization.

TSP Problem Description:

Given a set of cities and the distances between every pair of cities, the task is to find the shortest possible tour that visits each city exactly once and returns to the starting city.

Brute Force Approach:

One way to solve TSP is to use the brute force approach, which involves generating all possible permutations of the cities and calculating the total distance for each permutation. The smallest distance found is the optimal solution.

```
#include <stdio.h>

#include <limits.h>

#define MAX 10  // Maximum number of cities

// Function to find the minimum cost of the TSP and the path taken

int tsp(int graph[][MAX], int pos, int visited, int cost, int start, int V, int path[], int *pathIndex)
{
   if (visited == (1 << V) - 1) {  // If all cities have been visited

      path[*pathIndex] = start;  // Add starting city to complete the cycle

      (*pathIndex)++;

      return cost + graph[pos][start];  // Return to the starting city

   }

   int minCost = INT_MAX;
```

```
    int bestCity = -1;
    for (int city = 0; city < V; city++) {
        if ((visited & (1 << city)) == 0) {  // If the city has not been visited yet
            int newVisited = visited | (1 << city);
            int newCost = cost + graph[pos][city];
            int tspCost = tsp(graph, city, newVisited, newCost, start, V, path, pathIndex);
            if (tspCost < minCost) {
                minCost = tspCost;
                bestCity = city;
            }
        }
    }
    // Add the best city to the path
    path[*pathIndex] = bestCity;
    (*pathIndex)++;
    return minCost;
}

int main() {
    int V;
    // Input the number of cities
    printf("Enter the number of cities: ");
    scanf("%d", &V);
    if (V > MAX) {
```

```
    printf("Number of cities exceeds the maximum limit of %d\n", MAX);
    return 1;
}
int graph[MAX][MAX];
// Input the distances between cities
printf("Enter the distances between the cities in matrix form:\n");
for (int i = 0; i < V; i++) {
    for (int j = 0; j < V; j++) {
        printf("Distance from city %d to city %d: ", i + 1, j + 1);
        scanf("%d", &graph[i][j]);
    }
}
int start = 0;  // Starting city
int visited = 1 << start;  // Mark the starting city as visited
int path[MAX + 1];  // To store the path (extra space for returning to the start)
int pathIndex = 0;  // Path index
path[pathIndex++] = start;  // Start from the first city
int result = tsp(graph, start, visited, 0, start, V, path, &pathIndex);
printf("The minimum cost of the TSP is: %d\n", result);
printf("The path taken is: ");
for (int i = 0; i < pathIndex; i++) {
    printf("%d ", path[i] + 1);  // Print the path (adjusting for 1-based indexing)
}
printf("\n");
```

```
    return 0;
}
```

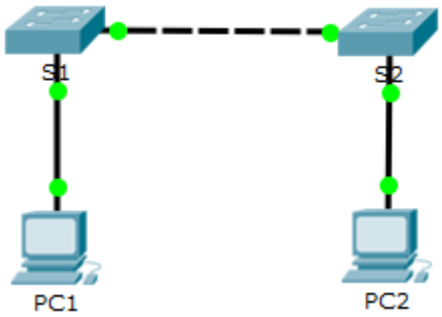
PART B

Network Simulation

Program 1:

Implementing Basic Connectivity of a computer network

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask
S1	VLAN 1	192.168.1.253	255.255.255.0
S2	VLAN 1	192.168.1.254	255.255.255.0
PC1	NIC	192.168.1.1	255.255.255.0
PC2	NIC	192.168.1.2	255.255.255.0

Objectives

Part 1: Perform a Basic Configuration on S1 and S2

Part 2: Configure the PCs

Part 3: Configure the Switch Management Interface

Background

In this activity, you will first perform basic switch configurations. Then, you will implement basic connectivity by configuring IP addressing on switches and PCs. When the IP addressing configuration is complete, you will use various **show** commands to verify configurations and use the **ping** command to verify basic connectivity between devices.

Part 1: Perform a Basic Configuration on S1 and S2

Complete the following steps on S1 and S2.

Step 1: Configure S1 with a hostname.

a. Click S1 and then click the **CLI** tab.

b. Enter the correct command to configure the hostname as **S1**.

Step 2: Configure the console and privileged EXEC mode passwords.

a. Use **cisco** for the console password.

b. Use **class** for the privileged EXEC mode password.

Step 3: Verify the password configurations for S1.

How can you verify that both passwords were configured correctly?

After you exit user EXEC mode, the switch will prompt you for a password to access the console interface and will prompt you a second time when accessing the privileged EXEC mode. You can also use the **show run** command to view the passwords.

Step 4: Configure an MOTD banner.

Use an appropriate banner text to warn unauthorized access. The following text is an example:

Authorized access only. Violators will be prosecuted to the full extent of the law.

Step 5: Save the configuration file to NVRAM.

Which command do you issue to accomplish this step?

S1(config)# **exit** (or end)

S1# **copy run start**

Step 6: Repeat Steps 1 to 5 for S2.

Part 2: Configure the PCs

Configure PC1 and PC2 with IP addresses.

Step 1: Configure both PCs with IP addresses.

a. Click PC1 and then click the **Desktop** tab.

b. Click **IP Configuration**. In the Addressing Table above, you can see that the IP address for PC1 is 192.168.1.1 and the subnet mask is 255.255.255.0. Enter this information for PC1 in the **IP Configuration** window.

c. Repeat steps 1a and 1b for PC2.

Step 2: Test connectivity to switches.

a. Click PC1. Close the **IP Configuration** window if it is still open. In the **Desktop** tab, click **Command Prompt**.

b. Type the **ping** command and the IP address for S1 and press Enter.

Packet Tracer PC Command Line 1.0

PC> **ping 192.168.1.253**

Were you successful? Explain.

Your ping should have been unsuccessful because the switches have not been configured with an IP address.

Part 3: Configure the Switch Management Interface

Configure S1 and S2 with an IP address.

Step 1: Configure S1 with an IP address.

Switches can be used as plug-and-play devices. This means that they do not need to be configured for them to work.

Switches forward information from one port to another based on MAC addresses. If this is the case, why would we configure it with an IP address?

In order for you to connect remotely to a switch, you need to assign it an IP address. The default configuration on the switch is to have the management of the switch controlled through VLAN 1.

Use the following commands to configure S1 with an IP address.

```
S1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
S1(config)# interface vlan 1
S1(config-if)# ip address 192.168.1.253 255.255.255.0
S1(config-if)# no shutdown
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
S1(config-if)#
S1(config-if)# exit
S1#
```

Why do you enter the **no shutdown** command?

The **no shutdown** command administratively places the interface in an active state.

Step 2: Configure S2 with an IP address.

Use the information in the Addressing Table to configure S2 with an IP address.

Step 3: Verify the IP address configuration on S1 and S2.

Use the **show ip interface brief** command to display the IP address and status of all the switch ports and interfaces. You can also use the **show running-config** command.

Step 4: Save configurations for S1 and S2 to NVRAM.

Which command is used to save the configuration file in RAM to NVRAM? **copy running-config startup-config**

Step 5: Verify network connectivity.

Network connectivity can be verified using the **ping** command. It is very important that connectivity exists throughout the network. Corrective action must be taken if there is a failure. Ping S1 and S2 from PC1 and PC2.

a. Click PC1 and then click the **Desktop** tab.

b. Click **Command Prompt**.

c. Ping the IP address for PC2.

d. Ping the IP address for S1.

e. Ping the IP address for S2.

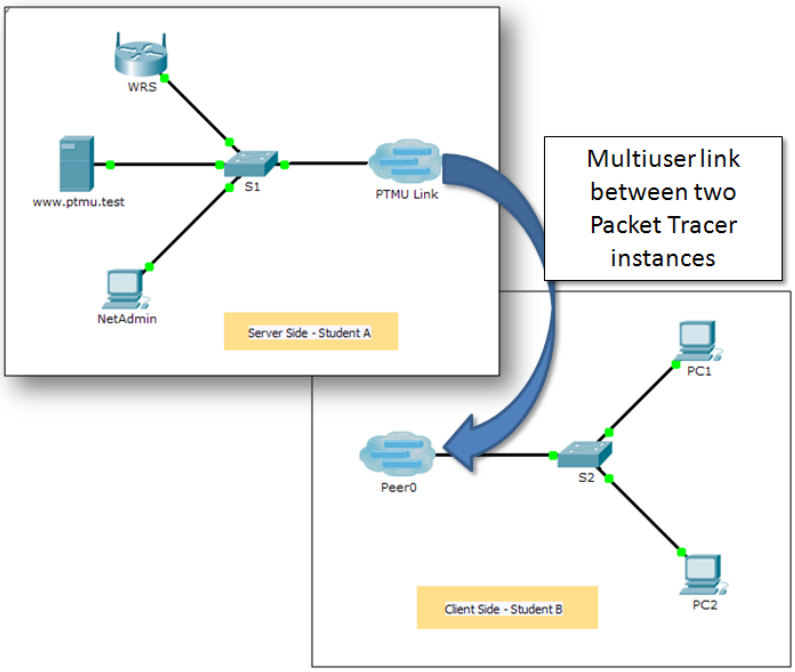
Note: You can also use the **ping** command on the switch CLI and on PC2.

