**Implementation of HTTP Protocol**

HTTP\_Server.py

from http.server import SimpleHTTPRequestHandler, HTTPServer

# Define the server's address and port

host = "localhost"

port = 8000

# Create and start the server

server = HTTPServer((host, port), SimpleHTTPRequestHandler)

print(f"Server running on http://{host}:{port}")

server.serve\_forever()

HTTP\_Client.py

import requests

url = "http://localhost:8000" # Replace with any server URL

response = requests.get(url)

print(f"Status Code: {response.status\_code}")

print(f"Response:\n{response.text}")

**Implementation of DNS Protocol**

DNS.py

import socket

# Hostname to IP lookup

hostname = "google.com"

ip\_address = socket.gethostbyname(hostname)

print(f"IP Address of {hostname}: {ip\_address}")

# Reverse DNS Lookup (IP to Hostname)

host = socket.gethostbyaddr(ip\_address)

print(f"Hostname for {ip\_address}: {host[0]}")

**Implementation of ARP Protocol**

from scapy.all import ARP, Ether, srp

target\_ip = "192.168.1.1" # Change this to the target IP in your network

# Create an ARP request

arp\_request = ARP(pdst=target\_ip)

ether = Ether(dst="ff:ff:ff:ff:ff:ff") # Broadcast request

packet = ether / arp\_request

# Send the packet and receive response

result = srp(packet, timeout=2, verbose=False)[0]

# Extract and print MAC address

for sent, received in result:

print(f"IP Address: {received.psrc} | MAC Address: {received.hwsrc}")

**Implementation of RARP Protocol**

from scapy.all import RARP, Ether, srp

target\_mac = "00:1A:2B:3C:4D:5E" # Replace with a real MAC address

# Create RARP request

rarp\_request = RARP(hwsrc=target\_mac)

ether = Ether(dst="ff:ff:ff:ff:ff:ff") # Broadcast request

packet = ether / rarp\_request

# Send the packet and receive response

result = srp(packet, timeout=2, verbose=False)[0]

# Extract and print IP address

for sent, received in result:

print(f"MAC Address: {received.hwsrc} | IP Address: {received.psrc}")