

Department of Computer Science and Engineering

4th Year 2nd Semester  
  
Course Title: Wireless Networks Laboratory  
Course Code: CSE-454  
  
**LAB REPORT**

**Submitted by:**Towfiqul Hoque

Class Roll: 379

Exam Roll: 170496

**Submitted To:**Dr. Md. Imdadul Islam

Professor, Department of CSE, JU

Bulbul Ahammad

Lecturer, Department of CSE, JU

**Submission Date**

10-03-2022

**Experiment-01**

**Title**: Determination of market penetration rate of a network.

**Objective:** From this we see Rp(Penetration rate) with respect to number of channels and also to GoS(grade of service) respectively.

**Problem:** A city has a population of 3×106. A mobile cellular service provider has 500 cells with 15 channels each to serve the people. On an average each user generates 1 call/hour with duration of 1.5 minutes. Determine market penetration of the service provider. Given the GoS (grade of service) of the network is 5%.

**Analytical solution:**Traffic intensity/user = λ.th =1.(1/60).1.5 = 0.025Erl/userFor each cell, n = 15, B = 5%From Erlang’s table, The offered traffic, A=10.633Erls.Number of users/cell = 10.63/0.025 = 425Total number of users = 500×425 = 212500Penetration rate = (Number of users/Total population) ×100 = (212500/3× 106) \*100 =7.08%

**MATLAB code:**

1. %Taking A=12, n=15 determine B(n, A) **using** recurrence relation
2. A = 12; % initial assumed offered traffic
3. n = 15; %number of channel
4. Bo=1;
5. B(1) = A\*1/(1+A\*1);
6. **for** i=2:n
7. B(i)=A\*B(i-1)/(i+A\*B(i-1));
8. end
9. B(n) %blocking probability
10. plot(1:n, B, 'r\*--', 'LineWidth', 2, 'MarkerSize',10)
11. xlabel('n')
12. ylabel('B')
13. grid on

**Result:**



1. %Determine offered traffic A taking n=15 and B=0.05
2. n=15;
3. Bexp = 0.05; %expected blocking probability
4. A = 8; %initial assumed offered traffic
5. e=2; %initial value of error
6. **while** e>0.0001
7. s=1;
8. **for** i=1:n;
9. s=s+(A^i)/factorial(i);
10. end
11. B=((A^n)./factorial(n))/s; %blocking probability
12. e= abs(B-Bexp);
13. A=A+0.001;
14. end %end ofwhile loop

**Result:** A = 8

1. %Using the offered traffic, A=10.63 determine market penetration.
2. Lamda =1/60; %arrival rate per user
3. A=10.63; %offered traffic
4. th = 1.5; %holding time of each user
5. a = Lamda\*th; %offered traffic per user
6. N = A/a ;%Number of user per cell
7. Nt = N\*500; %Total user of the network
8. P = 3\*10^6; %Total population of the city
9. Rp = (Nt/P)\*100; %penetration rate

**Result:** Rp = 5.3333

**Discussion:** We have used Erlang's theory and the recurrence relation to determine the blocking probability in this experiment. In terms of recurrence, we can observe that as the number of channels grows, the probability of blocking reduces. We can also calculate the market penetration rate and blocking probability for a specific traffic. The number of channels per cell can be raised to increase market penetration. We see from this that Rp increases as no. of channels increases and also as GoS increases.

**Experiment-02**

**Title**: 2D traffic for both similar and dissimilar cases.

**Objective:** In this experiment, we show mobile traffic narrowband and wideband for both similar and dissimilar cases.

**MATLAB code:**

1. %A1 = 3 Erls and A2 = 2 Erls, n = 8
2. A1=3; A2=2;
3. n=8; %Number of channel
4. **for** i=1:n+1 % Matab has lowest argument of 1 not 0 like C
5. **for** j=1:n+1
6. k=(i-1)+(j-1);
7. **if** k <= n
8. S(i, j)= ((A1^(i-1))/factorial(i-1))\* ((A2^(j-1))/factorial(j-1));
9. **else**% {(A1^0)/0!}\* {(A2^0)/0!} is stored at S(1,1)
10. S(i, j)=0;
11. end
12. end
13. end

**Result:**

S =1.0000 2.0000 2.0000 1.3333 0.6667 0.2667 0.0889 0.0254 0.0063

3.0000 6.0000 6.0000 4.0000 2.0000 0.8000 0.2667 0.0762 0

4.5000 9.0000 9.0000 6.0000 3.0000 1.2000 0.4000 0 0

4.5000 9.0000 9.0000 6.0000 3.0000 1.2000 0 0 0

3.3750 6.7500 6.7500 4.5000 2.2500 0 0 0 0

2.0250 4.0500 4.0500 2.7000 0 0 0 0 0

1.0125 2.0250 2.0250 0 0 0 0 0 0

0.4339 0.8679 0 0 0 0 0 0 0

0.1627 0 0 0 0 0 0 0 0

1. % rows should be made reversed to cope with Cartesian co-ordinate system
2. for i=1:n+1
3. **for** j=1:n+1
4. k=(n+1)-i+1; % k =n+1, n, n-1, …, 1 i.e. rows will be reversed
5. S1(i, j)=S(k, j); %sates are now S1(x,y)
6. end
7. end

**Result:**

S1 = 0.1627 0 0 0 0 0 0 0 0

0.4339 0.8679 0 0 0 0 0 0 0

1.0125 2.0250 2.0250 0 0 0 0 0 0

2.0250 4.0500 4.0500 2.7000 0 0 0 0 0

3.3750 6.7500 6.7500 4.5000 2.2500 0 0 0 0

4.5000 9.0000 9.0000 6.0000 3.0000 1.2000 0 0 0

4.5000 9.0000 9.0000 6.0000 3.0000 1.2000 0.4000 0 0

3.0000 6.0000 6.0000 4.0000 2.0000 0.8000 0.2667 0.0762 0

1.0000 2.0000 2.0000 1.3333 0.6667 0.2667 0.0889 0.0254 0.0063

1. 1.  T=0; %initial value of entire sample space
2. 2.  Su=0; %initial value of complete occupied states
3. 3.  **for** i=1:n+1
4. 4.  **for** j=1:n+1
5. 5.  T=T+S(i,j); % Sum of entire states
6. 6.  **if** (i+j)==(n+1)+1
7. 7.  Su=Su+S(i,j); % Sum of diagonal states
8. 8.  end
9. 9.  end
10. 10. end
11. 11. B=Su/T;%Blocking Probability

**Result:** B = 0.0700

Let us assume bandwidth of each arrival of A1 is 2kbps and that of A2 is 3kbps. Given A1 = 1 and A2 = 2 Erls. If total Bandwidth supported by the network is 9kbps, then probability state will be with the condition: 0 ≤ BW1 x1+BW2 x2 ≤ 9. Determine QoS of each traffic.

