

Batch: SY-IT(B3)**Experiment Number: 5****Roll Number: 16010423076****Name: Ritesh Jha**

Aim of the Experiment: To write a program to identify the class to which a given IP Address belong to.

Program/ Steps:

```
import re

def validate_ip(ip_address):
    # defined a pattern to match the basic structure of an IP address so that invalid formats are not
    # entered
    pattern = re.compile(r'^(\d{1,3}\.){3}\d{1,3}$')

    if not pattern.match(ip_address):
        return False

    octets = ip_address.split('.')
    return all(0 <= int(octet) <= 255 for octet in octets)

def convert_ip_to_binary(ip_address):
    return ".join(['{int(octet):08b}' for octet in ip_address.split('.')])

def determine_ip_class(ip_address):
    # validate the IP address first
    if not validate_ip(ip_address):
        return "Invalid IP address", None

    binary_ip_address = convert_ip_to_binary(ip_address)

    first_octet_binary = binary_ip_address.split('.')[0]

    if first_octet_binary.startswith('0'):
        ipclass = 'A'
    elif first_octet_binary.startswith('10'):
```

```
    ipclass = 'B'
elif first_octet_binary.startswith('110'):
    ipclass = 'C'
elif first_octet_binary.startswith('1110'):
    ipclass = 'D'
elif first_octet_binary.startswith('1111'):
    ipclass = 'E'
else:
    ip_class = 'Unknown'

return ipclass, binary_ip_address

ip_address = input("Please enter an IP address: ")
ipclass, binary_ip = determine_ip_class(ip_address)

if ipclass == "Invalid IP address":
    print("The IP address provided is invalid.")
else:
    print(f"IP address {ip_address} converts to binary as {binary_ip}")
    print(f"It falls under Class {ipclass}.")
```

Output/Result:**Output**

```
Please enter an IP address: 192.168.0.100
IP address 192.168.0.100 converts to binary as
    11000000101010000000000001100100.
It falls under Class C.

=== Code Execution Successful ===
```

Output

```
Please enter an IP address: 256.300.888.1
The IP address provided is invalid.

=== Code Execution Successful ===
```

Output

```
Please enter an IP address: 10.52.36.11
IP address 10.52.36.11 converts to binary as
    00001010001101000010010000001011
It falls under Class A.

=== Code Execution Successful ===
```

Post Lab Question-Answers:

1. Which OSI layer corresponds to IP Layer?

Ans. Network Layer

2. Compare IPv4 and IPv6 header

Ans. **IPv4 Header:**

- IPv4 uses a 32-bit address, allowing for about 4.3 billion unique addresses.
- The IPv4 header is typically 20 bytes long.

IPv6 Header:

- IPv6 uses a 128-bit address, allowing for a vastly larger number of unique addresses.
- The IPv6 header is 40 bytes long.
- It has fewer fields than IPv4, making it simpler and more efficient.

3. What is fragmentation?

Ans. Fragmentation is the process of breaking down a large data packet into smaller pieces so it can be transmitted over networks that have a smaller maximum transmission unit (MTU) size. Each fragment is then sent separately and reassembled at the destination. This is necessary because different networks can have different MTU sizes and not all packets can pass through in one piece.

4. Explain Subnetting and Supernetting with examples ?

Ans.

Subnetting:

Subnetting is the process of dividing a large network into smaller, more manageable subnetworks (subnets). This helps organize the network and improve security and performance. For example, if you have a network with IP range 192.168.1.0/24, you can subnet it into two smaller networks: 192.168.1.0/25 and 192.168.1.128/25, each supporting 128 IP addresses.

Supernetting:

Supernetting is the opposite of subnetting. It involves combining multiple smaller networks into a larger network. This is often done to simplify routing. For example, if you have two networks 192.168.1.0/24 and 192.168.2.0/24, you can supernet them into one larger network 192.168.0.0/23, which includes all the IP addresses from both networks.

Outcomes:

CO2. Enumerate the layers of the OSI model and TCP/IP model, their functions and Protocols

CO3. Build the skills of sub-netting and routing mechanisms

Conclusion (based on the Results and outcomes achieved):

From the experiment number 5, I learned about IP addresses and their classifications in detail. I also understood the classification algorithm for classful IP Addresses. In the end I implemented a Python code for conversion of the IP Addresses from decimal to binary system and then their classification based on the algorithm which was taught previously.

References:

References:

Books/ Journals/ Websites:

- Behrouz A Forouzan, Data Communication and Networking, Tata Mc Graw hill, India, 4th Edition
- S. Tanenbaum, "Computer Networks", 4th edition, Prentice Hall