All pings should be successful. If your first ping result is 80%, try again. It should now be 100%. You will learn why a ping may sometimes fail the first time later in your studies. If you are unable to ping any of the devices, recheck your configuration for errors.

Program 2:

Packet Tracer Multiuser - Implement Services

Topology



Addressing Table

Device	IP Address	Subnet Mask
Server Side Player		
WRS	172.16.1.254	255.255.255.0
S1	172.16.1.1	255.255.255.0
www.ptmu.test	172.16.1.5	255.255.255.0
NetAdmin	DHCP Assigned	DHCP Assigned
Client Side Player		
S2	172.16.1.2	255.255.255.0
PC1	DHCP Assigned	DHCP Assigned
PC2	DHCP Assigned	DHCP Assigned

Objectives

Part 1: Establish a Local Multiuser Connection to another Instance of Packet Tracer

Part 2: Server Side Player - Implement and Verify Services

Part 3: Client Side Player - Configure and Verify Access to Services

Background

Note: Completing the prior activities in this chapter, including the **Packet Tracer Multiuser - Tutorial**, are prerequisites to completing this activity.

In this multiuser activity, two students (players) cooperate to implement and verify services including DHCP, HTTP, Email, DNS, and FTP. The server side player will implement and verify services on one server. The client side player will configure two clients and verify access to services.

Part 4: Establish a Local Multiuser Connection to Another Instance of Packet Tracer

Step 1: Select a partner and determine the role for each student.

a. Find a fellow classmate with whom you will cooperate to complete this activity. Your computers must both be connected to the same LAN.

b. Determine which of you will play the server side and which of you will play the client side in this activity.

- The server side player opens **Packet Tracer Multiuser - Implement Services - Server Side.pka**.
- The client side player opens **Packet Tracer Multiuser - Implement Services - Client Side.pka**.

Note: Solo players can open both files and complete the steps for both sides.

Step 2: Configure the switches with initial configurations.

Each player: configure your respective switch with the following:

a. Hostname using the name in the addressing tables. (**S1** for the switch in the Server Side Player or **S2** for the switch in the Client Side Player). Change the Display Name of each switch to match the new hostname using the **Config** tab.

b. An appropriate message-of-the-day (MOTD) banner.

c. Privileged EXEC mode and line passwords.

d. Correct IP addressing, according to the Addressing Table.

e. Scoring should be 8/33 for the client side player and 8/44 for the server side player.

Step 3: Server Side Player - Configure the PTMU link and communicate addressing.

a. Complete the steps necessary to verify that the **PTMU Link** is ready to receive an incoming connection.

b. Communicate the necessary configuration information to the client side player.

Step 4: Client Side Player - Configure the outgoing multiuser connection.

a. Client side player: Record the following information supplied to you by the server side player:

IP Address: _____

Port Number: _____

Password (**cisco**, by default) _____

b. Configure **Peer0** to connect to the server side player's **PTMU Link**.

c. Connect the **S2 GigabitEthernet0/1** to **Link0** on **Peer0**.

Step 5: Verify connectivity across the local multiuser connection.

a. The server side player should be able to ping S2 in the client side player's instance of Packet Tracer.

b. The client side player should be able to ping S1 in the server side player's instance of Packet Tracer.

c. Scoring should be 11/33 for the client side player and 9/44 for the server side player.

Part 5: Server Side Player - Implement and Verify Services

Step 1: Configure WRS as the DHCP server.

WRS provides DHCP services. Configure DHCP Server Settings with the following:

a. Starting IP address is **172.16.1.11**.

b. Maximum number of users is **100**.

c. **Static DNS 1** is **172.16.1.5**.

d. Verify **NetAdmin** received IP addressing through DHCP.

e. From **NetAdmin**, access the User Account Information web page at **172.16.1.5**. You will use this information to configure user accounts in Step 2.

f. Scoring should be 17/44 for the server side player.

Step 2: Configure services on www.ptmu.test.

The **www.ptmu.test** server provides the rest of the services and should be configured with the following:

a. Enable the DNS service and create a DNS record associating the IP address for **www.ptmu.test** server to the name www.ptmu.test.

b. Enable the Email services and create user accounts using the user list from Part 2 Step 1e. The Domain Name is **ptmu.test**.

c. Enable the FTP service and create user accounts using the user list from Part 2 Step 1e. Give each user permission to write, read, and list.

d. Scoring should be 38/44 for the server side player.

Step 3: Verify that all services are implemented according to the requirements.

From **NetAdmin**, complete the following:

a. Configure the email client for the NetAdmin user account. (Hint: Use www.ptmu.test for both the incoming and outgoing mail server.)

b. Send an email to the user at **PC1**.

c. Upload the **secret.txt** file to the FTP server. Do not change the file.

Note: The score for the server side player will be **43/44** until the client side player successfully downloads the **secret.txt** file, modifies the file, and then uploads it to the **www.ptmu.test** FTP server.

Part 6: Client Side Player - Configure and Verify Access to Services

Step 1: Configure and verify PC addressing.

a. Configure **PC1** and **PC2** to automatically obtain addressing.

b. PC1 and PC2 should be able to access the web page using the IP address, **http://172.16.1.5**, as well as the domain name, **http://www.ptmu.test**.

c. The score for the client side player should be 21/33.

Step 2: Configure and verify PC email accounts.

a. Configure email accounts according to the requirements at **www.ptmu.test/user.html**.

b. Verify that PC1 received an email from NetAdmin and send a reply.

c. Send an email from PC1 to PC2. **Note:** Scoring will not change.

d. Verify that PC2 received an email from PC1.

e. The score for the client side player should be 31/33.

Step 3: Upload and download a file from the FTP server.

a. From PC2, access the FTP server and download the **secret.txt** file.

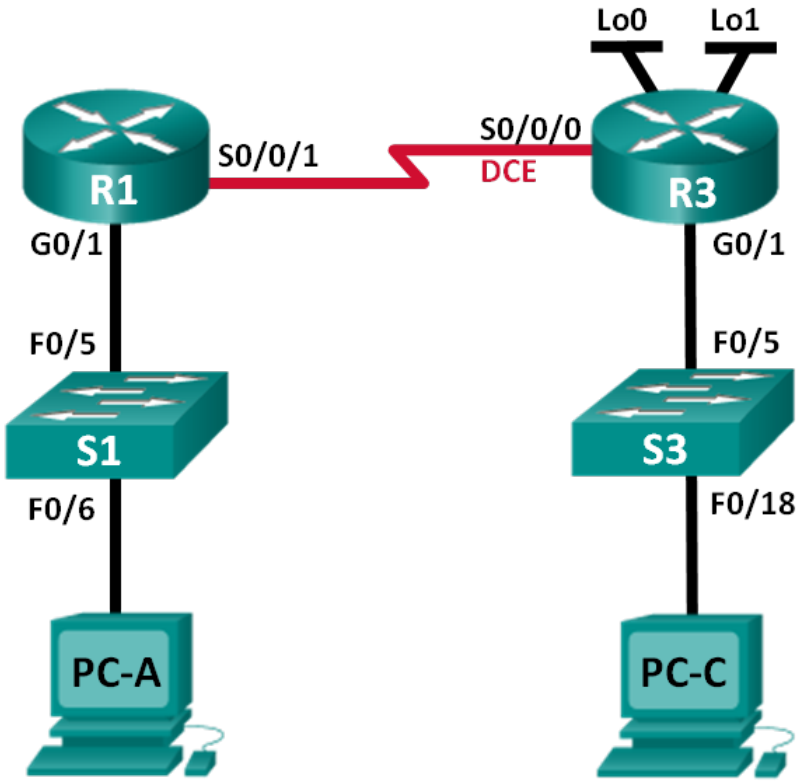
b. Open the **secret.txt** file, change only the secret word to **apple**, and upload the file.

c. The server side player score should be **44/44** and the client side player score should be **33/33**.

Program 3

Configuring IPv4 Static and Default Routes

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.0.1	255.255.255.0	N/A
	S0/0/1	10.1.1.1	255.255.255.252	N/A
R3	G0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/0 (DCE)	10.1.1.2	255.255.255.252	N/A
	Lo0	209.165.200.225	255.255.255.224	N/A
	Lo1	198.133.219.1	255.255.255.0	N/A
PC-A	NIC	192.168.0.10	255.255.255.0	192.168.0.1
PC-C	NIC	192.168.1.10	255.255.255.0	192.168.1.1

Objectives

Part 1: Set Up the Topology and Initialize Devices

Part 2: Configure Basic Device Settings and Verify Connectivity

Part 3: Configure Static Routes

- Configure a recursive static route.
- Configure a directly connected static route.
- Configure and remove static routes.

Part 4: Configure and Verify a Default Route

Background / Scenario

A router uses a routing table to determine where to send packets. The routing table contains a set of routes that describe which gateway or interface the router uses to reach a specified network. Initially, the routing table contains only directly connected networks. To communicate with distant networks, routes must be specified and added to the routing table.

In this lab, you will manually configure a static route to a specified distant network based on a next-hop IP address or exit interface. You will also configure a static default route. A default route is a type of static route that specifies a gateway to use when the routing table does not contain a path for the destination network.