1. %Probability states before normalization in matrix form
2. A1=1; A2=2;
3. x1=2; x2=3;
4. BW = 9; M =ceil(BW/x1); %size of the matrix is M by M
5. **for** i=1:M
6. **for** j=1:M
7. e = (i-1)\*x1+(j-1)\*x2;
8. **if** e <= 9
9. S(i, j)= ((A1^(i-1))/factorial(i-1))\* ((A2^(j-1))/factorial(j-1));
10. **else**
11. S(i, j)=0;
12. end
13. end
14. end
16. % rows should be made reversed to cope with
17. %Cartesian co-ordinate system
18. **for** i=1:M
19. **for** j=1:M
20. k=M-i+1;
21. S1(i, j)=S(k, j); %sates are now S1(x,y)
22. end
23. end
24. % Normalized states
25. T=sum(sum(S1));
26. Sn=S1/T;
28. %Side open states
29. B2 = 0;
30. **for** i =1:M
31. **for** j =1:M
32. k=j+1;
33. **if** Sn(i, j) > 0 &&Sn(i, k)= =0
34. B2 = B2+Sn(i, j);
35. end
36. end
37. end
39. % open states of roof
40. B1= 0;
41. **for** i = 2:M %start with 2nd row
42. %index of row with k = i-1 cannot be 0
43. **for** j=1:M
44. k=i-1;
45. **if** Sn(i, j)>0 &&Sn(k, j)= =0
46. B1= B1+Sn(i, j);
47. end
48. end
49. end
50. % Sn(1,1) is the **default** open state of roof
51. B1=B1+Sn(1,1);

**Result:**

S =

1.0000 2.0000 2.0000 1.3333 0

1.0000 2.0000 2.0000 0 0

0.5000 1.0000 0 0 0

0.1667 0.3333 0 0 0

0.0417 0 0 0 0

Sn =

0.0031 0 0 0 0

0.0125 0.0249 0 0 0

0.0374 0.0748 0 0 0

0.0748 0.1495 0.1495 0 0

0.0748 0.1495 0.1495 0.0997 0

B1=0.2773

B2=0.3520

**Discussion:**

We see that our theoretical Result and practical Result are approximately same. Thus, we can compute blocking probability(B) of the network and QoS(Quality of service) for both newly originating call and handover call.

**Experiment-03**

**Title**-Simulation of off-on binary transmission.

**Objective:** Our task in this is to simulate off-on binary transmission.

**Theory:**

Exponential density function is expressed as, *x*(*t*)*=λeλt*

where *λ* is the PDF parameter and *t* is the observation period.

The cumulative distribution function(CDF) of exponential distribution,

*F(t)=*

=1-e-λt

Mean=1*/λ*

Variance=1*/λ*2

**MATLAB code:**

1. %Plot PDF of exponential distribution with λ=0.5.
2. t = 0:0.01:10;
3. Lambda=0.5;
4. x = exppdf(t,1/Lambda);
5. plot(t,x)
6. xlabel('t')
7. ylabel('f(t)')
8. title('Exponential PDF')

**Result:**



1. %Plot CDF of exponential distribution with λ=0.5
2. t = 0:0.001:10;
3. Lambda=0.5;
4. P = expcdf(t,1/Lambda);
5. plot(t,P)
6. xlabel('t')
7. ylabel('F(t)')
8. title('Exponential CDF')

**Result:**



1. %Matlab code Simulation of on-off binary transmission
2. M=200000; %Number of bits used **for** simulation against each SNR
3. **for** k=1:8, % loop **for** SNR
4. SNR=2+k\*2; %The value of SNR in dB
5. tx=randi(2, M, 1)-1; %Sequence of 0 and 1
6. rx=awgn(tx,SNR);
7. e(k)=0; %initialization of error
8. **for** i=1:M, %**for** loop of error
9. **if** tx(i)==1;
10. **if** rx(i)<=0.5;
11. e(k)=e(k)+1;
12. end
13. end
15. **if** tx(i)==0;
16. **if** rx (i)>=0.5;
17. e(k)=e(k)+1;
18. end
19. end
20. end %**for** loop of error
21. end %loop **for** SNR
23. pe=e/M; %probability of error
24. SNR=4:2:18;
25. SNR\_a=10.^(SNR/10); %absolute value of SNR
26. pb=qfunc(sqrt(SNR\_a/4)); %Theoretical Pb
27. plot(SNR,pe,'r>:',SNR,pb,'bs:')
28. legend('simulation','theory')
29. xlabel('SNR in db')
30. ylabel('Pe')
31. grid on;
32. abs((pe-pb)./pb)\*100

**Result:**



ans = 0.0864 0.6349 1.0918 1.7184 1.2638 0.1411 24.5210 1.9322

1. %Taking M =2000
2. M=2000; %Num of bits used **for** simulation against each SNR
3. **for** k=1:8, %loop **for** SNR
4. SNR=2+k\*2; %the value of the SNR in dB
5. tx=randi(2,M,1)-1; %sequence of 0and 1
6. rx=awgn(tx,SNR);
8. e(k)=0;
9. fori=1:M,
10. iftx(i)==1;
11. ifrx(i)<=0.5;
12. e(k)=e(k)+1;
13. end
14. end
16. iftx(i)==0;
17. ifrx(i)>=0.5;
18. e(k)=e(k)+1;
19. end
20. end
21. end
22. end
24. pe=e/M;
25. SNR=4:2:18;
26. SNR\_a=10.^(SNR/10);
27. pb=qfunc(sqrt(SNR\_a/4));
28. plot(SNR,pe,'r>:',SNR,pb,'bs:')
29. legend('simulation','theory')
30. xlabel('SNR in db')
31. ylabel('Pe')
32. grid on;
33. abs((pe-pb)./pb)\*100

**Result:** ans = 3.7140 2.3682 0.0653 0.1350 3.2922 14.6341 37.7395 100.0000



1. %Taking M =20000
2. M=20000; %Num of bits used **for** simulation against each SNR
3. **for** k=1:8, %loop **for** SNR
4. SNR=2+k\*2; %the value of the SNR in dB
5. tx=randi(2,M,1)-1; %sequence of 0and 1
6. rx=awgn(tx,SNR);
8. e(k)=0;
9. fori=1:M,
10. iftx(i)==1;
11. ifrx(i)<=0.5;
12. e(k)=e(k)+1;
13. end
14. end
16. iftx(i)==0;
17. ifrx(i)>=0.5;
18. e(k)=e(k)+1;
19. end
20. end
21. end
22. end
24. pe=e/M;
25. SNR=4:2:18;
26. SNR\_a=10.^(SNR/10);
27. pb=qfunc(sqrt(SNR\_a/4));
28. plot(SNR,pe,'r>:',SNR,pb,'bs:')
29. legend('simulation','theory')
30. xlabel('SNR in db')
31. ylabel('Pe')
32. grid on;
34. abs((pe-pb)./pb)\*100

**Result:** ans = 0.0001 0.3009 0.1738 4.0812 1.1432 6.4460 37.7395 180.1937



**Discussion:**

We can see that as we increase the value of number of bits used for simulation against each SNR (M), the theoretical and simulation values almost match.

**Experiment-04**

**Title**: Simulation of Cognitive Radio Network (CRN).

**Objective:** We simulate CRN to ensure maximum utilization of the frequency spectrum and to increase the efficiency of a frequency spectrum in this experiment.

