CSE 3501

INFORMATION SECURITY ANALYSIS & AUDIT

Lab Assessment – 1

L9+L10 | PLBG04

FALL SEMESTER 2021-22

by

SHARADINDU ADHIKARI

19BCE2105

Exercise 1 .

1. **Aim:** To implement either a symmetric or an asymmetric algorithm.

2. **Objective:** After some research and studying the algos briefly, I've decided to go forward with the objective of implementing an asymmetric algorithm: RSA. In C++.

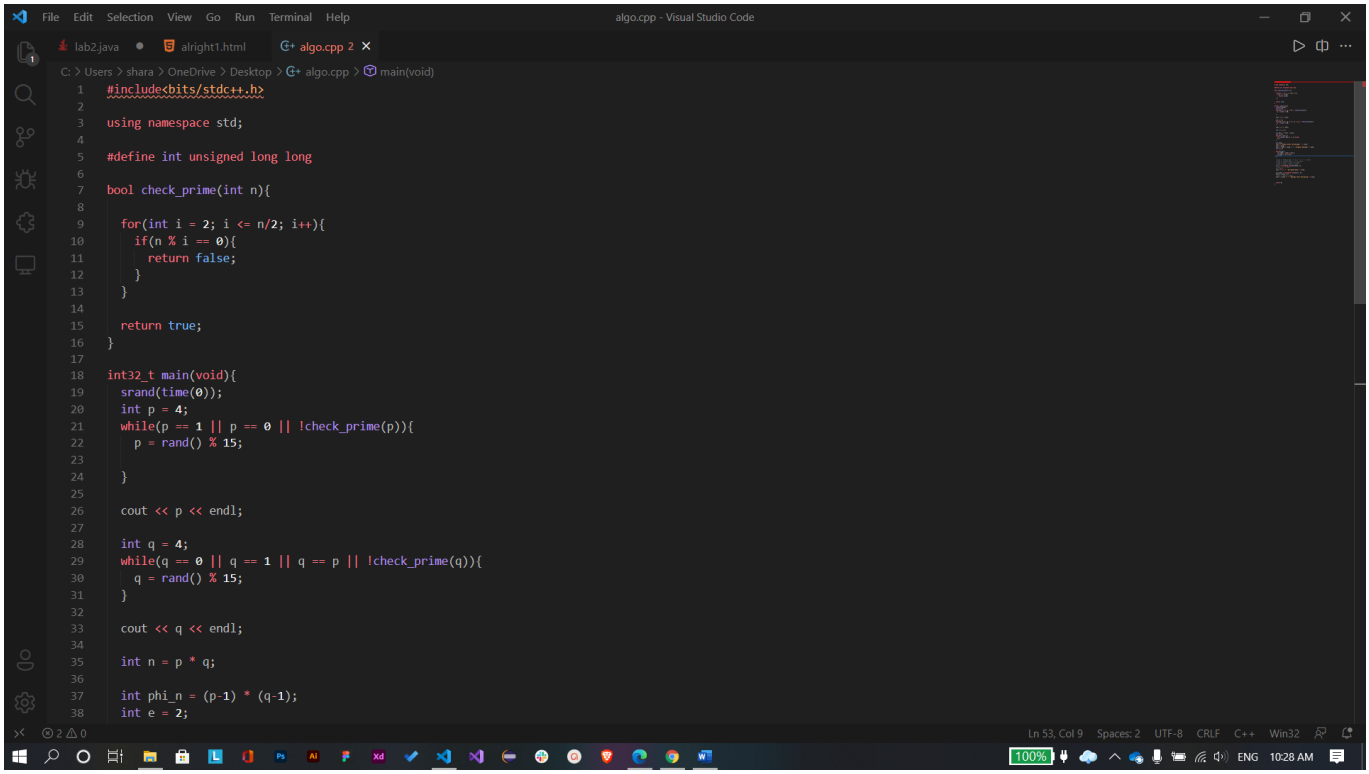
3. **Procedure:**
We know right, RSA Algorithm is an example for Public Key Encryption algorithm; so here we are supposed to find two keys:
1) Public Key which is used at encryption, & 2) Private Key which is used at decryption.

Step 1: Select two large Primes P, Q
Step 2: Calculate n=P*Q & O(n) = (P-1)*(Q-1)
Step 3: Assume e and d (Public and Private Key).
Step 4: Encrypt the Plain Text using Public Key e.
Step 5: Decrypt the Cipher Text using Private Key d.