Note: This lab provides minimal assistance with the actual commands necessary to configure static routing. However, the required commands are provided in Appendix A. Test your knowledge by trying to configure the devices without referring to the appendix.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

Note: Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 2 Routers (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 2 Switches (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)
- 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Part 7: Set Up the Topology and Initialize Devices

Step 1: Cable the network as shown in the topology.

Step 2: Initialize and reload the router and switch.

Part 8: Configure Basic Device Settings and Verify Connectivity

In Part 2, you will configure basic settings, such as the interface IP addresses, device access, and passwords. You will verify LAN connectivity and identify routes listed in the routing tables for R1 and R3.

Step 1: Configure the PC interfaces.

Step 2: Configure basic settings on the routers.

a. Configure device names, as shown in the Topology and Addressing Table.

b. Disable DNS lookup.

c. Assign **class** as the enable password and assign **cisco** as the console and vty password.

d. Save the running configuration to the startup configuration file.

Step 3: Configure IP settings on the routers.

a. Configure the R1 and R3 interfaces with IP addresses according to the Addressing Table.

b. The S0/0/0 connection is the DCE connection and requires the **clock rate** command. The R3 S0/0/0 configuration is displayed below.

Instructor Note: For Cisco 1941 routers, DCE is automatically detected and the clock rate is automatically set to 2000000, and does not need to be configured.

R3(config)# **interface s0/0/0**
R3(config-if)# **ip address 10.1.1.2 255.255.255.252**
R3(config-if)# **clock rate 128000**
R3(config-if)# **no shutdown**

Step 4: Verify connectivity of the LANs.

a. Test connectivity by pinging from each PC to the default gateway that has been configured for that host.

From PC-A, is it possible to ping the default gateway? _____ Yes

From PC-C, is it possible to ping the default gateway? _____ Yes

b. Test connectivity by pinging between the directly connected routers.

From R1, is it possible to ping the S0/0/0 interface of R3? _____ Yes

If the answer is **no** to any of these questions, troubleshoot the configurations and correct the error.

c. Test connectivity between devices that are not directly connected.

From PC-A, is it possible to ping PC-C? _____ No

From PC-A, is it possible to ping Lo0? _____ No

From PC-A, is it possible to ping Lo1? _____ No

Were these pings successful? Why or why not?

_____

_____

No, the router does not contain routes to the distant networks.

Note: It may be necessary to disable the PC firewall to ping between PCs.

Step 5: Gather information.

a. Check the status of the interfaces on R1 with the **show ip interface brief** command.

R1# **show ip interface brief**

Interface	IP-Address	OK?	Method	Status	Protocol
Embedded-Service-Engine0/0	unassigned	YES	unset	administratively down	down
GigabitEthernet0/0	unassigned	YES	unset	administratively down	down
GigabitEthernet0/1	192.168.0.1	YES	manual	up	up

```
Serial0/0/0            unassigned      YES unset  administratively down down
Serial0/0/1            10.1.1.1        YES manual up                    up
```

How many interfaces are activated on R1? _____ Two

b. Check the status of the interfaces on R3.

```
R3# show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
Embedded-Service-Engine0/0 unassigned      YES unset  administratively down down
GigabitEthernet0/0         unassigned      YES unset  administratively down down
GigabitEthernet0/1         192.168.1.1     YES manual up                    up
Serial0/0/0                10.1.1.2        YES manual up                    up
Serial0/0/1                unassigned      YES unset  administratively down down
Loopback0                  209.165.200.225 YES manual up                    up
Loopback1                  198.133.219.1   YES manual up                    up
```

How many interfaces are activated on R3? _____ Four

c. View the routing table information for R1 using the **show ip route** command.

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.1.0/30 is directly connected, Serial0/0/1
L        10.1.1.1/32 is directly connected, Serial0/0/1
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, GigabitEthernet0/1
L        192.168.0.1/32 is directly connected, GigabitEthernet0/1
```

What networks are present in the Addressing Table of this lab, but not in the routing table for R1?

_____

192.168.1.0, 198.133.219.0, 209.165.200.224

d. View the routing table information for R3.

```
R3# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set
```

```
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.1.0/30 is directly connected, Serial0/0/0
L        10.1.1.2/32 is directly connected, Serial0/0/0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/1
L        192.168.1.1/32 is directly connected, GigabitEthernet0/1
      198.133.219.0/24 is variably subnetted, 2 subnets, 2 masks
C        198.133.219.0/24 is directly connected, Loopback1
L        198.133.219.1/32 is directly connected, Loopback1
      209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.200.224/27 is directly connected, Loopback0
L        209.165.200.225/32 is directly connected, Loopback0
```

What networks are present in the Addressing Table in this lab, but not in the routing table for R3?

192.168.0.0

Why are all the networks not in the routing tables for each of the routers?

The routers are not configured with static or dynamic routing; therefore, the routers only know about the directly connected networks.

Part 9: Configure Static Routes

In Part 3, you will employ multiple ways to implement static and default routes, you will confirm that the routes have been added to the routing tables of R1 and R3, and you will verify connectivity based on the introduced routes.

Note: This lab provides minimal assistance with the actual commands necessary to configure static routing. However, the required commands are provided in Appendix A. Test your knowledge by trying to configure the devices without referring to the appendix.

Step 1: Configure a recursive static route.

With a recursive static route, the next-hop IP address is specified. Because only the next-hop IP is specified, the router must perform multiple lookups in the routing table before forwarding packets. To configure recursive static routes, use the following syntax:

Router(config)# **ip route** *network-address subnet-mask ip-address*

a. On the R1 router, configure a static route to the 192.168.1.0 network using the IP address of the Serial 0/0/0 interface of R3 as the next-hop address. Write the command you used in the space provided.

R1(config)# **ip route 192.168.1.0 255.255.255.0 10.1.1.2**

b. View the routing table to verify the new static route entry.

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
```

```
       + - replicated route, % - next hop override

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/30 is directly connected, Serial0/0/1
L       10.1.1.1/32 is directly connected, Serial0/0/1
     192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, GigabitEthernet0/1
L       192.168.0.1/32 is directly connected, GigabitEthernet0/1
S    192.168.1.0/24 [1/0] via 10.1.1.2
```

How is this new route listed in the routing table?

_____

S 192.168.1.0/24 [1/0] via 10.1.1.2

From host PC-A, is it possible to ping the host PC-C? _____ No

These pings should fail. If the recursive static route is correctly configured, the ping arrives at PC-C. PC-C sends a ping reply back to PC-A. However, the ping reply is discarded at R3 because R3 does not have a return route to the 192.168.0.0 network in the routing table.

Step 2: Configure a directly connected static route.

With a directly connected static route, the *exit-interface* parameter is specified, which allows the router to resolve a forwarding decision in one lookup. A directly connected static route is typically used with a point-to-point serial interface. To configure directly connected static routes with an exit interface specified, use the following syntax:

Router(config)# **ip route** *network-address subnet-mask exit-intf*

a. On the R3 router, configure a static route to the 192.168.0.0 network using S0/0/0 as the exit interface. Write the command you used in the space provided.

_____

R3(config)# **ip route 192.168.0.0 255.255.255.0 s0/0/0**

b. View the routing table to verify the new static route entry.

```
R3# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/30 is directly connected, Serial0/0/0
L       10.1.1.2/32 is directly connected, Serial0/0/0
S    192.168.0.0/24 is directly connected, Serial0/0/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/1
L       192.168.1.1/32 is directly connected, GigabitEthernet0/1
     198.133.219.0/24 is variably subnetted, 2 subnets, 2 masks
```

C 198.133.219.0/24 is directly connected, Loopback1
L 198.133.219.1/32 is directly connected, Loopback1
209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C 209.165.200.224/27 is directly connected, Loopback0
L 209.165.200.225/32 is directly connected, Loopback0

How is this new route listed in the routing table?

S 192.168.0.0/24 is directly connected, Serial0/0/0

c. From host PC-A, is it possible to ping the host PC-C? _____ Yes

This ping should be successful.

Note: It may be necessary to disable the PC firewall to ping between PCs.

Step 3: Configure a static route.

a. On the R1 router, configure a static route to the 198.133.219.0 network using one of the static route configuration options from the previous steps. Write the command you used in the space provided.

R1(config)# **ip route 198.133.219.0 255.255.255.0 S0/0/1**

or

R1(config)# **ip route 198.133.219.0 255.255.255.0 10.1.1.2**

b. On the R1 router, configure a static route to the 209.165.200.224 network on R3 using the other static route configuration option from the previous steps. Write the command you used in the space provided.

R1(config)# **ip route 209.165.200.224 255.255.255.224 S0/0/1**

or

R1(config)# **ip route 209.165.200.224 255.255.255.224 10.1.1.2**

c. View the routing table to verify the new static route entry.

Note: The students may have different routing table outputs depending on the type of configured static routes.

R1# **show ip route**

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 10.1.1.0/30 is directly connected, Serial0/0/1
L 10.1.1.1/32 is directly connected, Serial0/0/1
192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.0.0/24 is directly connected, GigabitEthernet0/1
L 192.168.0.1/32 is directly connected, GigabitEthernet0/1
S 192.168.1.0/24 [1/0] via 10.1.1.2
S 198.133.219.0/24 is directly connected, Serial0/0/1

209.165.200.0/27 is subnetted, 1 subnets
S 209.165.200.224 [1/0] via 10.1.1.2

How is this new route listed in the routing table?