**MATLAB code:**

1. % Gaussian Noise (AWGN) under H0
2. var =5; %variance of random variable
3. N = 10; %average of N random number will be considered as the random numberT(i)
4. M =100; %data size of simulation
5. T = zeros(M,1); % Initialization of array
6. % Compute realizations of the sample mean.
7. **for** i = 1:M
8. x = sqrt (var)\*randn(N,1);
9. %generate random number with normal pdf of mean 0 and variance of var
10. T (i) = mean (x);
11. end
12. % Set up SNR array.
13. gamma = 0:0.25:6;
14. P = zeros (length (gamma), 1);
15. % Determine the number of elements of the array T exceeds gamma(i) **for** each i
16. % the number is Mgam and use **this** to estimate the probability.
17. **for** i =1: length (gamma)
18. clear Mgam;
19. Mgam = find (T>gamma (i));
20. PFA\_sim(i) = length (Mgam)/M;
21. end
22. % Compute the analytical probability.
23. PFA\_th = qfunc (gamma/ (sqrt (var/N)));
24. % Under H1
25. A=2; %average value of signal when PU is active
26. % % Initialization of array
27. T = zeros(M,1);
28. % Compute realizations of the sample mean.
29. **for** i = 1:M
30. x = sqrt (var)\*randn (N,1)+A;
31. %generate random number with normal pdf of mean A and variance of var
32. T (i) = mean (x);
33. end
34. P = zeros (length (gamma), 1) ;% Set number of values of gamma.
35. % Determine the number of elements of the array T exceeds gamma(i) **for** each i
36. % the number is Mgam and use **this** to estimate the probability.
37. **for** i =1: length (gamma)
38. clear Mgain;
39. Mgam = find (T>gamma (i));
40. PD\_sim(i) = length (Mgam)/M;
41. end
42. % Compute the Analytical probability.
43. PD\_th = qfunc((gamma-A)/ (sqrt (var/N)));
44. plot(gamma, PD\_sim, 'b^--', gamma, PD\_th, 'dr-', gamma, PFA\_sim, 'k>--',
45. gamma, PFA\_th, 'sr-')
46. legend('Simuation PD', 'Analytical PD', 'Simuation PF', 'Analytical PF')
47. xlabel('gamma')
48. ylabel('PF and PD')
49. grid on

**Result:**



**Discussion:**

In this experiment, we see if both N and M increase then the simulation Result: is as close as the analytical Result:.

**Experiment-05**

**Title**: Simulation of wireless link under Rayleigh function.

**Objective:** The objective of this is to simulate the bit error rate of wireless communication system under Rayleigh fading. In this we will also simulate the Probability density function of the random variable. Then we will simulate probability of bit error rate. We will compare theoretical or analytical result with simulation result.

**MATLAB code:**

1. N=200000; x=0:0.1:5; sigma=1; u=rand(1,N);
2. r=sigma\*sqrt(-2\*log(u)); %Rayleigh distributed random data
3. pdf=(x/sigma^2).\*exp(-(x/sigma).^2/2); %Theoretical pdf
4. subplot(2,1,1)
5. z=hist(r, x);
6. zn=z/sum(z);
7. stem(x, zn, 'r')
8. title('Simulation')
9. subplot(2,1,2)
10. stem(x,pdf/sum(pdf))
11. title('Theoritical')
12. xlabel('pdf')
13. ylabel('x')
14. stem(abs(pdf/sum(pdf)-zn))

**Result:**



1. N=200000; x=0:0.1:5; sigma=1; u=rand(1,N);
2. r=sigma\*sqrt(-2\*log(u)); %Rayleigh distributed random data
3. pdf=(x/sigma^2).\*exp(-(x/sigma).^2/2); %Theoretical pdf
4. subplot(2,1,1)
5. z=hist(r, x);
6. zn=z/sum(z);
7. stem(x, zn, 'r')
8. title('Simulation')
9. subplot(2,1,2)
10. stem(x,pdf/sum(pdf))
11. title('Theoritical')
12. xlabel('pdf')
13. ylabel('x')
14. stem(abs(pdf/sum(pdf)-zn))

**Result:**



1. N=2000; x=0:0.1:5; sigma=1; u=rand(1,N);
2. r=sigma\*sqrt(-2\*log(u)); %Rayleigh distributed random data
3. pdf=(x/sigma^2).\*exp(-(x/sigma).^2/2); %Theoretical pdf
4. subplot(2,1,1)
5. z=hist(r, x);
6. zn=z/sum(z);
7. stem(x, zn, 'r')
8. title('Simulation')
9. subplot(2,1,2)
10. stem(x,pdf/sum(pdf))
11. title('Theoritical')

**Result:**



1. %Initialization of parameters
2. Eb=1; %energy per bit
3. Eb\_N0\_dB=0:1:20; %range of Eb/N0
4. N0=Eb\*10.^(-Eb\_N0\_dB/10);
5. %The range of N0
6. sigma=1;
7. BER=zeros(1,length(Eb\_N0\_dB));
9. %initialization of array
10. **for** i=1:length(Eb\_N0\_dB);
11. no\_error=0; no\_bit=0;
12. %taking m = 0 i.e. all 0s are transmitted
13. **while** no\_error<=100
14. u=rand; x=sigma\*sqrt(-2\*log(u));
15. noise=sqrt(N0(i))\*randn;
16. %normal pdf with var of No
17. y=x\*sqrt(Eb)+noise;
18. **if** y<=0
19. y\_d=1;
20. **else**
21. y\_d=0;
22. end%end of **if** statement
23. no\_bit=no\_bit+1;
24. no\_error=no\_error+y\_d;
25. end%end of **while** loop
26. BER(i)=no\_error/no\_bit;
27. end%end of **for** loop

30. rho=(Eb./N0)\* sigma\*sigma;
31. Pb=0.5\*(1-sqrt(rho./(1+rho)));
32. semilogy(Eb\_N0\_dB, BER, '--r\*',Eb\_N0\_dB, Pb, ':b>', 'LineWidth',2, 'MarkerSize',10)
33. legend('Simulation','Analytical')
34. xlabel('Eb/N0')
35. ylabel('BER')
36. grid on

**Result:**



1. %Initialization of parameters
2. clear all
3. close all
4. Eb=1; %energy per bit
5. Eb\_N0\_dB=0:1:20; %range of Eb/N0
6. N0=Eb\*10.^(-Eb\_N0\_dB/10);
7. %The range of N0
8. sigma=1;
9. BER=zeros(1,length(Eb\_N0\_dB));
10. %initialization of array
11. **for** i=1:length(Eb\_N0\_dB);
12. no\_error=0; no\_bit=0;
13. %taking m = 0 i.e. all 0s are transmitted
14. **while** no\_error<=50
15. u=rand; x=sigma\*sqrt(-2\*log(u));
16. noise=sqrt(N0(i))\*randn;
17. %normal pdf with var of No
18. y=x\*sqrt(Eb)+noise;
19. **if** y<=0
20. y\_d=1;
21. **else**
22. y\_d=0;
23. end %end of **if** statement
24. no\_bit=no\_bit+1;
25. no\_error=no\_error+y\_d;
26. end %end of **while** loop
27. BER(i)=no\_error/no\_bit;
28. end %end of **for** loop
29. rho=(Eb./N0)\* sigma\*sigma;
30. Pb=0.5\*(1-sqrt(rho./(1+rho)));
31. semilogy(Eb\_N0\_dB, BER, '--r\*',Eb\_N0\_dB, Pb, ':b>', 'LineWidth',2, 'MarkerSize',10)
32. legend('Simulation','Analytical')
33. xlabel('Eb/N0')
34. ylabel('BER')
35. grid on

**Result:**



1. %Initialization of parameters
2. clear all
3. close all
4. Eb=1; %energy per bit
5. Eb\_N0\_dB=0:1:20; %range of Eb/N0
6. N0=Eb\*10.^(-Eb\_N0\_dB/10);
7. %The range of N0
8. sigma=1;
9. BER=zeros(1,length(Eb\_N0\_dB));
10. %initialization of array
11. **for** i=1:length(Eb\_N0\_dB);
12. no\_error=0; no\_bit=0;
13. %taking m = 0 i.e. all 0s are transmitted
14. **while** no\_error<=500
15. u=rand; x=sigma\*sqrt(-2\*log(u));
16. noise=sqrt(N0(i))\*randn;
17. %normal pdf with var of No
18. y=x\*sqrt(Eb)+noise;
19. **if** y<=0
20. y\_d=1;
21. **else**
22. y\_d=0;
23. end %end of **if** statement
24. no\_bit=no\_bit+1;
25. no\_error=no\_error+y\_d;
26. end %end of **while** loop
27. BER(i)=no\_error/no\_bit;
28. end %end of **for** loop
29. rho=(Eb./N0)\* sigma\*sigma;
30. Pb=0.5\*(1-sqrt(rho./(1+rho)));
31. semilogy(Eb\_N0\_dB, BER, '--r\*',Eb\_N0\_dB, Pb, ':b>', 'LineWidth',2, 'MarkerSize',10)
32. legend('Simulation','Analytical')
33. xlabel('Eb/N0')
34. ylabel('BER')
35. grid on

**Result:**



**Discussion:**

We can observe that as the value of error increases, the difference between simulation and analytical values decreases.

**Experiment-06**

**Title**: Implementation of Wireless LAN (IEEE 802.11) Using Wireless Access Point and DHCP Server.

**Objective:** In this we implement a wireless LAN using wireless access point and a DHCP server using CISCO Packet Tracer software.

**Software:** CISO Packet Tracer

**Working Procedure:**

**1.** Open the software cisco packet tracer.

**2.** Select a 2960 switch, an accesspoint, a server and five PCs, and connect in the following way shown in figure-6.1.

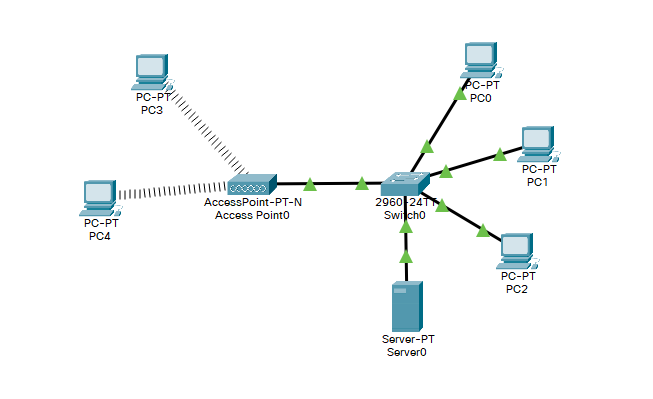


Figure-6.1

**3.** Click on PC3 and change the interface to The WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access. Perform similar task in PC4 to get wireless connection with the access point.

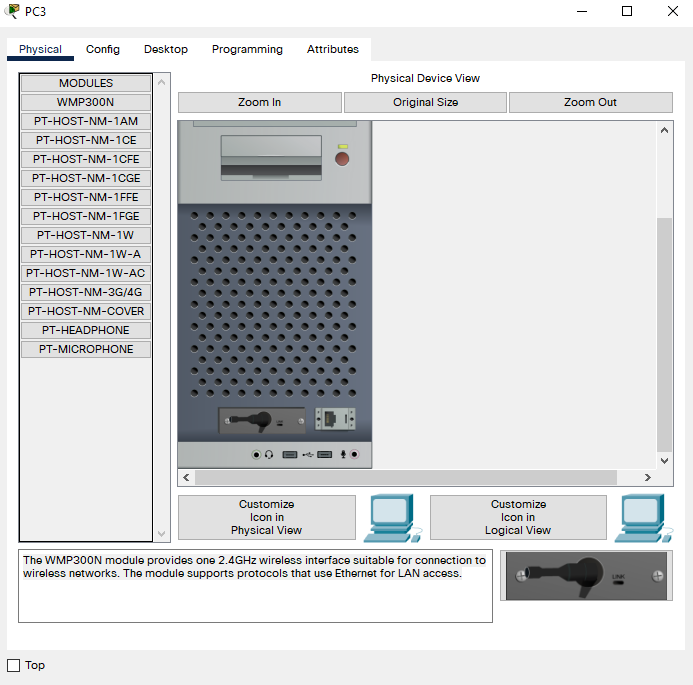


Figure-6.2

**4.** Click on the server0 and go to ‘Services’. In DHCP, select the service on and set the start IP address to 192.168.2.0 and subnet mask to 255.255.255.0 as shown in figure-. Then navigate to the ‘Desktop’ option and set the IPv4 address to 192.168.2.3 and subnet mask to 255.255.255.0 as shown in figure-6.3.

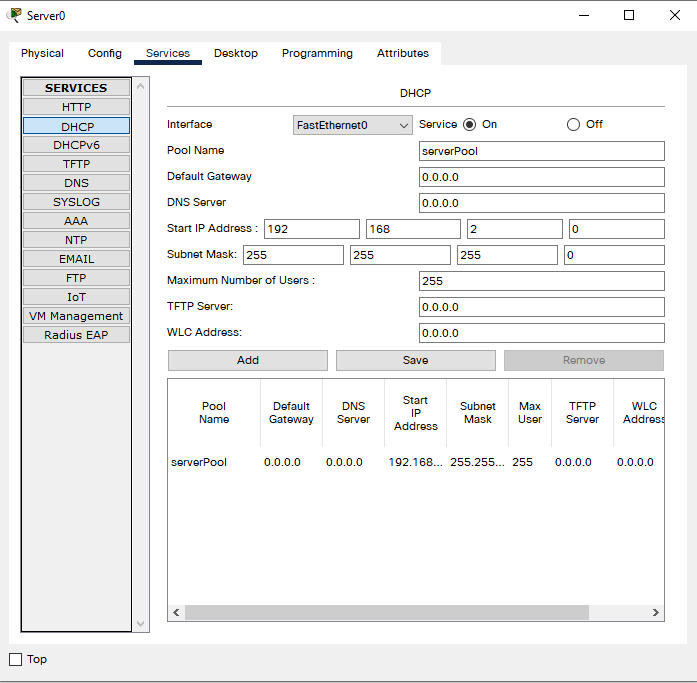


Figure-6.3

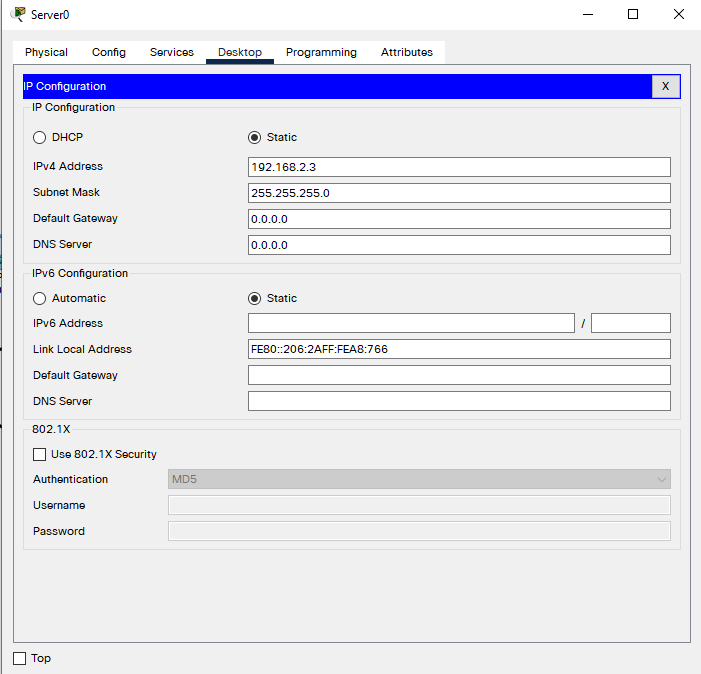


Figure-6.4

**5.** Now select the Access point and navigate to ‘Config’ option. Open ‘Port 1’ option and set SSID to “CSEJU”. Next, enable the WEP option and set WEP key to 0123456789 as shown in figure-6.5.