4. **Code:**

```
#include<bits/stdc++.h>
using namespace std;
#define int unsigned long long

bool check_prime(int n){

  for(int i = 2; i <= n/2; i++){
    if(n % i == 0){
      return false;
    }
  }
  return true;
}

int32_t main(void){
  srand(time(0));
  int p = 4;
  while(p == 1 || p == 0 || !check_prime(p)){
    p = rand() % 15;
  }

  cout << p << endl;
  int q = 4;
  while(q == 0 || q == 1 || q == p || !check_prime(q)){
    q = rand() % 15;
  }

  cout << q << endl;
  int n = p * q;
  int phi_n = (p-1) * (q-1);
  int e = 2;
  while(e < phi_n){
```

```
    if(__gcd(e, phi_n) == 1) break;
    e++;
  }

  int mssg;
  cout << "Please enter the message: " << endl;
  cin >> mssg;
  cout << endl << mssg << " - Original Message" << endl;
  int d = 1;

  while(true){
    int temp = (d*e) % phi_n;
    if(temp == 1) break;
    d++;
  }

  //cout << "Public Key " << e << " " << n << endl;
  //cout << "Private Key " << d << endl;
  //cout << "Phi " << phi_n << endl;
  //cout << (d*e) % phi_n << endl;

int c = (unsigned long)pow(mssg, e);
  //cout << c << endl;
  c = c % n;
  cout << c << " - Encrypted Data" << endl;

  int plain = (unsigned long)pow(c, d);
  plain = plain % n;
  //cout << plain << endl;
  cout << plain << " - Message after Decrypting" << endl;
  return 0;
}
```



```cpp
int phi_n = (p-1) * (q-1);
int e = 2;
while(e < phi_n){
  if(__gcd(e, phi_n) == 1) break;
  e++;
}

int mssg;
cout << "Please enter the message: " << endl;
cin >> mssg;
cout << endl << mssg << " - Original Message" << endl;
int d = 1;

while(true){
  int temp = (d*e) % phi_n;
  if(temp == 1) break;
  d++;
}

//cout << "Public Key " << e << " " << n << endl;
//cout << "Private Key " << d << endl;
//cout << "Phi " << phi_n << endl;
//cout << (d*e) % phi_n << endl;
int c = (unsigned long)pow(mssg, e);
//cout << c << endl;
c = c % n;
cout << c << " - Encrypted Data" << endl;

int plain = (unsigned long)pow(c, d);
plain = plain % n;
//cout << plain << endl;
cout << plain << " - Message after Decrypting" << endl;



return 0;
}
```

5. Results screenshot:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\shara> cd onedrive
PS C:\Users\shara\onedrive> cd desktop
PS C:\Users\shara\onedrive\desktop> g++ -o algo algo.cpp
PS C:\Users\shara\onedrive\desktop> ./algo
2
7
Please enter the message:
6