_____

S 198.133.219.0/24 is directly connected, Serial0/0/1

or

S 198.133.219.0/24 [1/0] via 10.1.1.2

d. From host PC-A, is it possible to ping the R1 address 198.133.219.1? _____ Yes

This ping should be successful.

Step 4: Remove static routes for loopback addresses.

a. On R1, use the **no** command to remove the static routes for the two loopback addresses from the routing table. Write the commands you used in the space provided.

_____

R1(config)# **no ip route 209.165.200.224 255.255.255.224 10.1.1.2**

R1(config)# **no ip route 198.133.219.0 255.255.255.0 S0/0/1**

Note: A static route can be removed with the **no** command without specifying the exit interface or next-hop ip address as displayed below.

R1(config)# **no ip route 209.165.200.224 255.255.255.224**

R1(config)# **no ip route 198.133.219.0 255.255.255.0**

b. View the routing table to verify the routes have been removed.

R1# **show ip route**

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 10.1.1.0/30 is directly connected, Serial0/0/1
L 10.1.1.1/32 is directly connected, Serial0/0/1
192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.0.0/24 is directly connected, GigabitEthernet0/1
L 192.168.0.1/32 is directly connected, GigabitEthernet0/1
S 192.168.1.0/24 [1/0] via 10.1.1.2

How many network routes are listed in the routing table on R1? _____ Three

Is the Gateway of last resort set? _____ No

Part 10: Configure and Verify a Default Route

In Part 4, you will implement a default route, confirm that the route has been added to the routing table, and verify connectivity based on the introduced route.

A default route identifies the gateway to which the router sends all IP packets for which it does not have a learned or static route. A default static route is a static route with 0.0.0.0 as the destination IP address and subnet mask. This is commonly referred to as a "quad zero" route.

In a default route, either the next-hop IP address or exit interface can be specified. To configure a default static route, use the following syntax:

Router(config)# **ip route 0.0.0.0 0.0.0.0** {*ip-address or exit-intf*}

a. Configure the R1 router with a default route using the exit interface of S0/0/1. Write the command you used in the space provided.

_____

R1(config)# **ip route 0.0.0.0 0.0.0.0 s0/0/1**

b. View the routing table to verify the new static route entry.

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

S*    0.0.0.0/0 is directly connected, Serial0/0/1
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.1.0/30 is directly connected, Serial0/0/1
L        10.1.1.1/32 is directly connected, Serial0/0/1
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, GigabitEthernet0/1
L        192.168.0.1/32 is directly connected, GigabitEthernet0/1
S     192.168.1.0/24 [1/0] via 10.1.1.2
```

How is this new route listed in the routing table?

_____

S* 0.0.0.0/0 is directly connected, Serial0/0/1

What is the Gateway of last resort?

_____

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

c. From host PC-A, is it possible to ping the 209.165.200.225? _____ Yes

d. From host PC-A, is it possible to ping the 198.133.219.1? _____ Yes

These pings should be successful.

Reflection

1. A new network 192.168.3.0/24 is connected to interface G0/0 on R1. What commands could be used to configure a static route to that network from R3?

_____

_____

Answers will vary. ip route 192.168.3.0 255.255.255.0 10.1.1.1, ip route192.168.3.0 255.255.255.0 s0/0/0, or ip route 0.0.0.0 0.0.0.0 s0/0/0.

2. Is there a benefit to configuring a directly connected static route instead of a recursive static route?

_____

_____

Configuring a directly attached static route allows the routing table to resolve the exit interface in a single search instead of in two searches as needed for recursive static routes.

3. Why is it important to configure a default route on a router?

_____

A default route prevents the router from dropping packets to unknown destinations.

Router Interface Summary Table

Router Interface Summary				
Router Model	**Ethernet Interface #1**	**Ethernet Interface #2**	**Serial Interface #1**	**Serial Interface #2**
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Appendix A: Configuration Commands for Parts 2, 3, and 4

The commands listed in Appendix A are for reference only. This Appendix does not include all the specific commands necessary to complete this lab.

Basic Device Settings

Configure IP settings on the router.

R3(config)# **interface s0/0/0**
R3(config-if)# **ip address 10.1.1.2 255.255.255.252**
R3(config-if)# **clock rate 128000**
R3(config-if)# **no shutdown**

Static Route Configurations

Configure a recursive static route.

R1(config)# **ip route 192.168.1.0 255.255.255.0 10.1.1.2**

Configure a directly connected static route.

R3(config)# **ip route 192.168.0.0 255.255.255.0 s0/0/0**

Remove static routes.

R1(config)# **no ip route 209.165.200.224 255.255.255.224 serial0/0/1**

or

R1(config)# **no ip route 209.165.200.224 255.255.255.224 10.1.1.2**

or

R1(config)# **no ip route 209.165.200.224 255.255.255.224**

Default Route Configuration

R1(config)# **ip route 0.0.0.0 0.0.0.0 s0/0/1**

Device Configs - R1 and R3

Router R1 (after Part 4)

```
R1#show run
Building configuration...

Current configuration : 1547 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2
!
no aaa new-model
!
!
!
!
!
!
!
no ip domain lookup
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
!
```

```
!
redundancy
!
!
!
!
!
! interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 ip address 10.1.1.1 255.255.255.252
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 Serial0/0/1
ip route 192.168.1.0 255.255.255.0 10.1.1.2
!
!
!
!
control-plane
!
!
banner motd ^CUnauthorized access prohibited!^C
!
line con 0
 password 7 01100F175804
 logging synchronous
 login
line aux 0
```

```
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 password 7 01100F175804
 logging synchronous
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Router R3

```
R3#show run
Building configuration...

Current configuration : 1700 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
!
enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2
!
no aaa new-model
memory-size iomem 15
!
!
!
!
!
!
!
no ip domain lookup
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
```

