

Project Title:

DNS Lookup and Traceroute Tool

Problem Statement:

In networking, diagnosing connectivity issues or understanding the route a packet takes to reach a destination is critical. This tool helps users **resolve domain names**, **measure latency (ping)**, and **trace the path (traceroute)** a packet takes to reach a remote server — similar to tools like ping, tracert, or nslookup.






Project Overview:

This is a Python-based **network diagnostic tool** that performs:

- **DNS resolution** of domain names,
- **Latency checking (Ping)** using ICMP,
- **Traceroute** using TTL-limited ICMP packets.

It is a command-line interface (CLI) tool and provides clean, easy-to-read output for technical users or network engineers who want to debug connection issues or analyze network performance.

Key Features:

Feature	Description
 DNS Lookup	Converts a domain name (e.g., google.com) to its corresponding IP address
 Latency Checker	Sends multiple ICMP requests and calculates round-trip times (RTT)
 Traceroute	Traces the network path taken by packets using Time-To-Live (TTL) values
 RTT Display	Shows each hop's IP address and round-trip time in milliseconds
 Timeout Detection	Detects and reports unreachable intermediate nodes or packet drops

Technologies Used:

Technology Purpose

Python 3	Core programming language
socket	For DNS resolution (gethostbyname)
scapy	For crafting and sending low-level IP/ICMP packets
time	Measuring round-trip time for latency/traceroute

Core Logic:

- **DNS Lookup:** Uses Python's `socket.gethostbyname()` to resolve the domain name.
 - **Latency Check (Ping):** Uses Scapy to send ICMP packets, and calculates the round-trip time using `time.time()`.
 - **Traceroute:** Sends ICMP packets with increasing TTL values and prints each responding node's IP, mimicking the behavior of the `tracert` command.
 - **Timeout Handling:** Handles unreachable hops or no response by printing Request timed out.
-

What You Learned:

- How DNS resolution works at the socket level
 - Basics of IP and ICMP protocols
 - How traceroute works using TTL and ICMP
 - Real-time performance measurement in milliseconds (RTT)
 - Sending and analyzing raw packets using Scapy
 - Building terminal-based network utilities with Python
-

How to Explain It in an Interview (1–2 min):

“I built a network diagnostic tool in Python that performs DNS lookup, latency checks, and traceroutes. It uses the `socket` library to resolve domain names and `scapy` to send ICMP packets for both pinging and tracing the route packets take to the target server. The tool calculates round-trip time (RTT) for each request and displays a detailed summary, including unreachable nodes. This project helped me understand low-level networking concepts like IP, ICMP, and TTL, and also how tools like `ping` and `traceroute` work under the hood.”