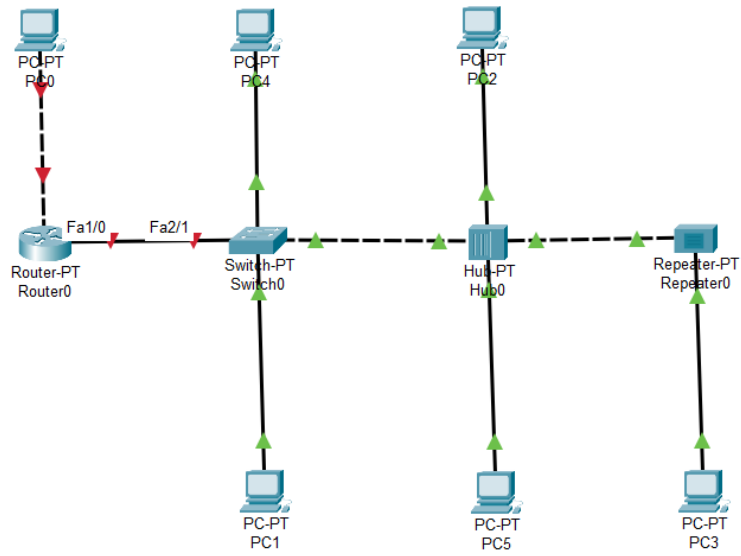
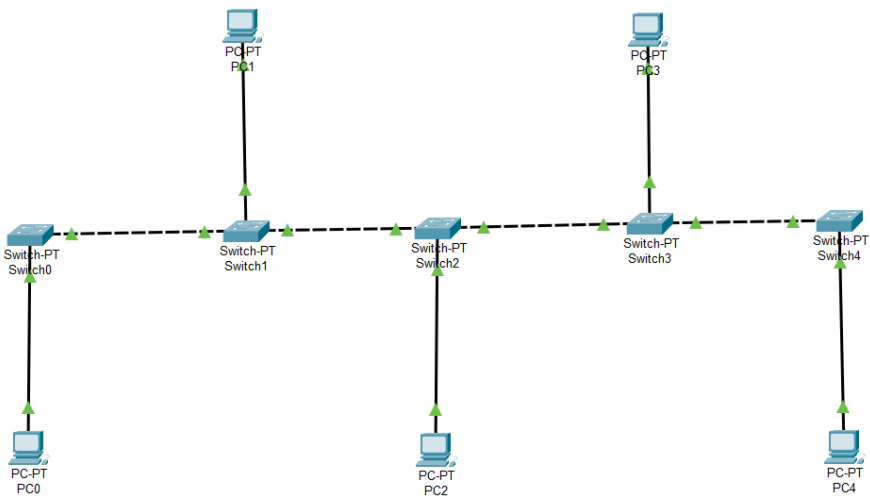


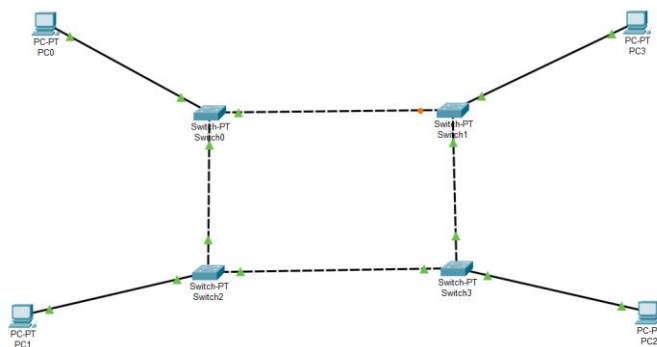
## Network devices



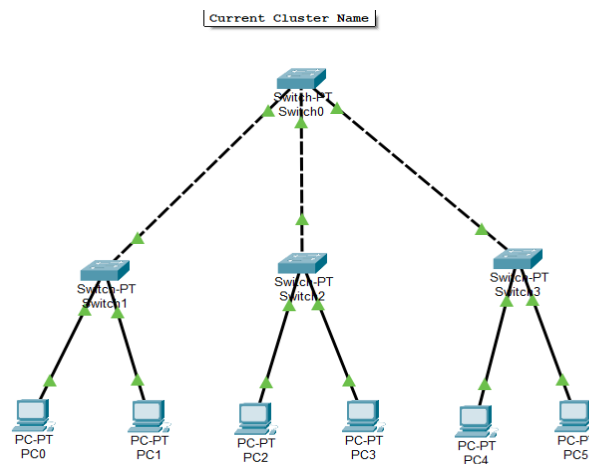
## Bus topology



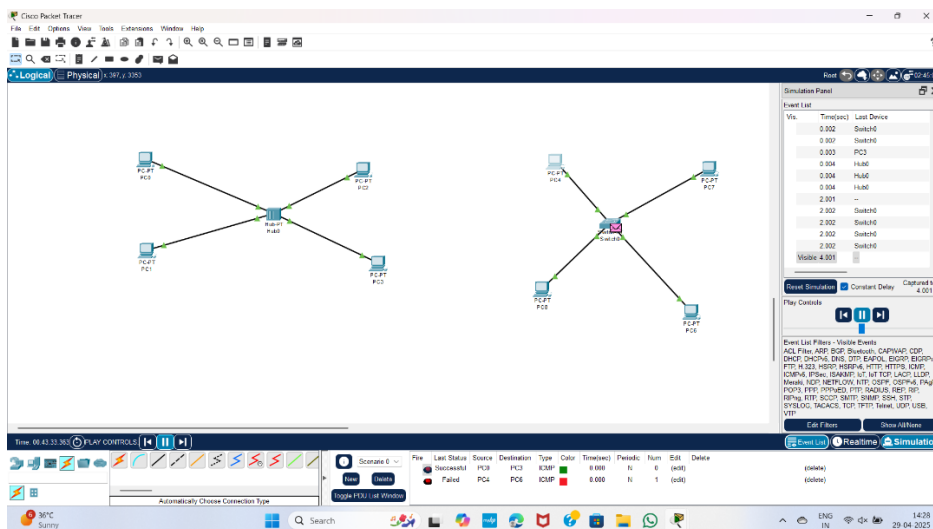
## Ring topology



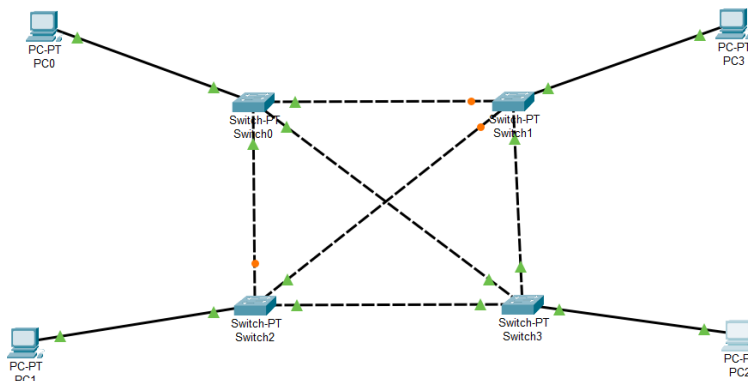
## Tree topology



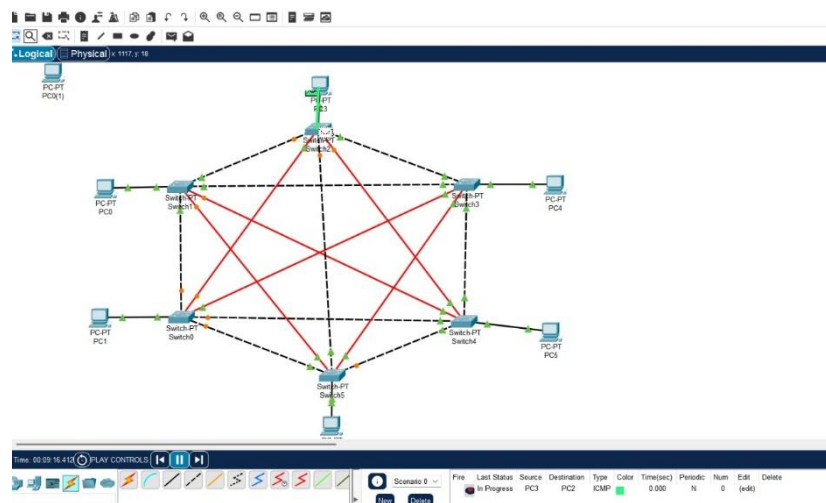
## Star topology



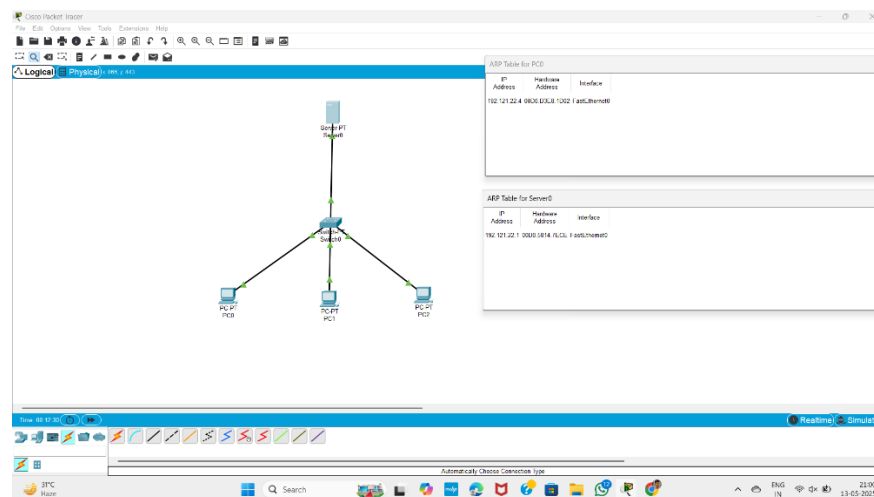
## Mesh topology



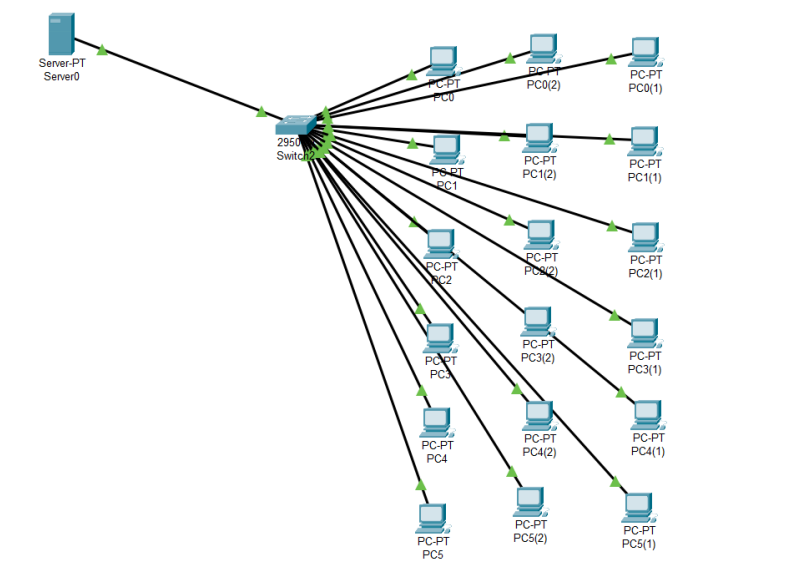
## Hybrid topology



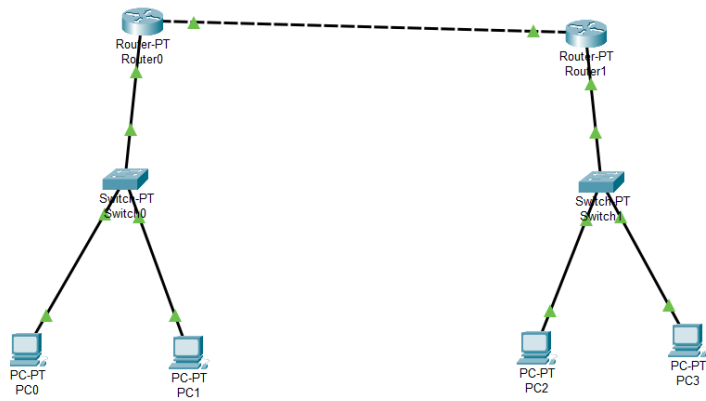
## Data link layer traffic simulation of ARP