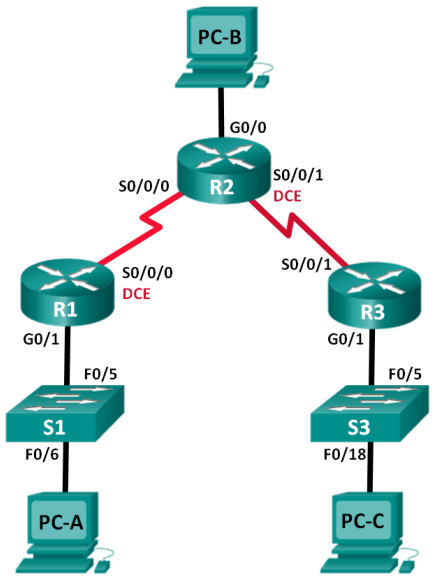
```
!
!
!
!
vtp domain TSHOOT
vtp mode transparent
!
redundancy
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 209.165.200.225 255.255.255.224
!
interface Loopback1
 ip address 198.133.219.1 255.255.255.0
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
duplex auto
 speed auto
!
interface Serial0/0/0
 ip address 10.1.1.2 255.255.255.252
 clock rate 256000
!
interface Serial0/0/1
 no ip address
 shutdown
!
ip forward-protocol nd
```

```
!
no ip http server
no ip http secure-server
!
ip route 192.168.0.0 255.255.255.0 Serial0/0/0
!
!
!
!
control-plane
!
!
banner motd ^CUnauthorized access prohibited!^C
!
line con 0
 password 7 110A1016141D
 logging synchronous
 login
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 password 7 00071A150754
logging synchronous
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Program 4

Configuring Basic RIPv2 protocol

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/1	172.30.10.1	255.255.255.0	N/A
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A
R2	G0/0	209.165.201.1	255.255.255.0	N/A
	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A
R3	G0/1	172.30.30.1	255.255.255.0	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
S1	N/A	VLAN 1	N/A	N/A
S3	N/A	VLAN 1	N/A	N/A
PC-A	NIC	172.30.10.3	255.255.255.0	172.30.10.1
PC-B	NIC	209.165.201.2	255.255.255.0	209.165.201.1
PC-C	NIC	172.30.30.3	255.255.255.0	172.30.30.1

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Configure and Verify RIPv2 Routing

- Configure RIPv2 on the routers and verify that it is running.
- Configure a passive interface.
- Examine routing tables.
- Disable automatic summarization.
- Configure a default route.
- Verify end-to-end connectivity.

Background / Scenario

RIP version 2 (RIPv2) is used for routing of IPv4 addresses in small networks. RIPv2 is a classless, distance-vector routing protocol, as defined by RFC 1723. Because RIPv2 is a classless routing protocol, subnet masks are included in the routing updates. By default, RIPv2 automatically summarizes networks at major network boundaries. When automatic summarization has been disabled, RIPv2 no longer summarizes networks to their classful address at boundary routers.

In this lab, you will configure the network topology with RIPv2 routing, disable automatic summarization, propagate a default route, and use CLI commands to display and verify RIP routing information.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in this lab. Refer to the Router Interface Summary Table at the end of the lab for the correct interface identifiers.

Note: Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 3 Routers (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 2 Switches (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)
- 3 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and Serial cables as shown in the topology

Part 11: **Build the Network and Configure Basic Device Settings**

In Part 1, you will set up the network topology and configure basic settings.

Step 1: Cable the network as shown in the topology.

Step 2: Initialize and reload the router and switch.

Step 3: Configure basic settings for each router and switch.

a. Disable DNS lookup.

b. Configure device names as shown in the topology.

c. Configure password encryption.

d. Assign **class** as the privileged EXEC password.

e. Assign **cisco** as the console and vty passwords.

f. Configure a MOTD banner to warn users that unauthorized access is prohibited.

g. Configure **logging synchronous** for the console line.

h. Configure the IP addresses listed in the Addressing Table for all interfaces.

i. Configure a description for each interface with an IP address.

j. Configure the clock rate, if applicable, to the DCE serial interface.

k. Copy the running-configuration to the startup-configuration.

Step 4: Configure PC IP Addressing.

Refer to the Addressing Table for IP address information of the PCs.

Step 5: Test connectivity.

At this point, the PCs are unable to ping each other.

a. Each workstation should be able to ping the attached router. Verify and troubleshoot if necessary.

b. The routers should be able to ping one another. Verify and troubleshoot if necessary.

Part 12: Configure and Verify RIPv2 Routing

In Part 2, you will configure RIPv2 routing on all routers in the network and then verify that the routing tables are updated correctly. After RIPv2 has been verified, you will disable automatic summarization, configure a default route, and verify end-to-end connectivity.

Step 1: Configure RIPv2 routing.

a. Configure RIPv2 on R1as the routing protocol and advertise the appropriate connected networks.

R1# **config t**
R1(config)# **router rip**
R1(config-router)# **version 2**
R1(config-router)# **passive-interface g0/1**
R1(config-router)# **network 172.30.0.0**
R1(config-router)# **network 10.0.0.0**

The **passive-interface** command stops routing updates out the specified interface. This process prevents unnecessary routing traffic on the LAN. However, the network that the specified interface belongs to is still advertised in routing updates that are sent out across other interfaces.

b. Configure RIPv2 on R3 and use the **network** statement to add the appropriate connected networks and prevent routing updates on the LAN interface.

c. Configure RIPv2 on R2 and use the network statements to add the appropriate connected networks. Do not advertise the 209.165.201.0 network.

Note: It is not necessary to make the G0/0 interface passive on R2 because the network associated with this interface is not being advertised.

Step 2: Examine the current state of the network.

a. The status of the two serial links can quickly be verified using the **show ip interface brief** command on R2.

R2# **show ip interface brief**

Interface	IP-Address	OK?	Method	Status	Protocol
Embedded-Service-Engine0/0	unassigned	YES	unset	administratively down	down
GigabitEthernet0/0	209.165.201.1	YES	manual	up	up
GigabitEthernet0/1	unassigned	YES	unset	administratively down	down
Serial0/0/0	10.1.1.2	YES	manual	up	up
Serial0/0/1	10.2.2.2	YES	manual	up	up

b. Check connectivity between PCs.

From PC-A, is it possible to ping PC-B? _____ Why?

_____

No, R2 is not advertising the route to PC-B.

From PC-A, is it possible to ping PC-C? _____ Why?

_____

No, R1 and R3 do not have routes to the remote networks, and R2, incorrectly has two equal cost load balancing routes to the 172.30.0.0/16 subnet..

From PC-C, is it possible to ping PC-B? _____ Why?

_____

No, R2 is not advertising the route to PC-B.

From PC-C, is it possible to ping PC-A? _____ Why?

_____

No, R1 and R3 do not have routes to the remote networks, and R2, incorrectly has two equal cost loadbalancing routes to the 172.30.0.0/16 subnet..

c. Verify that RIPv2 is running on the routers.

You can use the **debug ip rip**, **show ip protocols**, and **show run** commands to confirm that RIPv2 is running. The **show ip protocols** command output for R1 is shown below.

R1# **show ip protocols**

Routing Protocol is "rip"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Sending updates every 30 seconds, next due in 7 seconds

Invalid after 180 seconds, hold down 180, flushed after 240

Redistributing: rip

Default version control: send version 2, receive 2

Interface	Send	Recv	Triggered RIP	Key-chain
Serial0/0/0	2	2		

Automatic network summarization is in effect

Maximum path: 4

Routing for Networks:

10.0.0.0

172.30.0.0

Passive Interface(s):

GigabitEthernet0/1

Routing Information Sources:

Gateway	Distance	Last Update
10.1.1.2	120	

Distance: (default is 120)

When issuing the **debug ip rip** command on R2, what information is provided that confirms RIPv2 is running?

_____

RIP: sending v2 updates to 224.0.0.9 via Serial 0/0/0 (10.1.1.2).

When you are finished observing the debugging outputs, issue the **undebug all** command at the privileged EXEC prompt.

When issuing the **show run** command on R3, what information is provided that confirms RIPv2 is running?

router rip
version 2

d. Examine the automatic summarization of routes.

The LANs connected to R1 and R3 are composed of discontiguous networks. R2 displays two equal-cost paths to the 172.30.0.0/16 network in the routing table. R2 displays only the major classful network address of 172.30.0.0 and does not display any of the subnets for this network.

R2# **show ip route**

<Output omitted>

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.1.1.0/30 is directly connected, Serial0/0/0
L 10.1.1.2/32 is directly connected, Serial0/0/0
C 10.2.2.0/30 is directly connected, Serial0/0/1
L 10.2.2.2/32 is directly connected, Serial0/0/1
R 172.30.0.0/16 [120/1] via 10.2.2.1, 00:00:23, Serial0/0/1
[120/1] via 10.1.1.1, 00:00:09, Serial0/0/0
209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C 209.165.201.0/24 is directly connected, GigabitEthernet0/0
L 209.165.201.1/32 is directly connected, GigabitEthernet0/0

R1 displays only its own subnet for the 172.30.10.0/24 network. R1 does not have a route for the 172.30.30.0/24 subnet on R3.

R1# **show ip route**

<Output omitted>

10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C 10.1.1.0/30 is directly connected, Serial0/0/0
L 10.1.1.1/32 is directly connected, Serial0/0/0
R 10.2.2.0/30 [120/1] via 10.1.1.2, 00:00:21, Serial0/0/0
172.30.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.30.10.0/24 is directly connected, GigabitEthernet0/1
L 172.30.10.1/32 is directly connected, GigabitEthernet0/1

R3 only displays its own subnet for the 172.30.30.0/24 network. R3 does not have a route for the 172.30.10.0/24 subnets on R1.

R3# **show ip route**

<Output omitted>

10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C 10.2.2.0/30 is directly connected, Serial0/0/1
L 10.2.2.1/32 is directly connected, Serial0/0/1
R 10.1.1.0/30 [120/1] via 10.2.2.2, 00:00:23, Serial0/0/1
172.30.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.30.30.0/24 is directly connected, GigabitEthernet0/1
L 172.30.30.1/32 is directly connected, GigabitEthernet0/1

Use the **debug ip rip** command on R2 to determine the routes received in the RIP updates from R3 and list them here.

172.30.0.0/16

R3 is not sending any of the 172.30.0.0 subnets, only the summarized route of 172.30.0.0/16, including the subnet mask. Therefore, the routing tables on R1 and R2 do not display the 172.30.0.0 subnets on R3.

Step 3: Disable automatic summarization.

a. The **no auto-summary** command is used to turn off automatic summarization in RIPv2. Disable auto summarization on all routers. The routers will no longer summarize routes at major classful network boundaries. R1 is shown here as an example.

R1(config)# **router rip**
R1(config-router)# **no auto-summary**

b. Issue the **clear ip route *** command to clear the routing table.

R1(config-router)# **end**
R1# **clear ip route ***

c. Examine the routing tables. Remember that it will take some time to converge the routing tables after clearing them.

The LAN subnets connected to R1 and R3 should now be included in all three routing tables.

R2# **show ip route**
<Output omitted>
Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.1.1.0/30 is directly connected, Serial0/0/0
L 10.1.1.2/32 is directly connected, Serial0/0/0
C 10.2.2.0/30 is directly connected, Serial0/0/1
L 10.2.2.2/32 is directly connected, Serial0/0/1
172.30.0.0/16 is variably subnetted, 3 subnets, 2 masks
R 172.30.0.0/16 [120/1] via 10.2.2.1, 00:01:01, Serial0/0/1
[120/1] via 10.1.1.1, 00:01:15, Serial0/0/0
R 172.30.10.0/24 [120/1] via 10.1.1.1, 00:00:21, Serial0/0/0
R 172.30.30.0/24 [120/1] via 10.2.2.1, 00:00:04, Serial0/0/1
209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C 209.165.201.0/24 is directly connected, GigabitEthernet0/0
L 209.165.201.1/32 is directly connected, GigabitEthernet0/0

R1# **show ip route**
<Output omitted>
Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C 10.1.1.0/30 is directly connected, Serial0/0/0
L 10.1.1.1/32 is directly connected, Serial0/0/0
R 10.2.2.0/30 [120/1] via 10.1.1.2, 00:00:12, Serial0/0/0
172.30.0.0/16 is variably subnetted, 3 subnets, 2 masks
C 172.30.10.0/24 is directly connected, GigabitEthernet0/1
L 172.30.10.1/32 is directly connected, GigabitEthernet0/1
R 172.30.30.0/24 [120/2] via 10.1.1.2, 00:00:12, Serial0/0/0

R3# **show ip route**
<Output omitted>
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C 10.2.2.0/30 is directly connected, Serial0/0/1
L 10.2.2.1/32 is directly connected, Serial0/0/1
R 10.1.1.0/30 [120/1] via 10.2.2.2, 00:00:23, Serial0/0/1
172.30.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.30.30.0/24 is directly connected, GigabitEthernet0/1

L 172.30.30.1/32 is directly connected, GigabitEthernet0/1
R 172.30.10.0 [120/2] via 10.2.2.2, 00:00:16, Serial0/0/1

d. Use the **debug ip rip** command on R2 to examine the RIP updates.

R2# debug ip rip

After 60 seconds, issue the **no debug ip rip** command.

What routes are in the RIP updates that are received from R3?

_____

172.30.30.0/24

Are the subnet masks included in the routing updates? _____ yes

Step 4: Configure and redistribute a default route for Internet access.

a. From R2, create a static route to network 0.0.0.0 0.0.0.0, using the **ip route** command. This forwards any traffic with an unknown destination address to PC-B at 209.165.201.2, simulating the Internet by setting a Gateway of Last Resort on router R2.

R2(config)# **ip route 0.0.0.0 0.0.0.0 209.165.201.2**

b. R2 will advertise a route to the other routers if the **default-information originate** command is added to its RIP configuration.

R2(config)# **router rip**
R2(config-router)# **default-information originate**

Step 5: Verify the routing configuration.

a. View the routing table on R1.

R1# **show ip route**
<Output omitted>
Gateway of last resort is 10.1.1.2 to network 0.0.0.0

R* 0.0.0.0/0 [120/1] via 10.1.1.2, 00:00:13, Serial0/0/0
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C 10.1.1.0/30 is directly connected, Serial0/0/0
L 10.1.1.1/32 is directly connected, Serial0/0/0
R 10.2.2.0/30 [120/1] via 10.1.1.2, 00:00:13, Serial0/0/0
172.30.0.0/16 is variably subnetted, 3 subnets, 2 masks
C 172.30.10.0/24 is directly connected, GigabitEthernet0/1
L 172.30.10.1/32 is directly connected, GigabitEthernet0/1
R 172.30.30.0/24 [120/2] via 10.1.1.2, 00:00:13, Serial0/0/0

How can you tell from the routing table that the subnetted network shared by R1 and R3 has a pathway for Internet traffic?

_____

_____

There is a Gateway of Last Resort, and the default route shows up in the table as being learned via RIP.

b. View the routing table on R2.

How is the pathway for Internet traffic provided in its routing table?

_____

_____

R2 has a default static route to 0.0.0.0 via 209.165.201.2, which is directly connected to G0/0.

Step 6: Verify connectivity.

a. Simulate sending traffic to the Internet by pinging from PC-A and PC-C to 209.165.201.2.

Were the pings successful? _____ Yes

b. Verify that hosts within the subnetted network can reach each other by pinging between PC-A and PC-C.

Were the pings successful? _____ Yes

Note: It may be necessary to disable the PCs firewall.

Reflection

1. Why would you turn off automatic summarization for RIPv2?

_____

So the routers will no longer summarize routes at major classful network boundaries.

2. How did R1 and R3 learn the pathway to the Internet?

_____

From RIP routing updates received from the router where the default route was configured (R2).

Router Interface Summary Table

Router Interface Summary				
Router Model	**Ethernet Interface #1**	**Ethernet Interface #2**	**Serial Interface #1**	**Serial Interface #2**
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configs - Final

Router R1