6 - Original Message
6 - Encrypted Data
6 - Message after Decrypting
PS C:\Users\shara\onedrive\desktop>
```

Exercise 2 .

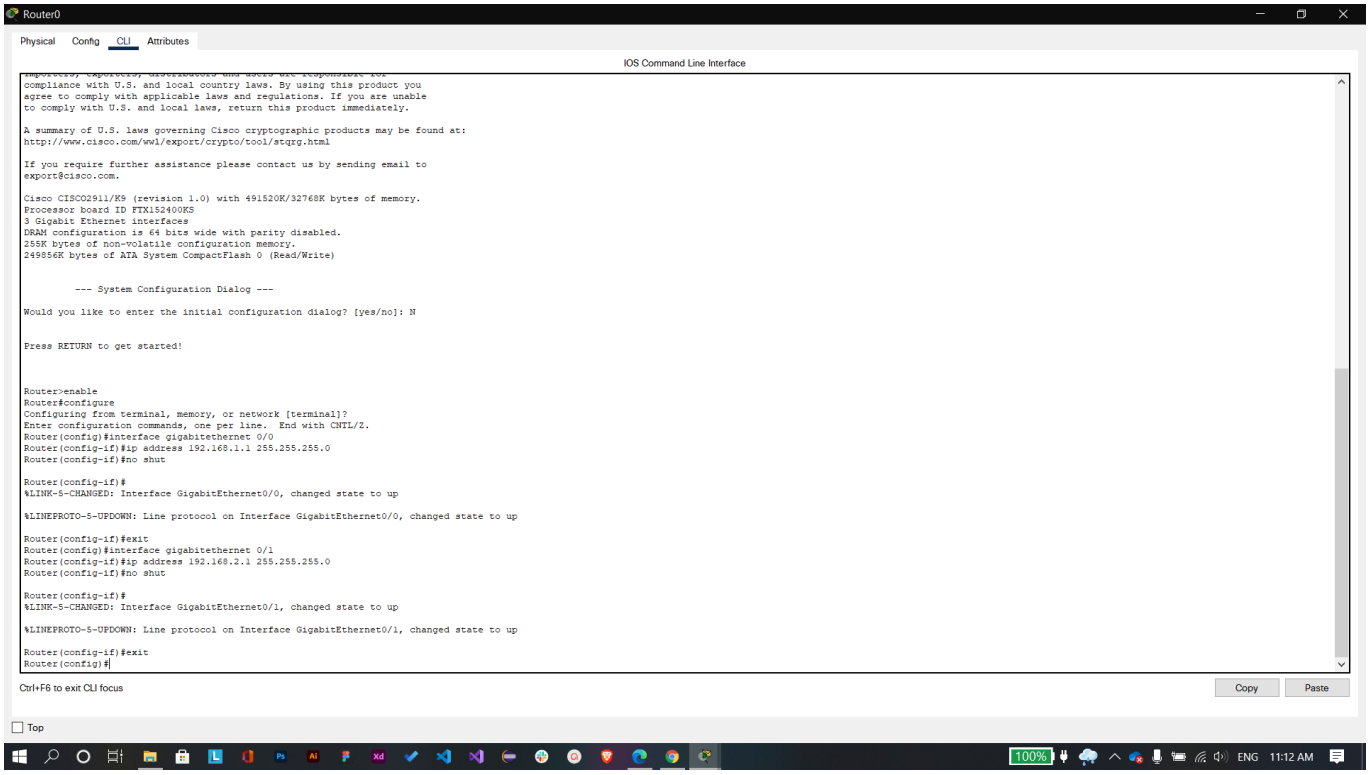
1. **Aim:** To connect two different LANs using Router configuration.

2. **Objective:** My objective in this exercise is to learn how PDUs (messages, signals, etc.) travel from one PC to another within the LAN network, as well as between different LAN networks using Packet Tracer® simulation.

3. **Procedure:**

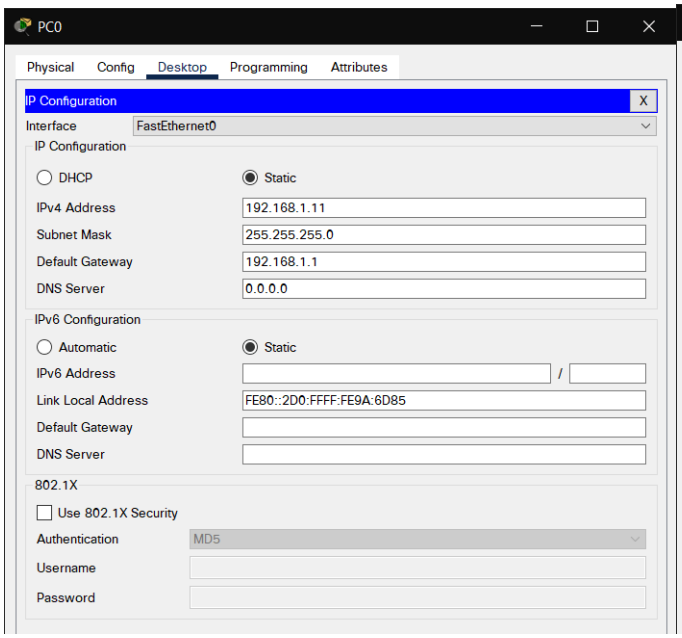
   - Select a 2911 router, followed by two 2960 switches and 3 PCs.
   - Select a straight-through cable.
   - Using it, connect router0 and switch0 (from gigabitethernet 0/0 port of former to the gigabitethernet 0/1 port of the latter).
   - Similarly, connect router0 and switch1 (gigabitethernet 0/1 to gigabitethernet 0/1).
   - After all these connections are wired up, it's time to configure the router. Go to router's CLI and type in the following commands:

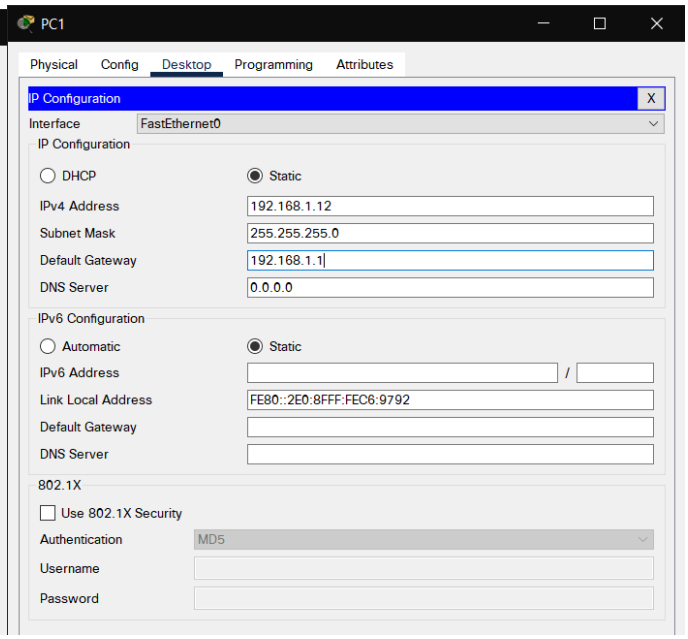
```
Would you like to enter the initial configuration dialog? [yes/no]: N