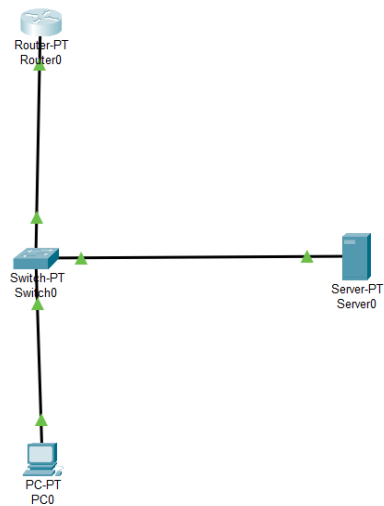
## Computer lab



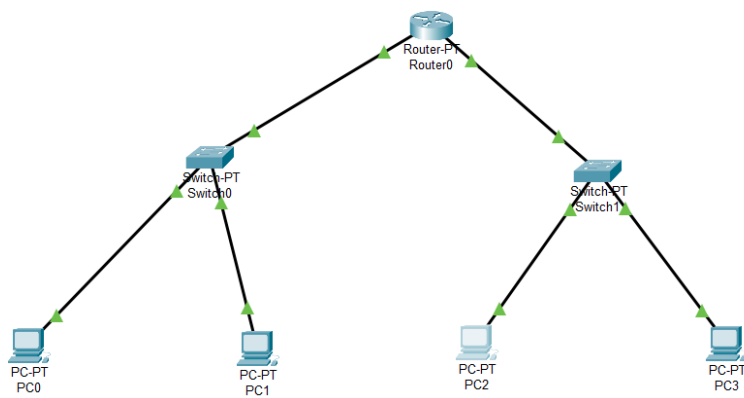
## Static router



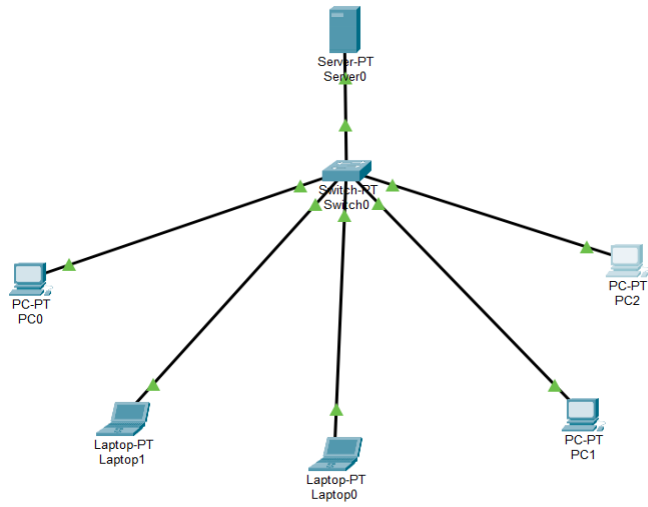
## TCP



## Subnetting

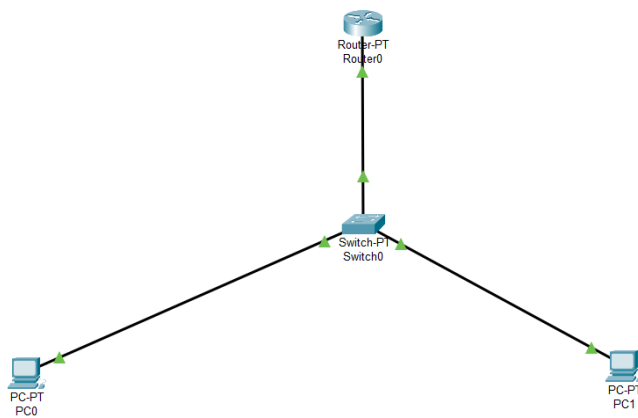


## DHCP

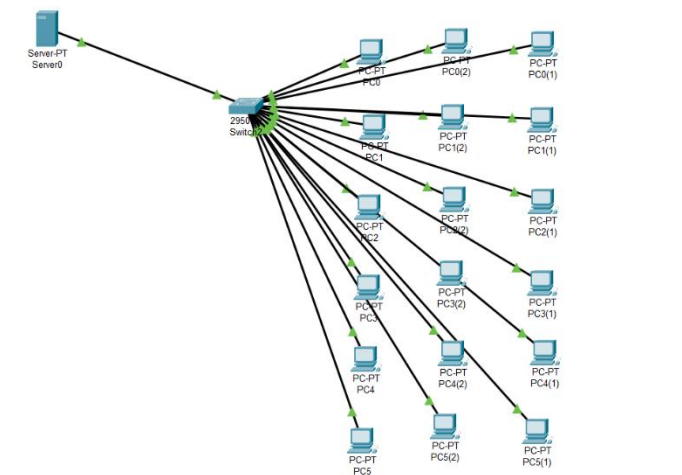


## FIREWALL

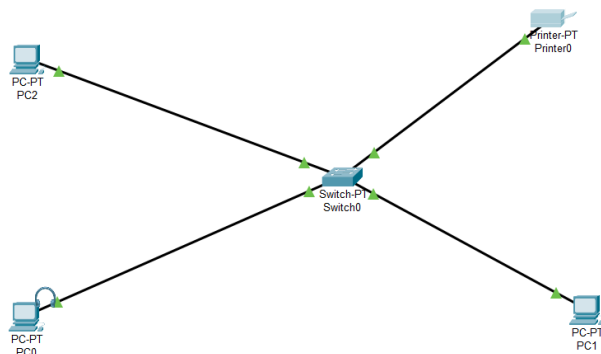
---



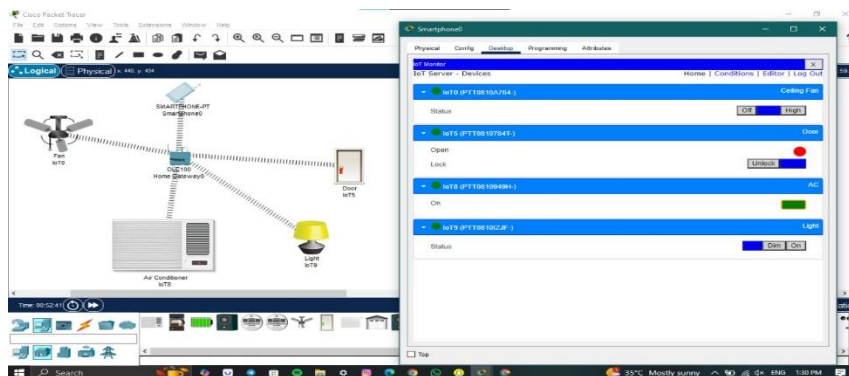
## COMPUTER LAB



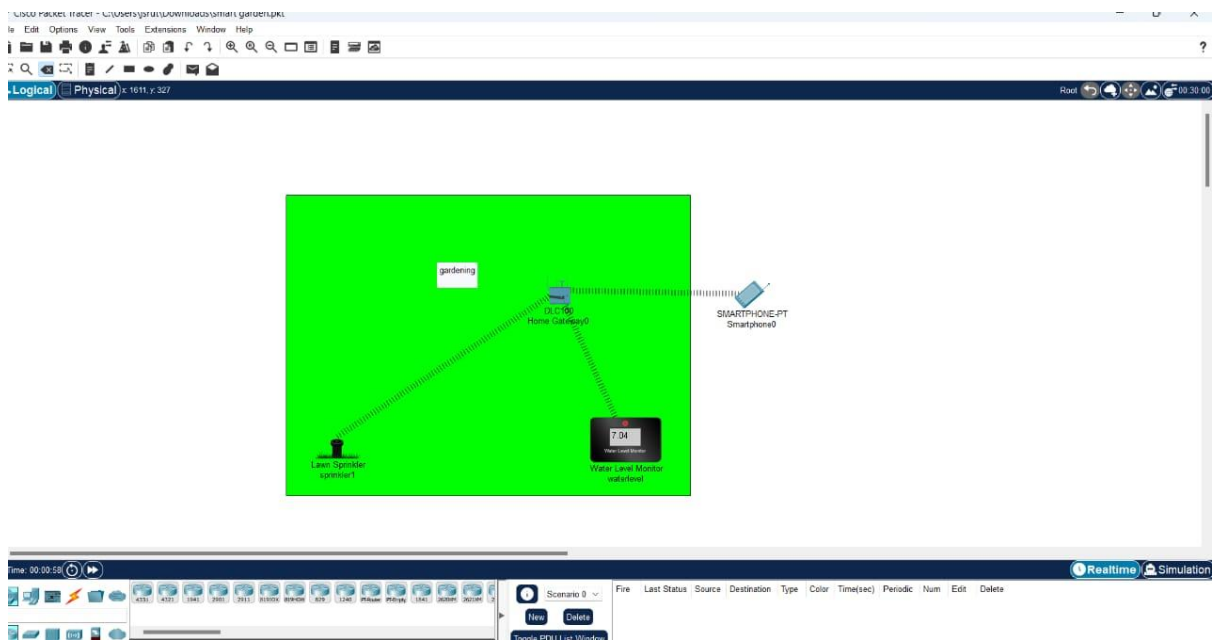
## MULTIMEDIA



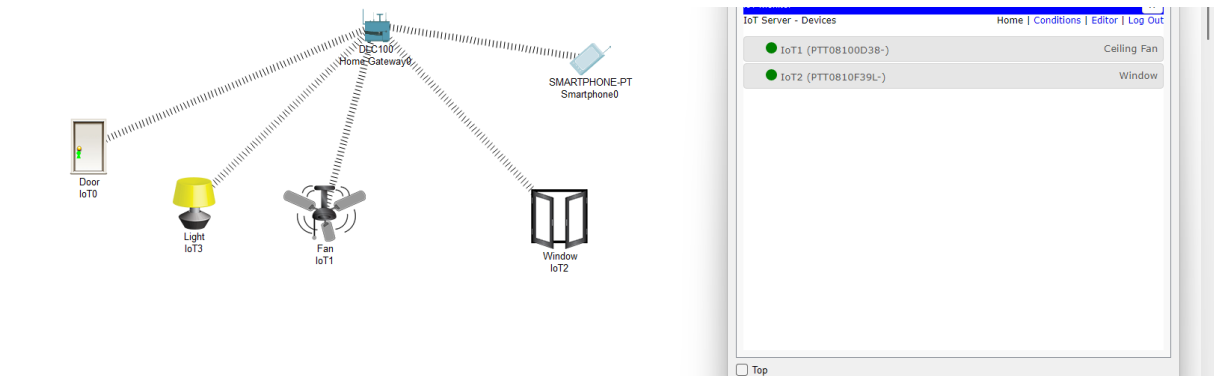
## SMART HOME



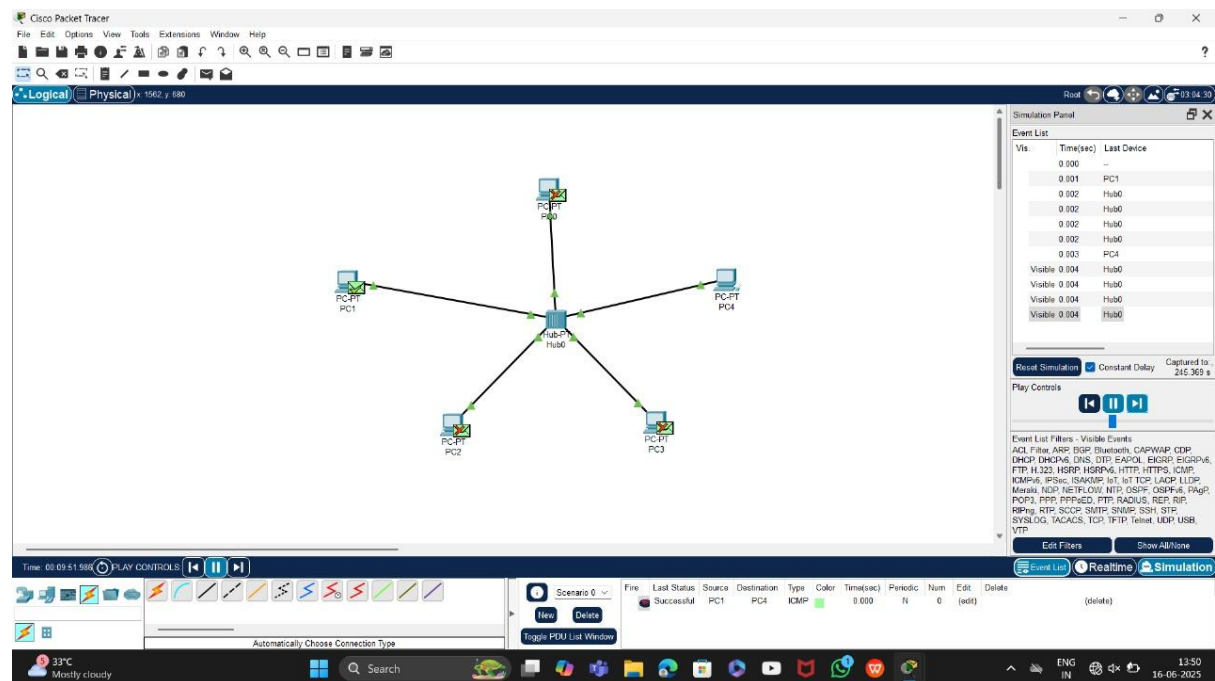
## SMART GARDEN



## SMART NETWORK DEVICES



## CSMA/CD



# HTTP

The image shows a Wireshark capture of an HTTP 400 Bad Request packet. The packet list pane shows a packet of length 558 bytes. The packet details pane shows the following structure:

- Internet Protocol Version 4, Src: 188.243.34.67, Dst: 192.168.1.7
- Transmission Control Protocol, Src Port: 26801, Dst Port: 54262, Seq: 1, Ack: 216, Len: 558
- Hypertext Transfer Protocol
  - HTTP/1.1 400 Bad Request (text/html)
  - Date: Thu, 10 Jul 2025 14:49:45 GMT
  - Server: exchange.prodigidcp.com
  - X-Frame-Options: SAMEORIGIN
  - Content-Security-Policy: script-src 'self' 'unsafe-inline' 'unsafe-eval'; object-src 'self'; worker-src 'self' blob:; r
  - Content-Length: 226
  - Connection: close
  - Content-Type: text/html; charset=iso-8859-1

The packet bytes pane shows the raw data of the packet, including the HTTP response status line and headers.