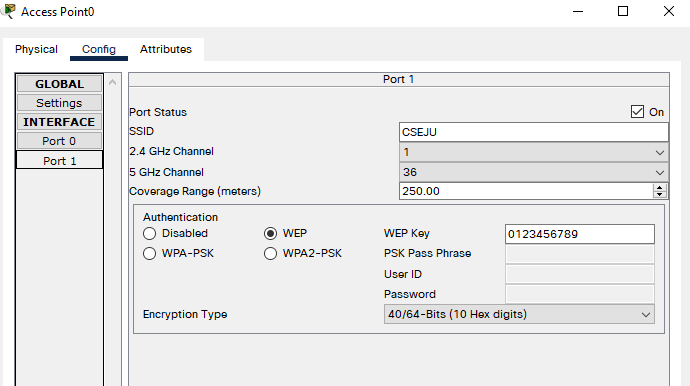


Figure-6.5

**6.**Select the wireless PC3 and navigate to the option Desktop 🡪 PC wireless 🡪 Connect. After a few seconds the name “CSEJU” will appear in the list of wireless networks and connect to the network with the provided WEP key as shown in figure-6.6.

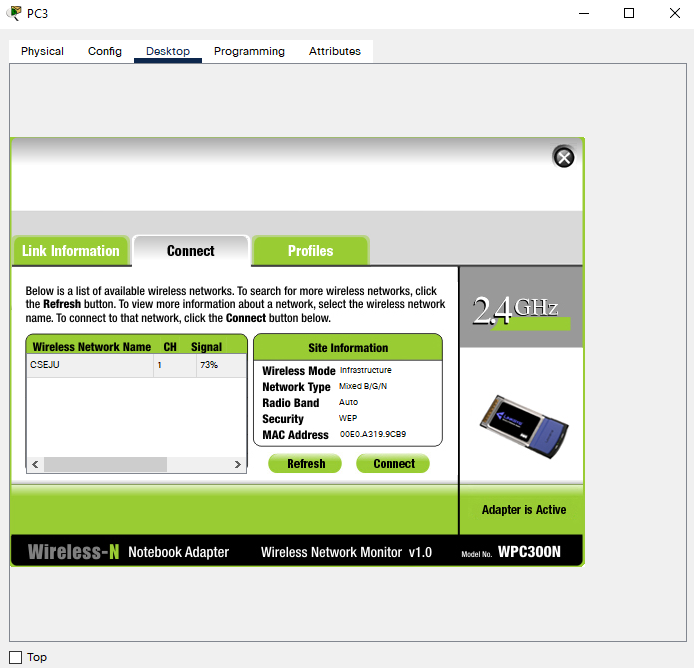


Figure-6.6

**7.**Now select the wired PC0 and navigate to the option Desktop. Select DHCP and the IP address and subnet mask will be set automatically. The same procedure is followed for PC1 and PC2. It is shown in figure-6.7.

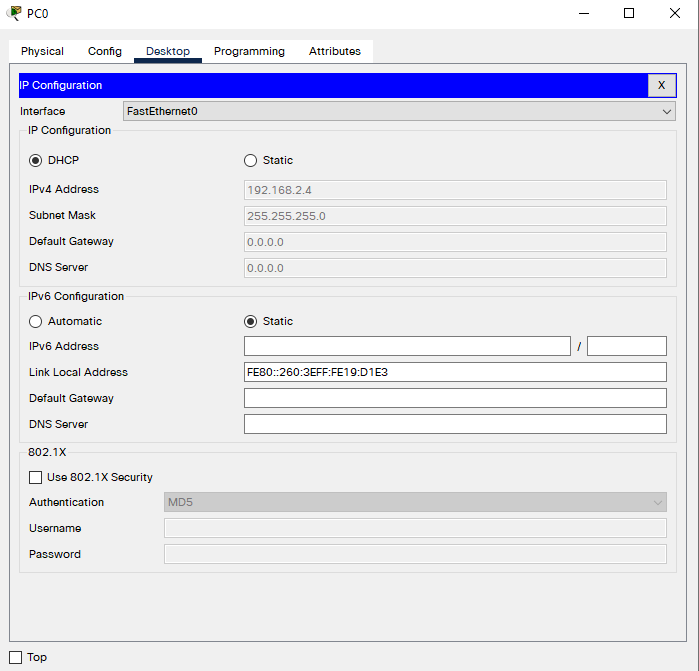


Figure-6.7

**8.** The IP addresses are:

Server0: 192.168.2.3 PC0: 192.168.2.4

PC1: 192.168.2.5 PC2: 192.168.2.6

PC3: 192.168.2.7 PC4: 192.168.2.8

**9.** Sent a simple PDU from PC0 to PC3 in simulation mode. The packe sent is successful as shown in figure-6.8.

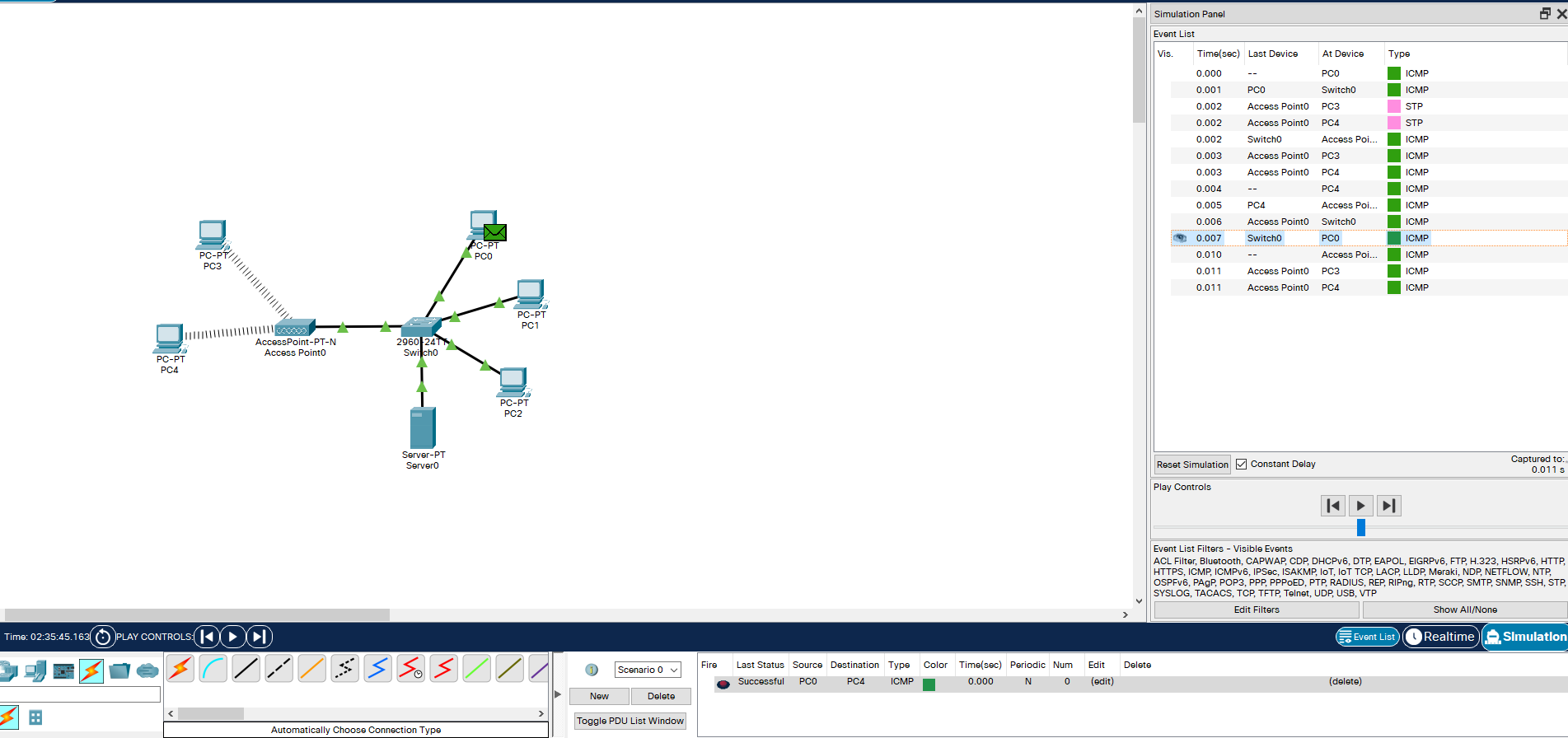


Figure-6.8

**10.** Now from PC3 open command prom tans sent a ping request to PC2. The ping request is successful as shown in figure-.

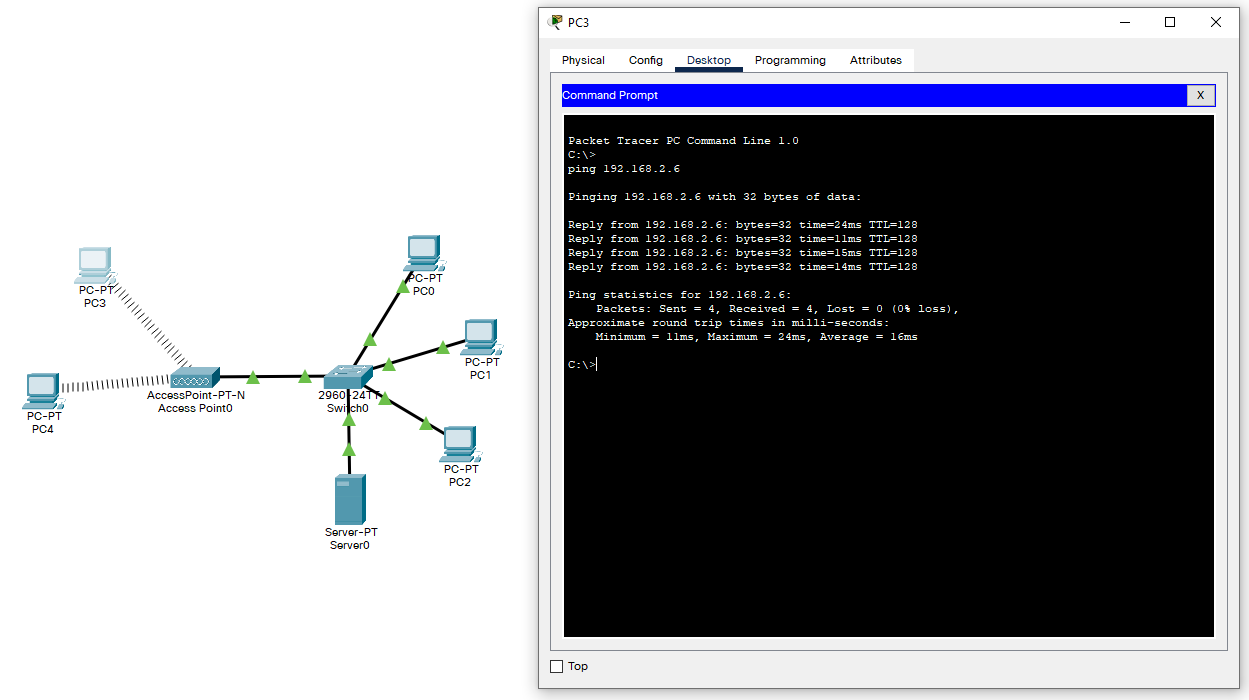


Figure-6.9

11. Now connect another new switch with the existing switch and connect two new PCs with the new switch as shown in figure- . Configure the IP address of the new PCs in similar way done before. The IP address of the new PC5 and PC6 are 192.168.2.9 and 192.168.2.10 respectively.

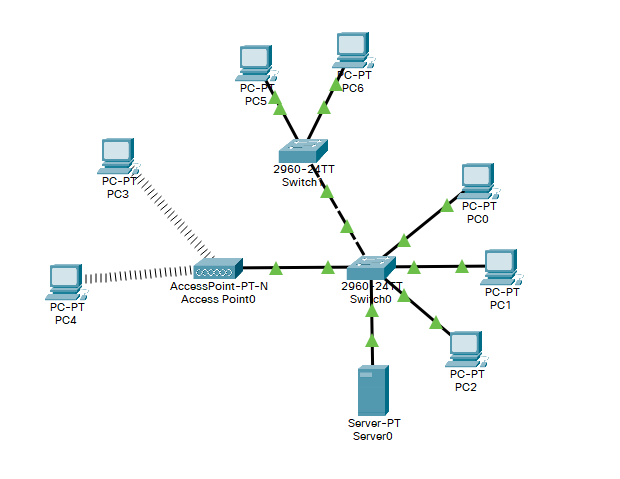


Figure-6.10

12.Add two new html pages in the server: index.html and profile.html. The html pages can be accessed from all the PCs connected to the network as shown in figure- and figure- .

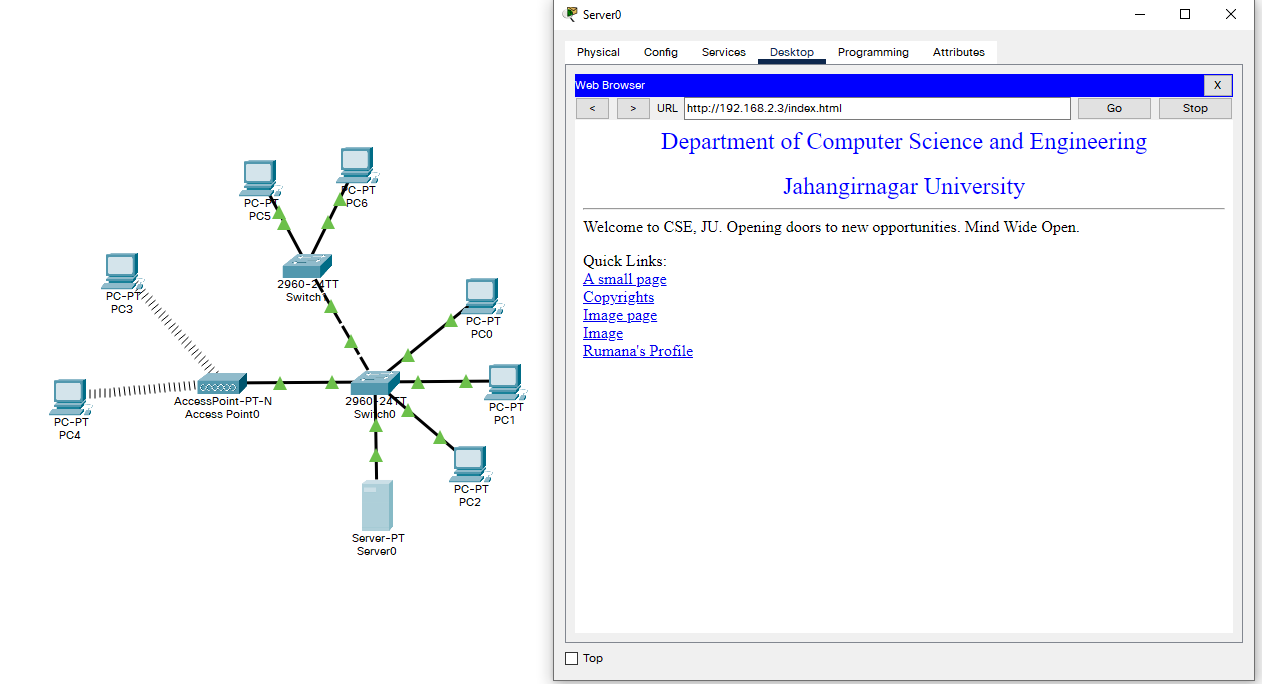


Figure-6.11

**Discussion:** A wireless LAN using wireless access point and a DHCP server using CISCO Packet Tracer software is implemented successfully.

**Experiment-07**

**Title**: Multiple Input Multiple Output (MIMO) System.

**Objective:**

Laboratory works help us for the better understanding of the theoretical studies in practical form. In this we convert one-to-one relation into many-to-many relation between transmitter and receiver in wireless communication. A channel may be affected by fading and this will impact the signal to noise ratio. MIMO or Multiple-Input Multiple-Output can be referred to as the communication channel created with multiple transmitters and receivers of an antenna to improve communication’s performance.

**Theory:**

* In wireless communication systems, the Multiple Input Multiple Output(MIMO) system provides benefits of spatial diversity with additional channel capacity with out increasing the required bandwidth of the communication system.
* A space diversity technique employs multiple transmitor receive antennas with the spacing between adjacent antennas. Usually, the separation between two adjacent antenna elements is kept half of the wavelength or slightly greater.
* To alleviate different types of fading in wireless communication system, spatial diversity and spatial multiplexing techniques are widely used and in corporation of MIMO is a simpler solution.

**MATLAB code:**

1. x=[0.2+0.4i; 1.1-0.6i; 0.45-0.34i; 1.2+1.4i];
2. n=[0.001+0.02i; 0.003-0.005i; 0.04-0.003i; 0.0012-0.003i];
3. H=[0.23-0.12i -0.612+0.09i -0.71+0.12i 0.32+0.11i;
4. 0.112-0.098i 0.16+0.23i 0.154-0.22i 0.32-0.23i;
5. -0.53-0.12i 0.321-0.25i 0.56-0.076i 0.71-0.22i;
6. 0.86-0.23i -0.887-0.099i 0.23+.76i 0.45-0.42i];
7. r=H\*x+n; %Received signal vector
8. Q1=H\*ctranspose(H); %HHH matrix
9. [U, D1]=eigs(Q1);
10. D=sqrt(D1);
11. np= ctranspose(U)\*n; %n΄=UHn
12. xp= ctranspose(U)\*x; % x΄=UHx;
13. rp= D\*xp +np;
14. nT=4;
15. H4=[0.23-0.12i -0.612+0.09i -0.71+0.12i 0.32+0.11i;
16. 0.112-0.098i 0.16+0.23i 0.154-0.22i 0.32-0.23i;
17. -0.53-0.12i 0.321-0.25i 0.56-0.076i 0.71-0.22i;
18. 0.86-0.23i -0.887-0.099i 0.23+.76i 0.45-0.42i];
19. Q4 = H4\*ctranspose(H4);
20. **for** i=1:10,
21. SNR(i)=1+3\*i; Im4 = [1 0 0 0; 0 1 0 0; 0 0 1 0;0 0 0 1];
22. CW4(i) = log2(det(Im4+(Q4/nT)\*SNR(i)));
23. CW(i) = log2(1+SNR(i));
24. end
25. plot(SNR,CW4, 'ks:', SNR,CW, 'rd:');
26. hold on
27. xlabel('SNR')
28. ylabel('Channel Capacity/Hz')
29. title('Channel Capacity of MIMO Link')
30. grid on
31. legend('MIMO', 'Single Antenna')

**Result:**



**Discussion:**

As a Result: of the use multiple antennas, MIMO wireless technology is able to considerably increase the capacity of a given channel while still obeying Shannon's law. By increasing the number of receive and transmit antennas it is possible to linearly increase the throughput of the channel with every pair of antennas added to the system.

**Experiment-08**

**Title**: Connection of Cisco Router and Switch to ISP Home Router and Access Internet.

**Objective:**

Laboratory works help us for the better understanding of the theoretical studies in practical form. In this we connect a router and switch to implement ISP home router and access internet.

**Theory:**

When building a small office network, the two most essential pieces of equipment you will need are [switches](https://www.cisco.com/c/en/us/solutions/small-business/networking/switches.html) and [routers](https://www.cisco.com/c/en/us/solutions/small-business/networking/routers.html). Though they look similar, the two devices perform different functions within a network.

* **Switch**

Switches facilitate the sharing of resources by connecting together all the devices, including computers, printers, and servers, in a small business network. Thanks to the switch, these connected devices can share information and talk to each other, regardless of where they are in a building or on a campus. Building a small business network is not possible without switches to tie devices together.

* **Router**

Just as a switch connects multiple devices to create a network, a router connects multiple switches, and their respective networks, to form an even larger network. These networks may be in a single location or across multiple locations. When building a small business network, you will need one or more routers. In addition to connecting multiple networks together, the router also allows networked devices and multiple users to access the Internet.

Ultimately, a router works as a dispatcher, directing traffic and choosing the most efficient route for information, in the form of data packets, to travel across a network. A router connects your business to the world, protects information from security threats, and even decides which devices have priority over others.

**Software used:** Cisco Packet Tracer

**Working Procedure:**

1. Open Cisco Packet Tracer software.

2. Connect six PCs with 2960-24TT switch and 1842 router as shown in figure-8.1.

3. Then open command line interface of the switch and write the following commands. The output are shown in figures-8.2, 8.3, 8.4 and 8.5.