```
R1# show run
Building configuration...

Current configuration : 1787 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

```
!
hostname R1
!
boot-start-marker
boot-end-marker
!
enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2
!
no aaa new-model
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 description R1 LAN
 ip address 172.30.10.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0/0
 description Link to R2
 ip address 10.1.1.1 255.255.255.252
clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
!
router rip
 version 2
 passive-interface GigabitEthernet0/1
 network 10.0.0.0
 network 172.30.0.0
 no auto-summary
!
ip forward-protocol nd
!
no ip http server
```

```
no ip http secure-server
!
control-plane
!
banner motd ^CUnauthorized access is strictly prohibited.^C
!
line con 0
 password 7 045802150C2E
 logging synchronous
 login
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 password 7 060506324F41
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Router R2

```
R2#show run
Building configuration...

Current configuration : 2073 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2
!
no aaa new-model
!
no ip domain lookup
ip cef
!
redundancy
```

```
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 description R2 LAN
 ip address 209.165.201.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 description Link to R1
 ip address 10.1.1.2 255.255.255.252
!
interface Serial0/0/1
 description Link to R3
 ip address 10.2.2.2 255.255.255.252
clock rate 2000000
!
router rip
 version 2
 network 10.0.0.0
 default-information originate
 no auto-summary
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 209.165.201.2
!
control-plane
!
banner motd ^CUnauthorized access is strictly prohibited.^C
!
line con 0
 password 7 0822455D0A16
 logging synchronous
 login
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
```

```
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 password 7 110A1016141D
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Router R3

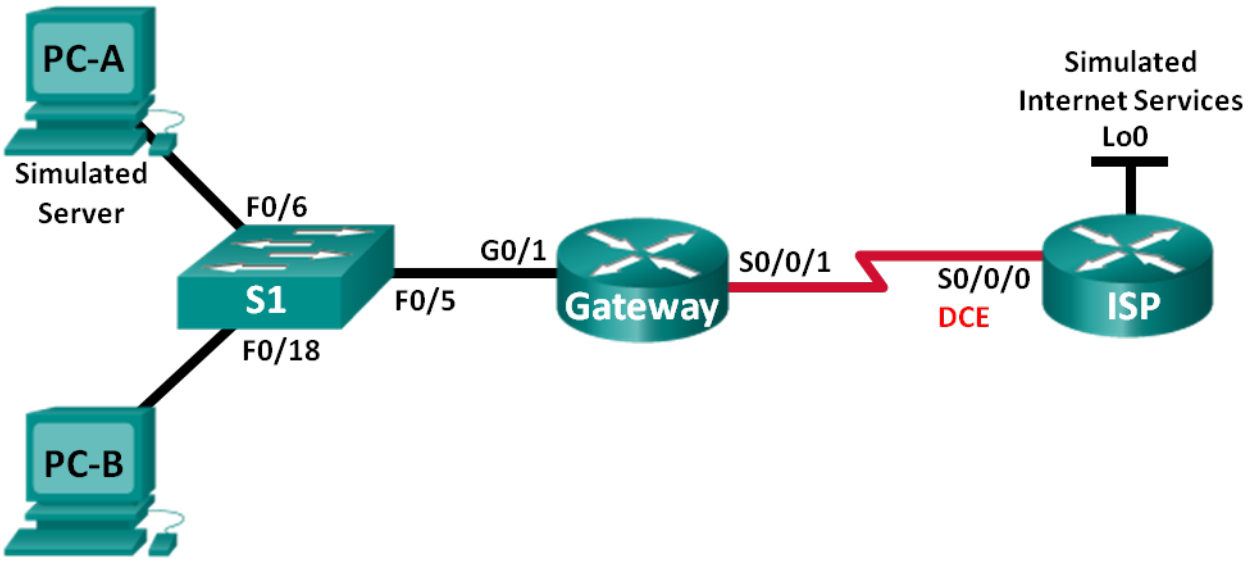
```
R3#show run
Building configuration...

Current configuration : 1847 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2
!
no aaa new-model
memory-size iomem 15
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
```

```
 description R3 LAN
 ip address 172.30.30.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 description Link to R2
 ip address 10.2.2.1 255.255.255.252
!
router rip
 version 2
 passive-interface GigabitEthernet0/1
 network 10.0.0.0
 network 172.30.0.0
 no auto-summary
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
control-plane
!
banner motd ^CUnauthorized access is strictly prohibited.^C
!
line con 0
 password 7 02050D480809
 logging synchronous
 login
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 password 7 14141B180F0B
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Program 5

Configuring Dynamic and Static NAT

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
Gateway	G0/1	192.168.1.1	255.255.255.0	N/A
	S0/0/1	209.165.201.18	255.255.255.252	N/A
ISP	S0/0/0 (DCE)	209.165.201.17	255.255.255.252	N/A
	Lo0	192.31.7.1	255.255.255.255	N/A
PC-A (Simulated Server)	NIC	192.168.1.20	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.1.21	255.255.255.0	192.168.1.1

Objectives

Part 1: Build the Network and Verify Connectivity

Part 2: Configure and Verify Static NAT

Part 3: Configure and Verify Dynamic NAT

Background / Scenario

Network Address Translation (NAT) is the process where a network device, such as a Cisco router, assigns a public address to host devices inside a private network. The main reason to use NAT is to reduce the number of public IP addresses that an organization uses because the number of available IPv4 public addresses is limited.

In this lab, an ISP has allocated the public IP address space of 209.165.200.224/27 to a company. This provides the company with 30 public IP addresses. The addresses, 209.165.200.225 to 209.165.200.241, are for static allocation and 209.165.200.242 to 209.165.200.254 are for dynamic allocation. A static route is used from the ISP to the

gateway router, and a default route is used from the gateway to the ISP router. The ISP connection to the Internet is simulated by a loopback address on the ISP router.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

Note: Make sure that the routers and switch have been erased and have no startup configurations. If you are unsure, contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 2 Routers (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 1 Switch (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)
- 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Part 13: Build the Network and Verify Connectivity

In Part 1, you will set up the network topology and configure basic settings, such as the interface IP addresses, static routing, device access, and passwords.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure PC hosts.

Step 3: Initialize and reload the routers and switches as necessary.

Step 4: Configure basic settings for each router.

a. Console into the router and enter global configuration mode.

b. Copy the following basic configuration and paste it to the running-configuration on the router.