# Arp

The image shows a Wireshark capture of an ARP request packet. The packet list pane shows a packet of length 42 bytes. The packet details pane shows the following structure:

- Frame 303: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF\_{871D9A99-2B94-4843-B7A9-C5D1676B45AF}, id 0
- Ethernet II, Src: c0:35:32:7e:eb:0f (c0:35:32:7e:eb:0f), Dst: f8:0d:a9:f4:01:8b (f8:0d:a9:f4:01:8b)
- Address Resolution Protocol (request)
  - Hardware type: Ethernet (1)
  - Protocol type: IPv4 (0x0800)
  - Hardware size: 6
  - Protocol size: 4
  - Opcode: request (1)
  - Sender MAC address: c0:35:32:7e:eb:0f (c0:35:32:7e:eb:0f)
  - Sender IP address: 192.168.1.7
  - Target MAC address: f8:0d:a9:f4:01:8b (f8:0d:a9:f4:01:8b)
  - Target IP address: 192.168.1.1

The packet bytes pane shows the raw data of the packet, including the Ethernet II header and the ARP request payload.



## TCP

The image shows a Wireshark packet capture of a TCP connection. The packet list shows a sequence of packets, including a handshake and several retransmissions. The selected packet is a TCP Reset (RST) packet with the following details:

- Frame 302: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF\_{871D9A99-2B94-4843-B7A9-C5D1676B45AF}, id 0
- Ethernet II, Src: c0:35:32:7e:eb:0f (c0:35:32:7e:eb:0f), Dst: f8:0d:a9:f4:01:8b (f8:0d:a9:f4:01:8b)
- Internet Protocol Version 4, Src: 192.168.1.7, Dst: 141.95.160.248
- Transmission Control Protocol, Src Port: 57882, Dst Port: 6881, Seq: 0, Len: 0

The packet details show the source port as 57882 and the destination port as 6881. The sequence number is 0, and the acknowledgment number is 0. The packet length is 0 bytes.

## BIT STUFFING

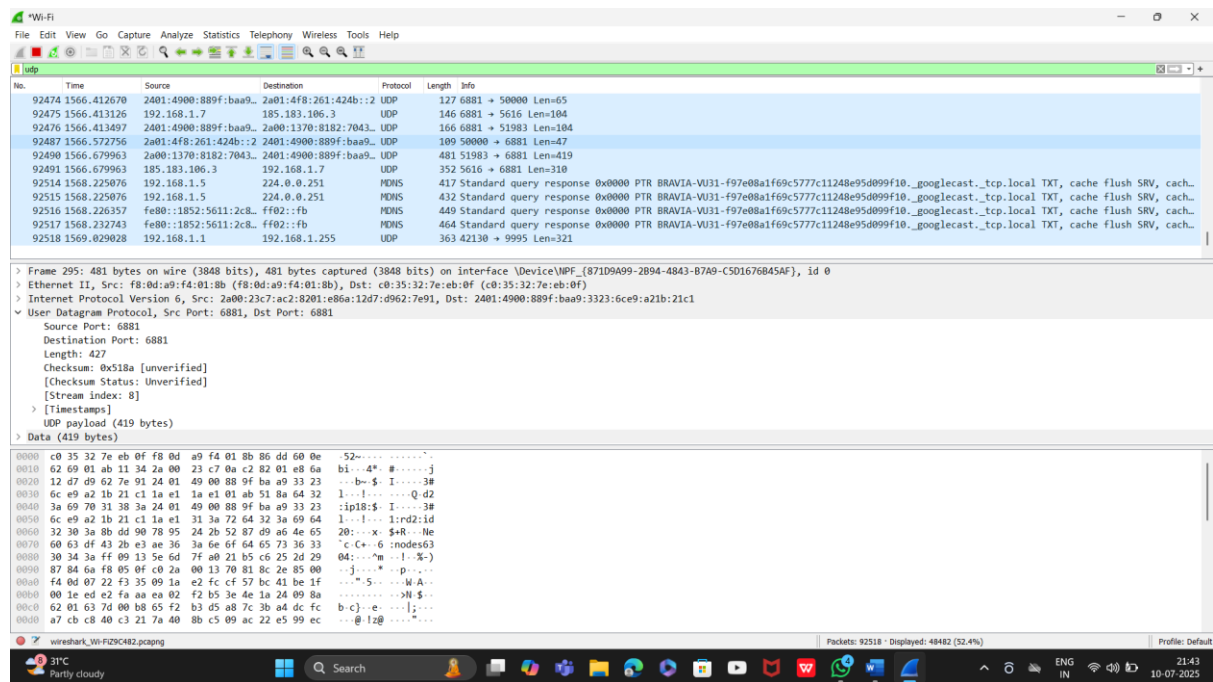
The image shows a C program named `main.c` that demonstrates bit stuffing. The program prompts the user to enter data bits and then displays the result after bit stuffing.

```
#include<stdio.h>
#include<string.h>
int main()
{
    int i=0,count=0;
    char databits[80];
    printf("Enter Data Bits: ");
    scanf("%s",databits);
    printf("\nData Bits After Bit stuffing: ");
    for(i=0; i<strlen(databits); i++)
    {
        if(databits[i]!='1')
            count++;
        else
        {
            count=0;
            printf("%c",databits[i]);
            if(count==5)
            {
                printf("0");
                count=0;
            }
        }
    }
    return 0 ;
}
```

The output of the program is as follows:

```
Enter Data Bits: 101010111111
Data Bits After Bit stuffing: 1010101111101
=== Code Execution Successful ===
```

## UDP



The image shows a Wireshark packet capture of UDP traffic. The top pane displays a list of packets, with packet 295 selected. The middle pane shows the details of packet 295, which is a User Datagram Protocol (UDP) packet. The bottom pane shows the raw data of the packet in hexadecimal and ASCII.

**Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
92474	1566.412670	2401:4900:889f:baa9...	2a01:4f8:261:424b::2	UDP	127	6881 → 50000 Len=65
92475	1566.413126	192.168.1.7	185.183.106.3	UDP	146	6881 → 5616 Len=104
92476	1566.413497	2401:4900:889f:baa9...	2a00:1370:8182:7043...	UDP	166	6881 → 51983 Len=104
92487	1566.572756	2a01:4f8:261:424b::2	2401:4900:889f:baa9...	UDP	109	50000 → 6881 Len=47
92490	1566.679963	2a00:1370:8182:7043...	2401:4900:889f:baa9...	UDP	481	51983 → 6881 Len=419
92491	1566.679963	185.183.106.3	192.168.1.7	UDP	352	5616 → 6881 Len=319
92514	1568.225076	192.168.1.5	224.0.0.251	MDNS	417	Standard query response 0x0000 PTR BRAVIA-WJ31-f97e08a1f69c5777c11248e95d099f10_googlecast_tcp.local TXT, cache flush SRV, cach...
92515	1568.225076	192.168.1.5	224.0.0.251	MDNS	432	Standard query response 0x0000 PTR BRAVIA-WJ31-f97e08a1f69c5777c11248e95d099f10_googlecast_tcp.local TXT, cache flush SRV, cach...
92516	1568.226357	fe80::1852:5611:2c8...	ff02::fb	MDNS	449	Standard query response 0x0000 PTR BRAVIA-WJ31-f97e08a1f69c5777c11248e95d099f10_googlecast_tcp.local TXT, cache flush SRV, cach...
92517	1568.232743	fe80::1852:5611:2c8...	ff02::fb	MDNS	464	Standard query response 0x0000 PTR BRAVIA-WJ31-f97e08a1f69c5777c11248e95d099f10_googlecast_tcp.local TXT, cache flush SRV, cach...
92518	1569.029028	192.168.1.1	192.168.1.255	UDP	363	42130 → 9995 Len=321

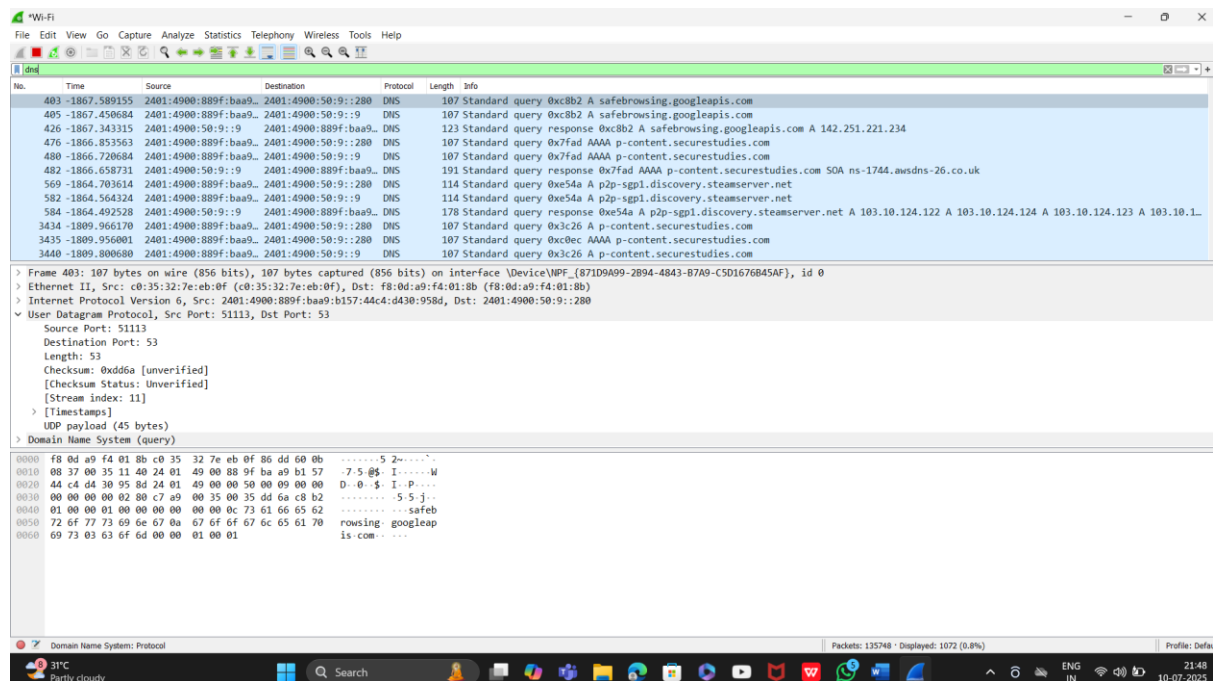
**Packet 295 Details:**

- Frame 295: 481 bytes on wire (3848 bits), 481 bytes captured (3848 bits) on interface \Device\NPF\_{871D9A99-2B94-4843-B7A9-C5D1676B45AF}, id 0
- Ethernet II, Src: f8:0d:a9:f4:01:8b (f8:0d:a9:f4:01:8b), Dst: c0:35:32:7e:eb:0f (c0:35:32:7e:eb:0f)
- Internet Protocol Version 6, Src: 2a00:23c7:ac2:8201:e06a:1d7:d962:7e91, Dst: 2401:4900:889f:baa9:3323:6ce9:a21b:21c1
- User Datagram Protocol, Src Port: 6881, Dst Port: 6881
- Source Port: 6881
- Destination Port: 6881
- Length: 427
- Checksum: 0x518a [unverified]
- [Checksum Status: Unverified]
- [Stream index: 8]
- [Timestamps]
- UDP payload (419 bytes)

**Data (419 bytes):**

```
0000 c0 35 32 7e eb 0f f8 0d a9 f4 01 8b 86 dd 60 0e 52.....
0010 62 69 01 ab 11 34 2a 00 23 c7 0a c2 82 01 e8 6a b1...4*#.....
0020 12 d7 d9 62 7e 91 24 01 49 00 88 9f ba a9 33 23 ...b-$I...3#
0030 6c e9 a2 1b 21 c1 1a e1 1a e1 01 ab 51 8a 64 32 1...f...Q.d2
0040 3a 69 70 31 38 3a 24 01 49 00 88 9f ba a9 33 23 :ip18:$I...3#
0050 6c e9 a2 1b 21 c1 1a e1 31 3a 72 64 32 3a 69 64 1...f...1rd2id
0060 32 30 3a 8b dd 90 78 95 24 2b 52 87 d9 a6 4e 65 20:...x-$R...Ne
0070 60 63 df 43 2b e3 ae 36 3a 6e 6f 64 65 73 36 33 "c.C+...6:nodes63
0080 30 34 3a ff 09 13 5e 6d 7f a0 21 b5 c6 25 2d 29 04:...m...l-5-)
0090 87 84 6a f8 05 0f c0 2a 00 13 70 81 8c 2e 85 00 ...j...p...
00a0 f4 0d 07 22 f3 35 09 1a e2 fc cf 57 bc 41 be 1f ...*5...W.A...
00b0 00 1e ed e2 fa aa ea 02 f2 b5 3e 4e 1a 24 09 8a .....>N.$...
00c0 62 01 63 7d 00 b8 65 f2 b3 d5 a8 7c 3b a4 dc fc b c)...e...|;...
00d0 a7 cb c8 c0 c3 21 7a 80 8b c5 09 ac 22 e5 99 ec ...@-12@.....
```

## DNS



The image shows a Wireshark packet capture of DNS traffic. The top pane displays a list of packets, with packet 403 selected. The middle pane shows the details of packet 403, which is a Domain Name System (DNS) query. The bottom pane shows the raw data of the packet in hexadecimal and ASCII.

**Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
403	1867.589155	2401:4900:889f:baa9...	2401:4900:50:9::280	DNS	107	Standard query 0xc8b2 A safebrowsing.googleapis.com
405	1867.450684	2401:4900:889f:baa9...	2401:4900:50:9::9	DNS	107	Standard query 0xc8b2 A safebrowsing.googleapis.com
426	1867.343315	2401:4900:50:9::9	2401:4900:889f:baa9...	DNS	123	Standard query response 0xc8b2 A safebrowsing.googleapis.com A 142.251.221.234
476	1866.853563	2401:4900:889f:baa9...	2401:4900:50:9::280	DNS	107	Standard query 0x7fad AAAA p-content.securestudies.com
480	1866.720684	2401:4900:889f:baa9...	2401:4900:50:9::9	DNS	107	Standard query 0x7fad AAAA p-content.securestudies.com
482	1866.658731	2401:4900:50:9::9	2401:4900:889f:baa9...	DNS	191	Standard query response 0x7fad AAAA p-content.securestudies.com SOA ns-1744.awsdns-26.co.uk
569	1864.703614	2401:4900:889f:baa9...	2401:4900:50:9::280	DNS	114	Standard query 0xe54a A p2p-sgpl.discovery.steamserver.net
582	1864.564324	2401:4900:889f:baa9...	2401:4900:50:9::9	DNS	114	Standard query 0xe54a A p2p-sgpl.discovery.steamserver.net
584	1864.492528	2401:4900:50:9::9	2401:4900:889f:baa9...	DNS	178	Standard query response 0xe54a A p2p-sgpl.discovery.steamserver.net A 103.10.124.122 A 103.10.124.124 A 103.10.124.123 A 103.10.1...
3434	1809.966170	2401:4900:889f:baa9...	2401:4900:50:9::280	DNS	107	Standard query 0x3c26 A p-content.securestudies.com
3435	1809.956001	2401:4900:889f:baa9...	2401:4900:50:9::280	DNS	107	Standard query 0xc0ec AAAA p-content.securestudies.com
3440	1809.800680	2401:4900:889f:baa9...	2401:4900:50:9::9	DNS	107	Standard query 0x3c26 A p-content.securestudies.com

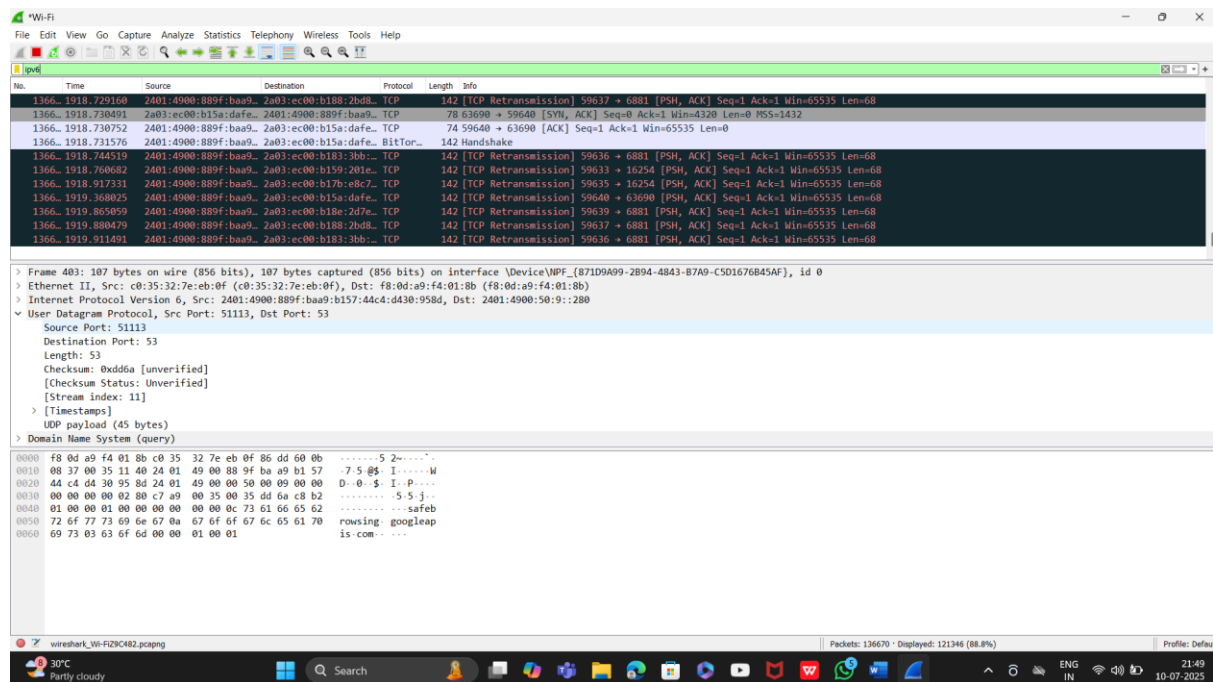
**Packet 403 Details:**

- Frame 403: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface \Device\NPF\_{871D9A99-2B94-4843-B7A9-C5D1676B45AF}, id 0
- Ethernet II, Src: c0:35:32:7e:eb:0f (c0:35:32:7e:eb:0f), Dst: f8:0d:a9:f4:01:8b (f8:0d:a9:f4:01:8b)
- Internet Protocol Version 6, Src: 2401:4900:889f:baa9:b157:44c4:d430:958d, Dst: 2401:4900:50:9::280
- User Datagram Protocol, Src Port: 51113, Dst Port: 53
- Source Port: 51113
- Destination Port: 53
- Length: 53
- Checksum: 0xdd6a [unverified]
- [Checksum Status: Unverified]
- [Stream index: 11]
- [Timestamps]
- UDP payload (45 bytes)

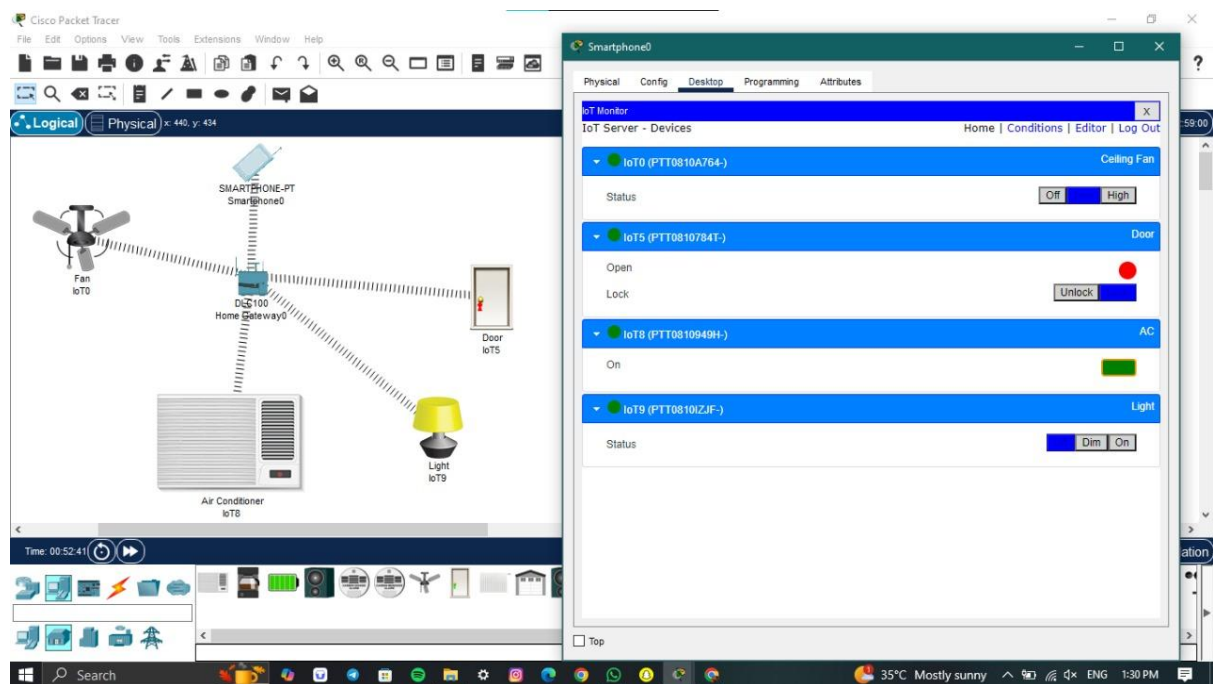
**Domain Name System (query)**

```
0000 f8 0d a9 f4 01 8b c0 35 32 7e eb 0f 86 dd 60 0b .....5 2.....
0010 08 37 00 35 11 40 24 01 49 00 00 9f ba a9 01 57 ...7.5@5-I.....W
0020 44 c4 d4 30 95 8d 24 01 49 00 00 50 00 09 00 00 D...0-$I:P...
0030 00 00 00 00 02 80 c7 a9 00 35 00 35 dd 6a c8 b2 .....5-5-j..
0040 01 00 00 01 00 00 00 00 00 00 0c 73 61 66 65 62 .....safeb
0050 72 6f 77 73 69 6e 67 0a 67 6f 6f 67 6c 65 61 70 rowing: googleap
0060 69 73 63 63 6f 6d 00 00 01 00 01 .....is-com-...
```

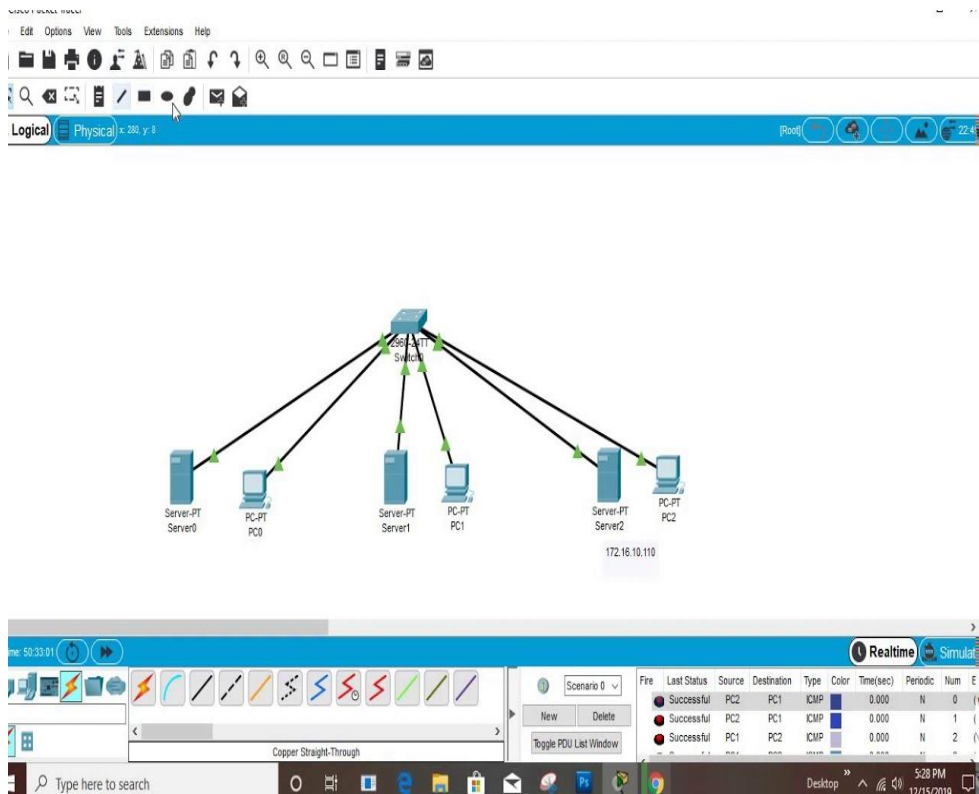
## IPV6



## Control of fan,light



## WLAN



## AAA server