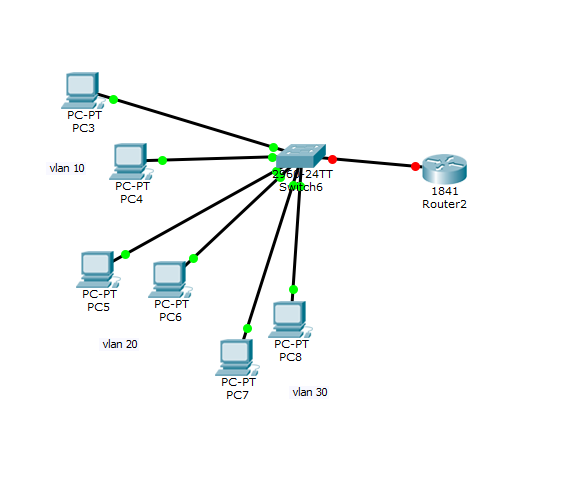


Figure-8.1

Switch>en

Switch#vlan database

Switch(vlan)#vlan 10 name A

Switch(vlan)#vlan 20 name B

Switch(vlan)#vlan 30 name C

Switch(vlan)#exit

Switch#conf t

Switch(config)#int range fa0/1-6

Switch(config-if-range)#switchport mode access

Switch(config-if-range)#switchport access vlan 10

Switch(config-if-range)#exit

Switch(config)#int range fa0/7-14

Switch(config-if-range)#switchport mode access

Switch(config-if-range)#switchport access vlan 20

Switch(config-if-range)#exit

Switch(config)#int range fa0/15-23

Switch(config-if-range)#switchport mode access

Switch(config-if-range)#switchport access vlan 30

Switch(config-if-range)#exit

Switch(config)#int fa0/24

Switch(config-if)#switchport mode trunk

Switch(config-if)#end

Switch#shvlan brief

Switch#conf t

Switch(config)#intvlan 10

Switch(config-if)#ip add 192.168.2.2 255.255.255.0

Switch(config-if)#no shut

Switch(config-if)#exit

Switch(config)#intvlan 20

Switch(config-if)#ip add 192.168.3.2 255.255.255.0

Switch(config-if)#no shut

Switch(config-if)#exit

Switch(config)#intvlan 30

Switch(config-if)#ip add 192.168.4.2 255.255.255.0

Switch(config-if)#no shut

Switch(config-if)#exit

Switch(config)#ipdhcp pool 10

Switch(dhcp-config)#network 192.168.2.0 255.255.255.0

Switch(dhcp-config)#default-router 192.168.2.1

Switch(dhcp-config)#exit

Switch(config)#ipdhcp pool 20

Switch(dhcp-config)#network 192.168.3.0 255.255.255.0

Switch(dhcp-config)#default-router 192.168.3.1

Switch(dhcp-config)#exit

Switch(config)#ipdhcp pool 30

Switch(dhcp-config)#network 192.168.4.0 255.255.255.0

Switch(dhcp-config)#default-router 192.168.4.1

Switch(dhcp-config)#exit

Switch(config)#int fa0/24

Switch(config-if)#switchport mode trunk

Switch(config-if)#intvlan 1

Switch(config-if)#ip add 192.168.100.2 255.255.255.0

Switch(config-if)#no shut

Switch(config-if)#exit

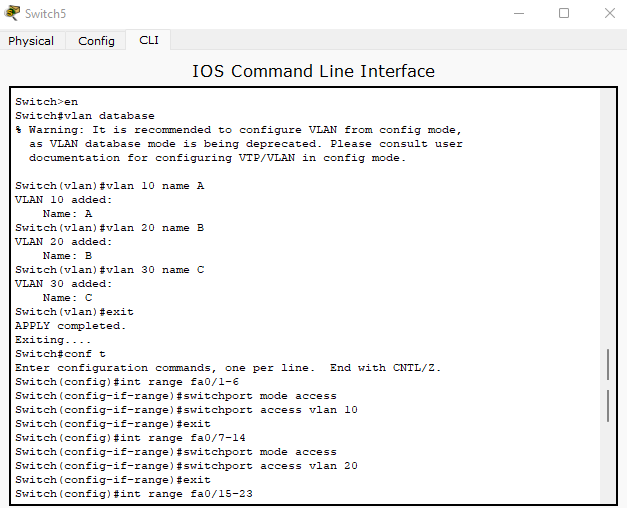


Figure-8.2

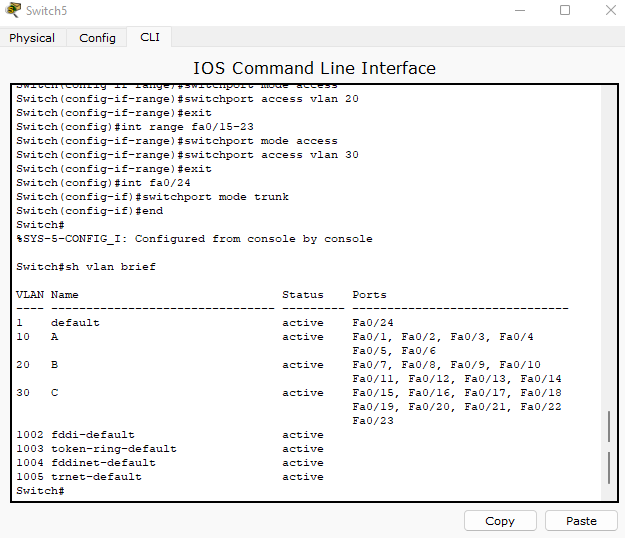


Figure-8.3

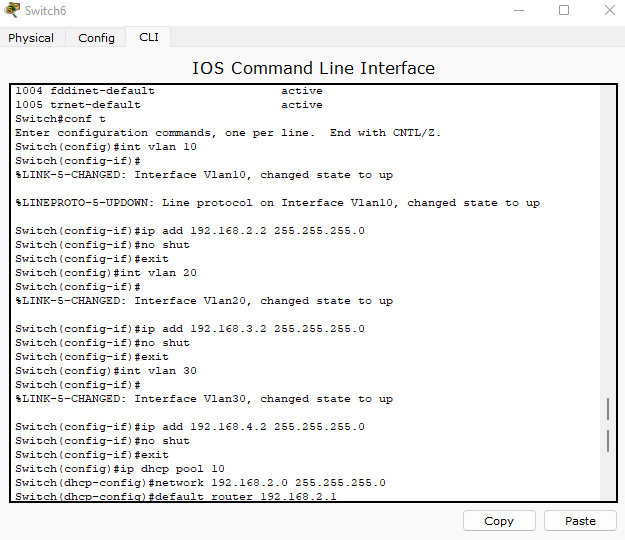


Figure-8.4

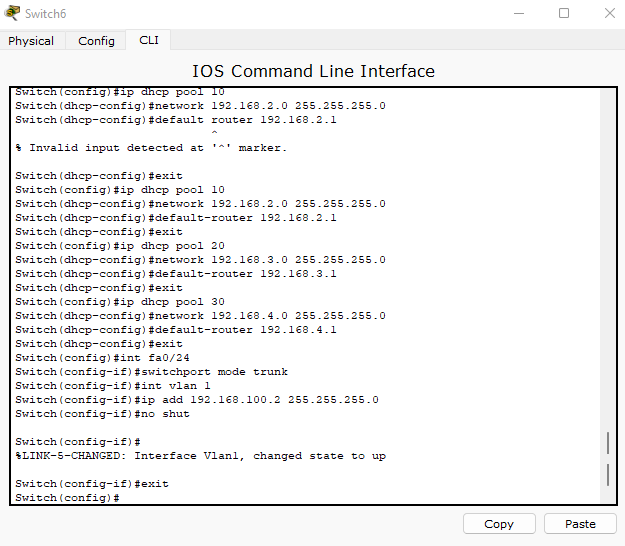


Figure-8.5

**Discussion:**

We have implemented this successfully. In this experiment, we connected a router and switch to implement ISP home router and access internet.