```
no ip domain-lookup
service password-encryption
enable secret class
banner motd #
Unauthorized access is strictly prohibited. #
line con 0
password cisco
login
logging synchronous
line vty 0 4
password cisco
login
```

c. Configure the host name as shown in the topology.

d. Copy the running configuration to the startup configuration.

Step 5: Create a simulated web server on ISP.

a. Create a local user named **webuser** with an encrypted password of **webpass**.

ISP(config)# **username webuser privilege 15 secret webpass**

b. Enable the HTTP server service on ISP.

ISP(config)# **ip http server**

c. Configure the HTTP service to use the local user database.

ISP(config)# **ip http authentication local**

Step 6: Configure static routing.

a. Create a static route from the ISP router to the Gateway router using the assigned public network address range 209.165.200.224/27.

ISP(config)# **ip route 209.165.200.224 255.255.255.224 209.165.201.18**

b. Create a default route from the Gateway router to the ISP router.

Gateway(config)# **ip route 0.0.0.0 0.0.0.0 209.165.201.17**

Step 7: Save the running configuration to the startup configuration.

Step 8: Verify network connectivity.

a. From the PC hosts, ping the G0/1 interface on the Gateway router. Troubleshoot if the pings are unsuccessful.

b. Display the routing tables on both routers to verify that the static routes are in the routing table and configured correctly on both routers.

Part 14: Configure and Verify Static NAT

Static NAT uses a one-to-one mapping of local and global addresses, and these mappings remain constant. Static NAT is particularly useful for web servers or devices that must have static addresses that are accessible from the Internet.

Step 1: Configure a static mapping.

A static map is configured to tell the router to translate between the private inside server address 192.168.1.20 and the public address 209.165.200.225. This allows a user from the Internet to access PC-A. PC-A is simulating a server or device with a constant address that can be accessed from the Internet.

Gateway(config)# **ip nat inside source static 192.168.1.20 209.165.200.225**

Step 2: Specify the interfaces.

Issue the **ip nat inside** and **ip nat outside** commands to the interfaces.

Gateway(config)# **interface g0/1**
Gateway(config-if)# **ip nat inside**
Gateway(config-if)# **interface s0/0/1**
Gateway(config-if)# **ip nat outside**

Step 3: Test the configuration.

a. Display the static NAT table by issuing the **show ip nat translations** command.

Gateway# **show ip nat translations**

Pro	Inside global	Inside local	Outside local	Outside global
---	209.165.200.225	192.168.1.20	---	---

What is the translation of the Inside local host address?

192.168.1.20 = _____ 209.165.200.225

The Inside global address is assigned by?

_____

The router from the NAT pool.

The Inside local address is assigned by?

_____

The administrator for the workstation.

b. From PC-A, ping the Lo0 interface (192.31.7.1) on ISP. If the ping was unsuccessful, troubleshoot and correct the issues. On the Gateway router, display the NAT table.

Gateway# **show ip nat translations**

Pro	Inside global	Inside local	Outside local	Outside global
icmp	209.165.200.225:1	192.168.1.20:1	192.31.7.1:1	192.31.7.1:1
---	209.165.200.225	192.168.1.20	---	---

A NAT entry was added to the table with ICMP listed as the protocol when PC-A sent an ICMP request (ping) to 192.31.7.1 on ISP.

What port number was used in this ICMP exchange? _____ 1, answers will vary.

Note: It may be necessary to disable the PC-A firewall for the ping to be successful.

c. From PC-A, telnet to the ISP Lo0 interface and display the NAT table.

Pro	Inside global	Inside local	Outside local	Outside global
icmp	209.165.200.225:1	192.168.1.20:1	192.31.7.1:1	192.31.7.1:1
tcp	209.165.200.225:1034	192.168.1.20:1034	192.31.7.1:23	192.31.7.1:23
---	209.165.200.225	192.168.1.20	---	---

Note: The NAT for the ICMP request may have timed out and been removed from the NAT table.

What was the protocol used in this translation? _____ tcp

What are the port numbers used?

Inside global / local: _____ 1034, answers will vary.

Outside global / local: _____ 23

d. Because static NAT was configured for PC-A, verify that pinging from ISP to PC-A at the static NAT public address (209.165.200.225) is successful.

e. On the Gateway router, display the NAT table to verify the translation.

Gateway# **show ip nat translations**

Pro	Inside global	Inside local	Outside local	Outside global
icmp	209.165.200.225:12	192.168.1.20:12	209.165.201.17:12	209.165.201.17:12
---	209.165.200.225	192.168.1.20	---	---

Notice that the Outside local and Outside global addresses are the same. This address is the ISP remote network source address. For the ping from the ISP to succeed, the Inside global static NAT address 209.165.200.225 was translated to the Inside local address of PC-A (192.168.1.20).

f. Verify NAT statistics by using the **show ip nat statistics** command on the Gateway router.

Gateway# **show ip nat statistics**

Total active translations: 2 (1 static, 1 dynamic; 1 extended)

Peak translations: 2, occurred 00:02:12 ago

Outside interfaces:

Serial0/0/1

Inside interfaces:
GigabitEthernet0/1
Hits: 39 Misses: 0
CEF Translated packets: 39, CEF Punted packets: 0
Expired translations: 3
Dynamic mappings:

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0

Note: This is only a sample output. Your output may not match exactly.

Part 15: Configure and Verify Dynamic NAT

Dynamic NAT uses a pool of public addresses and assigns them on a first-come, first-served basis. When an inside device requests access to an outside network, dynamic NAT assigns an available public IPv4 address from the pool. Dynamic NAT results in a many-to-many address mapping between local and global addresses.

Step 1: Clear NATs.

Before proceeding to add dynamic NATs, clear the NATs and statistics from Part 2.

Gateway# **clear ip nat translation ***
Gateway# **clear ip nat statistics**

Step 2: Define an access control list (ACL) that matches the LAN private IP address range.

ACL 1 is used to allow 192.168.1.0/24 network to be translated.

Gateway(config)# **access-list 1 permit 192.168.1.0 0.0.0.255**

Step 3: Verify that the NAT interface configurations are still valid.

Issue the **show ip nat statistics** command on the Gateway router to verify the NAT configurations.

Gateway# **show ip nat statistics**
Total active translations: 1 (1 static, 0 dynamic; 0 extended)
Peak translations: 0
Outside interfaces:
Serial0/0/1
Inside interfaces:
FastEthernet0/1
Hits: 0 Misses: 0
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0

Step 4: Define the pool of usable public IP addresses.

Gateway(config)# **ip nat pool public_access 209.165.200.242 209.165.200.254 netmask 255.255.255.224**

Step 5: Define the NAT from the inside source list to the outside pool.

Note: Remember that NAT pool names are case-sensitive and the pool name entered here must match that used in the previous step.

Gateway(config)# **ip nat inside source list 1 pool public_access**

Step 6: Test the configuration.

a. From PC-B, ping the Lo0 interface (192.31.7.1) on ISP. If the ping was unsuccessful, troubleshoot and correct the issues. On the Gateway router, display the NAT table.

Gateway# **show ip nat translations**

```
Pro Inside global      Inside local       Outside local      Outside global
--- 209.165.200.225    192.168.1.20       ---                ---
icmp 209.165.200.242:1 192.168.1.21:1     192.31.7.1:1       192.31.7.1:1
--- 209.165.200.242    192.168.1.21       ---                ---
```

What is the translation of the Inside local host address for PC-B?

192.168.1.21 = _____ 209.165.200.242

A dynamic NAT entry was added to the table with ICMP as the protocol when PC-B sent an ICMP message to 192.31.7.1 on ISP.

What port number was used in this ICMP exchange? _____ 1, answers will vary.

b. From PC-B, open a browser and enter the IP address of the ISP-simulated web server (Lo0 interface). When prompted, log in as **webuser** with a password of **webpass**.

c. Display the NAT table.