Press RETURN to get started!

Router>enable
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface gigabitethernet 0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state
to up

Router(config-if)#exit
Router(config)#interface gigabitethernet 0/1
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state
to up

Router(config-if)#exit

Router(config)#

Router con0 is now available

Press RETURN to get started.
```

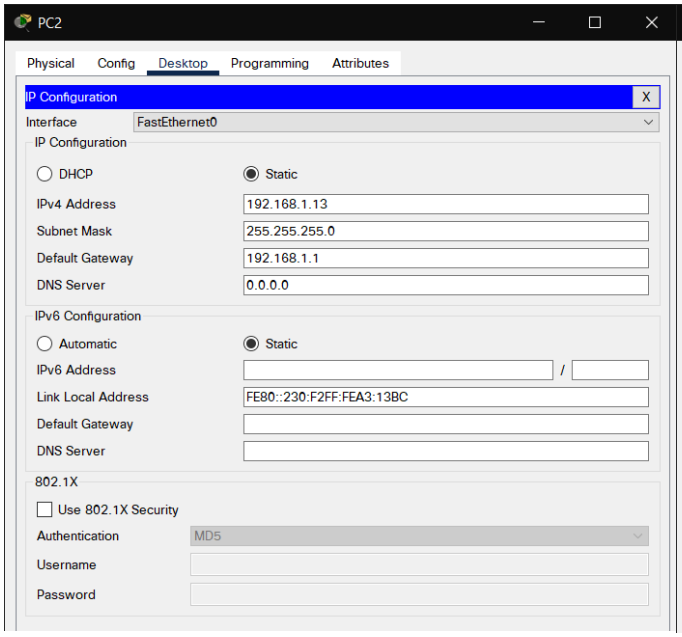


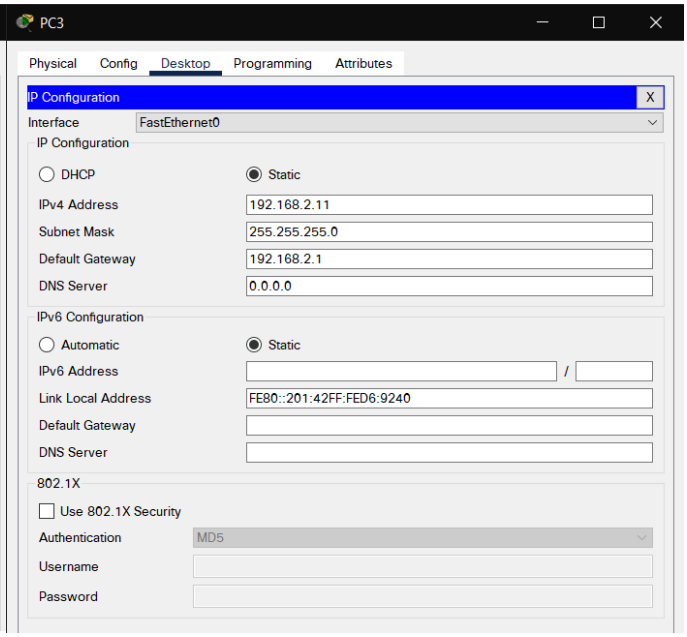
- After router0 has been configured, it's time for the PCs:

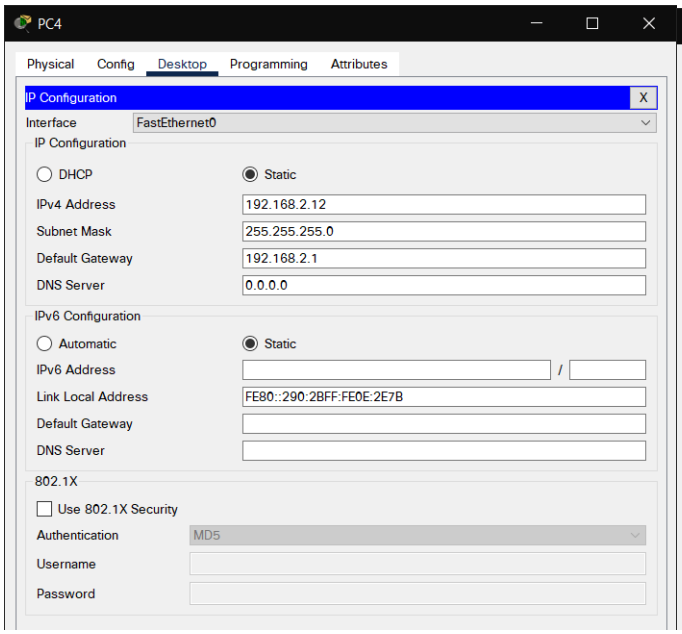
  Give addresses for PCs in Network1 as: 192.168.1.11 to 192.168.1.13
  Give addresses for PCs in Network1 as: 192.168.2.11 to 192.168.2.13
  Set default gateway for the leftside network as 192.168.1.1 (as similar to router gigabitethernet 0/0)
  Set default gateway for the rightside network as 192.168.2.1 (as similar to router gigabitethernet 0/1)

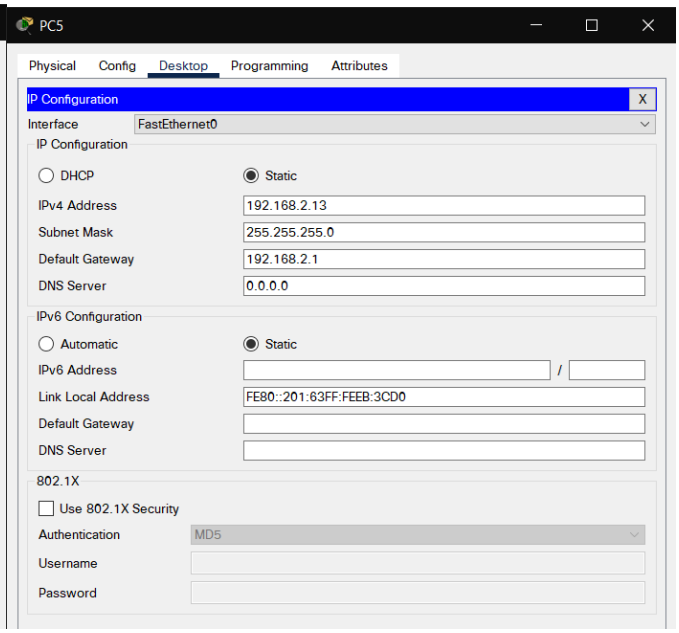




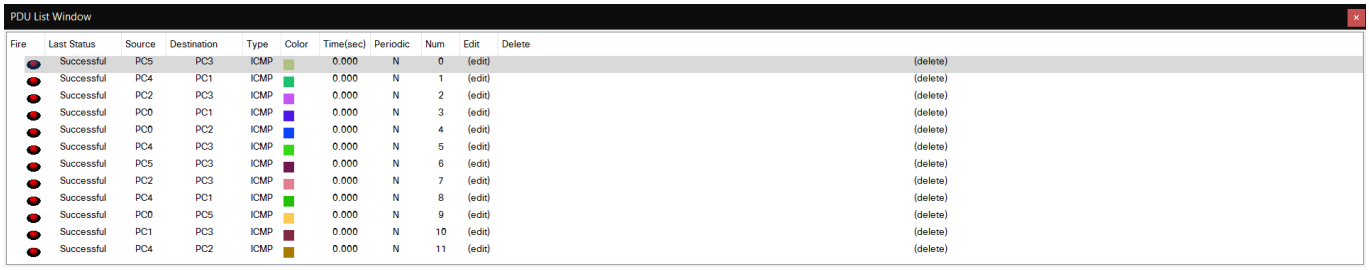






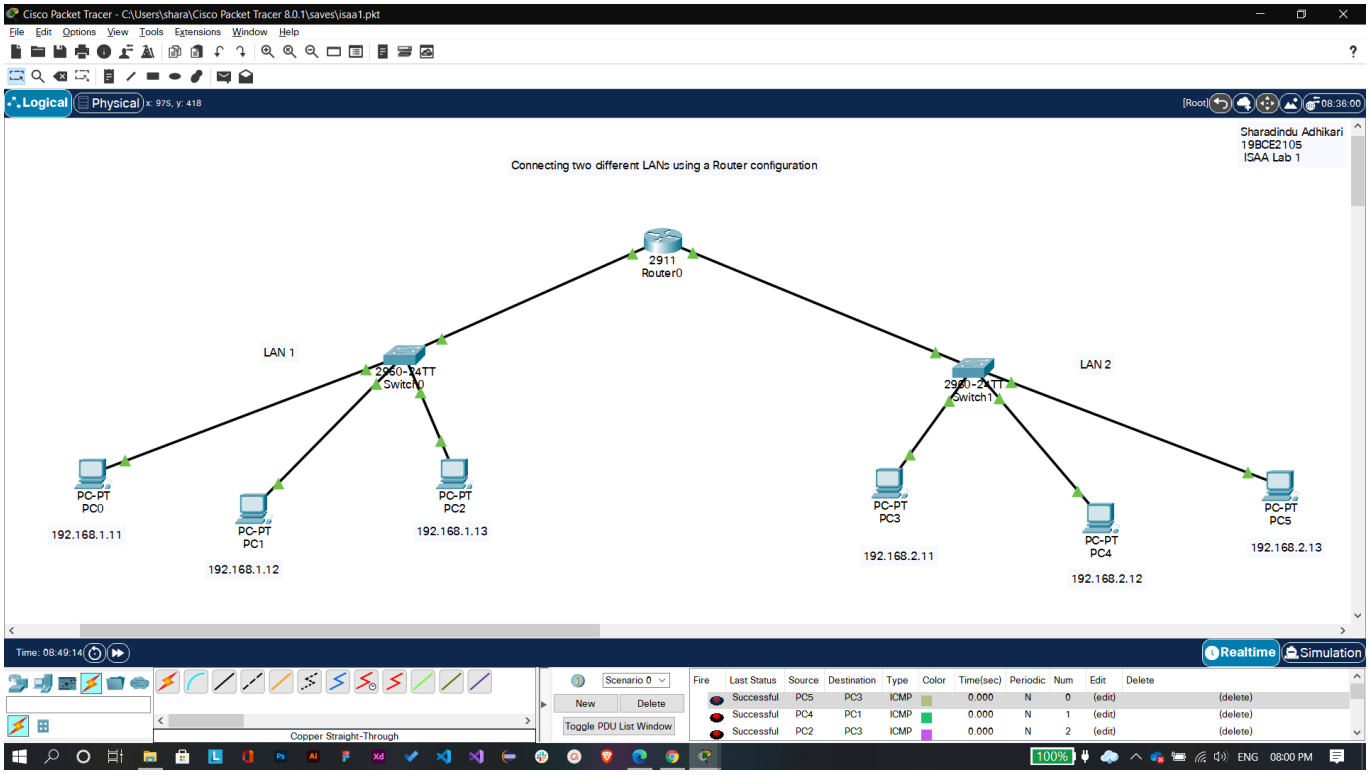


4. Results screenshots:



PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC5	PC3	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC4	PC1	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC2	PC3	ICMP		0.000	N	2	(edit)	(delete)
	Successful	PC0	PC1	ICMP		0.000	N	3	(edit)	(delete)
	Successful	PC0	PC2	ICMP		0.000	N	4	(edit)	(delete)
	Successful	PC4	PC3	ICMP		0.000	N	5	(edit)	(delete)
	Successful	PC5	PC3	ICMP		0.000	N	6	(edit)	(delete)
	Successful	PC2	PC3	ICMP		0.000	N	7	(edit)	(delete)
	Successful	PC4	PC1	ICMP		0.000	N	8	(edit)	(delete)
	Successful	PC0	PC5	ICMP		0.000	N	9	(edit)	(delete)
	Successful	PC1	PC3	ICMP		0.000	N	10	(edit)	(delete)
	Successful	PC4	PC2	ICMP		0.000	N	11	(edit)	(delete)



5. Observation: