

**21BPS1191****SPRIHA ANVI**

# **COMPUTER NETWORKS**

## **LAB ASSIGNMENT**

### **ARP SERVER:**

```
import socket
import struct

SERVER_IP = '127.0.0.1'
SERVER_PORT = 9999

ip_mac_map = {
    '192.168.1.1': '00:11:22:33:44:55',
    '192.168.1.2': '11:22:33:44:55:66',
    '192.168.1.3': '22:33:44:55:66:77',
    '192.168.1.4': '33:44:55:66:77:88'
}

server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

server_socket.bind((SERVER_IP, SERVER_PORT))

while True:

    data, client_address = server_socket.recvfrom(1024)

    ip_address = socket.inet_ntoa(data)

    mac_address = ip_mac_map.get(ip_address, 'Not found')

    mac_address_binary = bytes.fromhex(mac_address.replace(':', ''))

    server_socket.sendto(mac_address_binary, client_address)
```

### **ARP CLIENT**

```
import socket
import struct
```

```

SERVER_IP = '127.0.0.1'
SERVER_PORT = 9999

ip_address_str = input("Enter the IP address: ")

ip_address_binary = socket.inet_aton(ip_address_str)

client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

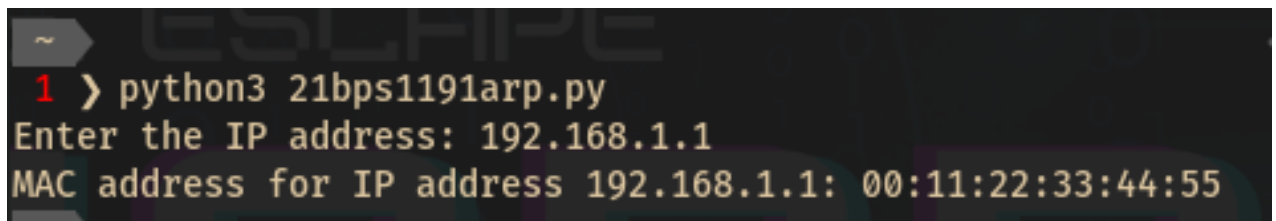
client_socket.sendto(ip_address_binary, (SERVER_IP, SERVER_PORT))

mac_address_binary, server_address = client_socket.recvfrom(1024)

mac_address = ':'.join('{:02x}'.format(b) for b in mac_address_binary)

print('MAC address for IP address {}: {}'.format(ip_address_str, mac_address))

```



```

~
1 > python3 21bps1191arp.py
Enter the IP address: 192.168.1.1
MAC address for IP address 192.168.1.1: 00:11:22:33:44:55

```

## RARP SERVER:

```

import socket
import struct

SERVER_IP = '127.0.0.1'
SERVER_PORT = 9999

ip_mac_map = {
    '192.168.1.1': '00:11:22:33:44:55',
    '192.168.1.2': '11:22:33:44:55:66',

```

```

'192.168.1.3': '22:33:44:55:66:77',
'192.168.1.4': '33:44:55:66:77:88'
}
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

server_socket.bind((SERVER_IP, SERVER_PORT))

while True:

    data, client_address = server_socket.recvfrom(1024)
    ip_address, protocol_type = struct.unpack('!Ic', data)

    mac_address = None
    if protocol_type == b'A':

        mac_address = ip_mac_map.get(str(ip_address), None)
        elif protocol_type == b'P':

            mac_address = ip_mac_map.get(socket.inet_ntoa(ip_address), None)

    mac_address_binary = bytes.fromhex(mac_address.replace(':', '')) if mac_address else b''

    server_socket.sendto(mac_address_binary, client_address)

```

## RARP CLIENT

```

import socket
import struct

SERVER_IP = '127.0.0.1'
SERVER_PORT = 9999

ip_address_str = input("Enter the IP address: ")

ip_address_binary = socket.inet_aton(ip_address_str)

client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

protocol_type_binary = protocol_type_str.encode('utf-8')
data = struct.pack('!Ic', int.from_bytes(ip_address_binary, 'big'), protocol_type_binary)

client_socket.sendto(data, (SERVER_IP, SERVER_PORT))

mac_address_binary, server_address = client_socket.recvfrom(1024)

mac_address = ':'.join('{:02x}'.format(b) for b in mac_address_binary)

```

```
print('MAC address for IP address {} using {} protocol: {}'.format(ip_address_str,  
protocol_type_str, mac_address))
```

```
> python3 rarpc21bps1191.py  
Enter the IP address: 192.168.1.4  
Enter the protocol type (A for AARP ,P for ARP): A  
MAC address for IP address 192.16.1.4 using A protocol
```