```
Pro Inside global      Inside local       Outside local      Outside global
--- 209.165.200.225    192.168.1.20       ---                ---
tcp 209.165.200.242:1038 192.168.1.21:1038 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1039 192.168.1.21:1039 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1040 192.168.1.21:1040 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1041 192.168.1.21:1041 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1042 192.168.1.21:1042 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1043 192.168.1.21:1043 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1044 192.168.1.21:1044 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1045 192.168.1.21:1045 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1046 192.168.1.21:1046 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1047 192.168.1.21:1047 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1048 192.168.1.21:1048 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1049 192.168.1.21:1049 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1050 192.168.1.21:1050 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1051 192.168.1.21:1051 192.31.7.1:80     192.31.7.1:80
tcp 209.165.200.242:1052 192.168.1.21:1052 192.31.7.1:80     192.31.7.1:80
--- 209.165.200.242    192.168.1.22       ---                ---
```

What protocol was used in this translation? _____ tcp

What port numbers were used?

Inside: _____ 1038 to 1052. Answers will vary.

Outside: _____ 80

What well-known port number and service was used? _____ port 80, www or http

d. Verify NAT statistics by using the **show ip nat statistics** command on the Gateway router.

Gateway# **show ip nat statistics**

```
Total active translations: 3 (1 static, 2 dynamic; 1 extended)
Peak translations: 17, occurred 00:06:40 ago
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  GigabitEthernet0/1
Hits: 345  Misses: 0
CEF Translated packets: 345, CEF Punted packets: 0
Expired translations: 20
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool public_access refcount 2
 pool public_access: netmask 255.255.255.224
        start 209.165.200.242 end 209.165.200.254
        type generic, total addresses 13, allocated 1 (7%), misses 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

Note: This is only a sample output. Your output may not match exactly.

Step 7: Remove the static NAT entry.

In Step 7, the static NAT entry is removed and you can observe the NAT entry.

a. Remove the static NAT from Part 2. Enter **yes** when prompted to delete child entries.

```
Gateway(config)# no ip nat inside source static 192.168.1.20 209.165.200.225

Static entry in use, do you want to delete child entries? [no]: yes
```

b. Clear the NATs and statistics.

c. Ping the ISP (192.31.7.1) from both hosts.

d. Display the NAT table and statistics.

```
Gateway# show ip nat statistics
Total active translations: 4 (0 static, 4 dynamic; 2 extended)
Peak translations: 15, occurred 00:00:43 ago
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  GigabitEthernet0/1
Hits: 16  Misses: 0
CEF Translated packets: 285, CEF Punted packets: 0
Expired translations: 11
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool public_access refcount 4
 pool public_access: netmask 255.255.255.224
        start 209.165.200.242 end 209.165.200.254
        type generic, total addresses 13, allocated 2 (15%), misses 0

Total doors: 0
```

Appl doors: 0
Normal doors: 0
Queued Packets: 0

Gateway# **show ip nat translation**

Pro	Inside global	Inside local	Outside local	Outside global
icmp	209.165.200.243:512	192.168.1.20:512	192.31.7.1:512	192.31.7.1:512
---	209.165.200.243	192.168.1.20	---	---
icmp	209.165.200.242:512	192.168.1.21:512	192.31.7.1:512	192.31.7.1:512
---	209.165.200.242	192.168.1.21	---	---

Note: This is only a sample output. Your output may not match exactly.

Router Interface Summary Table

Router Interface Summary				
Router Model	**Ethernet Interface #1**	**Ethernet Interface #2**	**Serial Interface #1**	**Serial Interface #2**
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configs

Gateway (After Part 2)

Gateway# **show run**
Building configuration...

Current configuration : 1666 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Gateway
!
boot-start-marker
boot-end-marker
!

```
enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2
!
no aaa new-model
memory-size iomem 15
!
no ip domain lookup
ip cef
no ipv6 cef
multilink bundle-name authenticated
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 ip address 209.165.201.18 255.255.255.252
 ip nat outside
 ip virtual-reassembly in
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip nat inside source static 192.168.1.20 209.165.200.225
ip route 0.0.0.0 0.0.0.0 209.165.201.17
!
control-plane
!
line con 0
 password cisco
 logging synchronous
 login
line aux 0
```

```
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 password cisco
login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Gateway (Final)

```
Gateway# show run
Building configuration...

Current configuration : 1701 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Gateway
!
boot-start-marker
boot-end-marker
!
enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2
!
no aaa new-model
memory-size iomem 15
!
no ip domain lookup
ip cef
no ipv6 cef
multilink bundle-name authenticated
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
```

```
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 ip address 209.165.201.18 255.255.255.252
 ip nat outside
 ip virtual-reassembly in
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip nat pool public_access 209.165.200.242 209.165.200.254 netmask 255.255.255.224
ip nat inside source list 1 pool public_access
ip route 0.0.0.0 0.0.0.0 209.165.201.17
!
access-list 1 permit 192.168.1.0 0.0.0.255
!
control-plane
!
line con 0
 password cisco
 logging synchronous
 login
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 password cisco
login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

ISP (Final)

```
ISP# show run
Building configuration...

Current configuration : 1557 bytes
!
! Last configuration change at 09:16:34 UTC Sun Mar 24 2013
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ISP
!
boot-start-marker
boot-end-marker
!
enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2
!
no aaa new-model
memory-size iomem 10
!
ip cef
no ipv6 cef
multilink bundle-name authenticated
!
username webuser privilege 15 secret 4 ZMYyKvmzVsyor8jHyP9ox.cMoz9loLfZN75illtozY2
!
interface Loopback0
 ip address 192.31.7.1 255.255.255.255
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 ip address 209.165.201.17 255.255.255.252
 clock rate 128000
!
```

```
interface Serial0/0/1
 no ip address
 shutdown
!
ip forward-protocol nd
!
ip http server
ip http authentication local
no ip http secure-server
!
ip route 209.165.200.224 255.255.255.224 209.165.201.18
!
control-plane
!
line con 0
 password cisco
 logging synchronous
 login
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 password cisco
login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

ADDITIONAL QUESTIONS

1. **What is a computer network?**

   A computer network is a group of computers and devices that are connected together to share resources such as data, printers, and internet access.

2. **What are the different types of computer networks?**

   There are several types of computer networks, including:

- Local Area Network (LAN): A LAN is a network that is limited to a small geographical area, such as a single building or campus.
- Wide Area Network (WAN): A WAN is a network that spans a large geographical area, such as a city or country.
- Metropolitan Area Network (MAN): A MAN is a network that spans a large metropolitan area, such as a city.
- Personal Area Network (PAN): A PAN is a network that is used to connect devices in close proximity, such as a computer and a printer.

3. **What is a network topology?**

   A network topology is the arrangement of the various components (links, nodes, etc.) of a computer network. Various types of network topologies exist, including:

- Bus topology: All devices in a bus topology are linked to one central cable, or backbone.
- Star topology: The hub or switch in the middle of a star topology serves as the connection point for all devices.
- Ring topology: Each device among a ring topology is connected to two more devices in a circular pattern.

4. **Describe a router.**

   A router is a piece of hardware that connects multiple networks and routes data between them. Routers consult routing tables and protocols to decide which network to send data to.

5. **What is meant by Router ?**

   A router is an electronic device that interconnects two ormore computer networks, and selectively interchanges packets of data between them . A router is a networking device whose software and hardware are customized to thetasks of routing and forwarding information.

6. **What do u mean by NIC (Network Interface Card) ?**

A network card, network adapter, or NIC (network interface card) is a piece of computer hardware designed to allow computers to communicate over a computer network. It provides physical access to a networking medium and often provides a low-level addressing system through the use of MAC address

7. List the layers of OSI

a. Physical Layer
b. Data Link Layer
c. Network Layer
d. Transport Layer
e. Session Layer
f. Presentation Layer
g. Application Layer

8. What are the responsibilities of Transport Layer?

The Transport Layer is responsible for source-to-destination delivery of the entire message.
a. Service-point Addressing
b. Segmentation and reassembly
c. Connection Control
d. Flow Control
e. Error Control

9. What is CRC? What is Checksum?

CRC, is the most powerful of the redundancy checking techniques, is based on binary division.Checksum is used by the higher layer protocols (TCP/IP) for error detection

10. What is Redundancy?

The concept of including extra information in the transmission solely for the purpose of comparison. This technique is called redundancy.

11. What is socket programming?

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection.

12. Differences between TCP and UDP

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Type of Service	TCP is a connection-oriented protocol. Connection orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.
Reliability	TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
Error checking mechanism	TCP provides extensive error-checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error-checking mechanism using checksums.
Acknowledgment	An acknowledgment segment is present.	No acknowledgment segment.
Sequence	Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in order at the receiver.	There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer.
Speed	TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.
Retransmission	Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP).
Header Length	TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
Weight	TCP is heavy-weight.	UDP is lightweight.
Handshaking Techniques	Uses handshakes such as SYN, ACK, SYN-ACK	It's a connectionless protocol i.e. No handshake

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Broadcasting	TCP doesn't support Broadcasting.	UDP supports Broadcasting.
Protocols	TCP is used by HTTP, HTTPs, FTP, SMTP and Telnet.	UDP is used by DNS, DHCP, TFTP, SNMP, RIP, and VoIP.
Stream Type	The TCP connection is a byte stream.	UDP connection is a message stream.
Overhead	Low but higher than UDP.	Very low.
Applications	This protocol is primarily utilized in situations when a safe and trustworthy communication procedure is necessary, such as in email, on the web surfing, and in military services.	This protocol is used in situations where quick communication is necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc.