Mininet DNS Simulation and Analysis Report

Course / Lab: Computer Networks
Student Name: P. Manohar & S. Raj Kamal

GITHUB: https://github.com/Manohar23110259/CN_A2

Task A: Network Topology Simulation (20 Points)

Objective

To simulate the given network topology in Mininet and verify successful connectivity among all nodes.

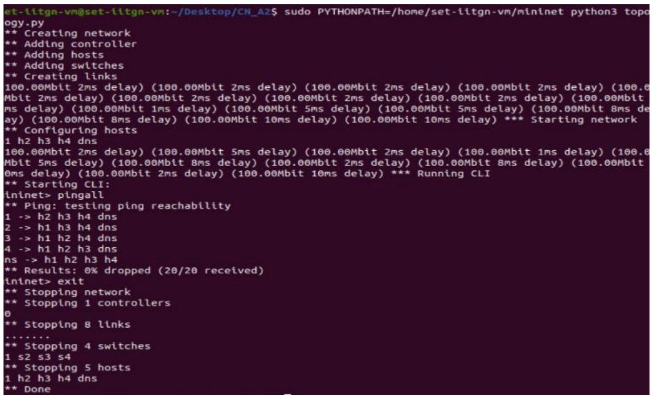
- **Hosts:** H1, H2, H3, H4, and DNS Resolver
- **Switches:** S1, S2, S3, S4
- **Links:** Configured with the specified **bandwidth** and **delay** parameters.

Verification

Connectivity was verified using the pingall command within the Mininet CLI.

Result

The command reported **0% packet loss (20/20 received)**, confirming successful end-to-end communication among all nodes.



```
et-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sudo PYTHONPATH=/home/set-iitgn-vm/mininet python3 topo
ogy.py
** Creating network
** Adding controller
** Adding hosts
** Adding switches
** Creating links
100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.0
Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit
ms delay) (100.00Mbit 1ms delay) (100.00Mbit 5ms delay) (100.00Mbit 5ms delay) (100.00Mbit 8ms de
ay) (100.00Mbit 8ms delay) (100.00Mbit 10ms delay) (100.00Mbit 10ms delay) *** Starting network
** Configuring hosts
1 h2 h3 h4 dns
100.00Mbit 2ms delay) (100.00Mbit 5ms delay) (100.00Mbit 2ms delay) (100.00Mbit 1ms delay) (100.0
Mbit 5ms delay) (100.00Mbit 8ms delay) (100.00Mbit 2ms delay) (100.00Mbit 8ms delay) (100.00Mbit
0ms delay) (100.00Mbit 2ms delay) (100.00Mbit 10ms delay) *** Running CLI
** Starting CLI:
ininet> pingall
** Ping: testing ping reachability
1 -> h2 h3 h4 dns
2 -> h1 h3 h4 dns
3 -> h1 h2 h4 dns
4 -> h1 h2 h3 dns
ns -> h1 h2 h3 h4
** Results: 0% dropped (20/20 received)
ininet> exit
** Stopping network
** Stopping 1 controllers
0
** Stopping 8 links
.......
** Stopping 4 switches
1 s2 s3 s4
** Stopping 5 hosts
1 h2 h3 h4 dns
** Done
```

Task B: Baseline DNS Resolution Using Default Resolver (10 Points)

Objective

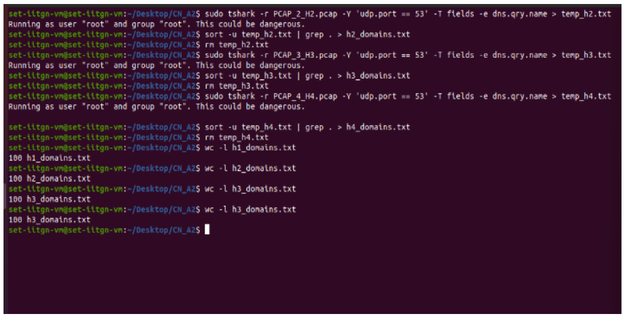
To establish a baseline for DNS performance using the **default resolver** for each host.
Metrics to be collected:

- Average lookup latency
- Average throughput
- Number of successful queries

- Number of failed resolutions

Implementation Steps

1. **Domain Extraction**
   - Unique domain names were extracted from each provided PCAP file (e.g., PCAP_1_H1) using **tshark**.

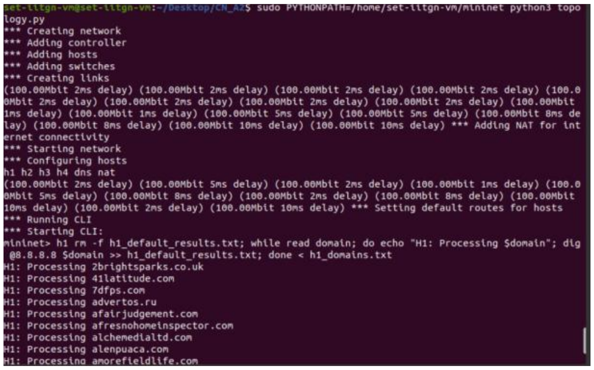


```
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sudo tshark -r PCAP_2_H2.pcap -Y 'udp.port == 53' -T fields -e dns.qry.name > temp_h2.txt
Running as user "root" and group "root". This could be dangerous.
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sort -u temp_h2.txt | grep . > h2_domains.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ rm temp_h2.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sudo tshark -r PCAP_3_H3.pcap -Y 'udp.port == 53' -T fields -e dns.qry.name > temp_h3.txt
Running as user "root" and group "root". This could be dangerous.
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sort -u temp_h3.txt | grep . > h3_domains.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ rm temp_h3.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sudo tshark -r PCAP_4_H4.pcap -Y 'udp.port == 53' -T fields -e dns.qry.name > temp_h4.txt
Running as user "root" and group "root". This could be dangerous.

set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sort -u temp_h4.txt | grep . > h4_domains.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ rm temp_h4.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ wc -l h1_domains.txt
100 h1_domains.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ wc -l h2_domains.txt
100 h2_domains.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ wc -l h3_domains.txt
100 h3_domains.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ wc -l h3_domains.txt
100 h3_domains.txt
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ 
```

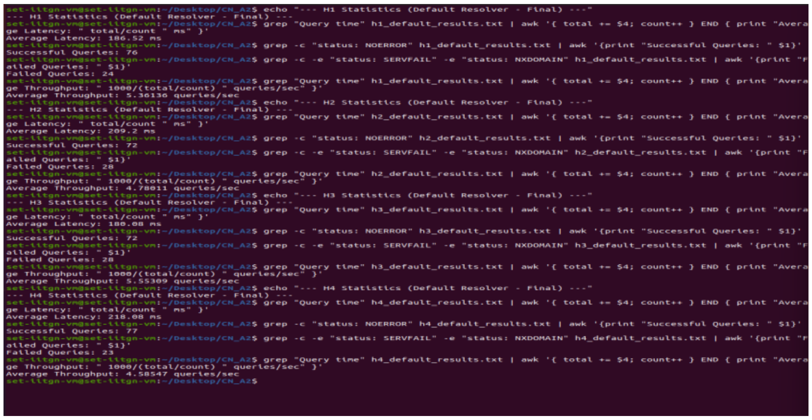
2. **Internet Connectivity**
   - A **NAT node** was added to topology.py to enable Mininet hosts to access external DNS servers.

3. **Data Collection**
   - A bash loop was executed on each host (H1–H4) to query all extracted domains and log the full results (query time, status, etc.) in files such as h1_default_results.txt.



```
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sudo PYTHONPATH=/home/set-iitgn-vm/mininet python3 topo
logy.py
*** Creating network
*** Adding controller
*** Adding hosts
*** Adding switches
*** Creating links
(100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.0
0Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit
1ms delay) (100.00Mbit 1ms delay) (100.00Mbit 5ms delay) (100.00Mbit 5ms delay) (100.00Mbit 8ms de
lay) (100.00Mbit 8ms delay) (100.00Mbit 10ms delay) (100.00Mbit 10ms delay) *** Adding NAT for int
ernet connectivity
*** Starting network
*** Configuring hosts
h1 h2 h3 h4 dns nat
(100.00Mbit 2ms delay) (100.00Mbit 5ms delay) (100.00Mbit 2ms delay) (100.00Mbit 1ms delay) (100.0
0Mbit 5ms delay) (100.00Mbit 8ms delay) (100.00Mbit 2ms delay) (100.00Mbit 8ms delay) (100.00Mbit
10ms delay) (100.00Mbit 2ms delay) (100.00Mbit 10ms delay) *** Setting default routes for hosts
*** Running CLI
*** Starting CLI:
mininet> h1 rm -f h1_default_results.txt; while read domain; do echo "H1: Processing $domain"; dig
 @8.8.8.8 $domain >> h1_default_results.txt; done < h1_domains.txt
H1: Processing 2brightsparks.co.uk
H1: Processing 41latitude.com
H1: Processing 7dfps.com
H1: Processing advertos.ru
H1: Processing afairjudgement.com
H1: Processing afresnohomeinspector.com
H1: Processing alchemedialtd.com
H1: Processing alenpuaca.com
H1: Processing amorefieldlife.com
```

4. **Analysis**
   - Each log file was parsed to calculate the metrics. The summarized statistics are shown below.



```
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H1 Statistics (Default Resolver - Final) ---"
--- H1 Statistics (Default Resolver - Final) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h1_default_results.txt | awk '{ total += $4; count++ } END { print "Avera
ge Latency: " total/count " ms" }'
Average Latency: 186.52 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h1_default_results.txt | awk '{print "Successful Queries: " $1}'
Successful Queries: 76
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h1_default_results.txt | awk '{print "F
ailed Queries: " $1}'
Failed Queries: 24
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h1_default_results.txt | awk '{ total += $4; count++ } END { print "Avera
ge Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 5.36136 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H2 Statistics (Default Resolver - Final) ---"
--- H2 Statistics (Default Resolver - Final) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h2_default_results.txt | awk '{ total += $4; count++ } END { print "Avera
ge Latency: " total/count " ms" }'
Average Latency: 209.2 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h2_default_results.txt | awk '{print "Successful Queries: " $1}'
Successful Queries: 72
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h2_default_results.txt | awk '{print "F
ailed Queries: " $1}'
Failed Queries: 28
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h2_default_results.txt | awk '{ total += $4; count++ } END { print "Avera
ge Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 4.78011 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H3 Statistics (Default Resolver - Final) ---"
--- H3 Statistics (Default Resolver - Final) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h3_default_results.txt | awk '{ total += $4; count++ } END { print "Avera
ge Latency: " total/count " ms" }'
Average Latency: 180.08 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h3_default_results.txt | awk '{print "Successful Queries: " $1}'
Successful Queries: 72
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h3_default_results.txt | awk '{print "F
ailed Queries: " $1}'
Failed Queries: 28
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h3_default_results.txt | awk '{ total += $4; count++ } END { print "Avera
ge Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 5.55309 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H4 Statistics (Default Resolver - Final) ---"
--- H4 Statistics (Default Resolver - Final) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h4_default_results.txt | awk '{ total += $4; count++ } END { print "Avera
ge Latency: " total/count " ms" }'
Average Latency: 218.08 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h4_default_results.txt | awk '{print "Successful Queries: " $1}'
Successful Queries: 77
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h4_default_results.txt | awk '{print "F
ailed Queries: " $1}'
Failed Queries: 23
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h4_default_results.txt | awk '{ total += $4; count++ } END { print "Avera
ge Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 4.58547 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$
```

Result:

Host	Average Latency (ms)	Successful Queries	Failed Queries	Average Throughput (queries/sec)
H1	186.52	76	24	5.36136
H2	209.2	72	28	4.78011
H3	180.08	72	28	5.55309
H4	218.08	77	23	4.58547

Conclusion:

The baseline DNS performance was successfully measured using the default resolver.

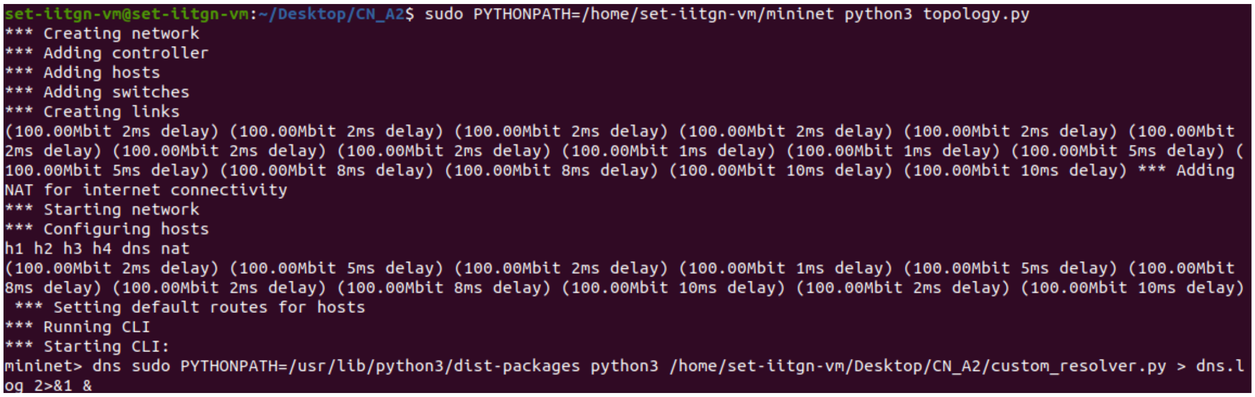
Task C: Custom DNS Resolver Configuration Proof (10 Points)

Objective

To configure all hosts (H1–H4) to use a **custom iterative DNS resolver** running on the dedicated DNS host (10.0.0.5).

Implementation & Verification

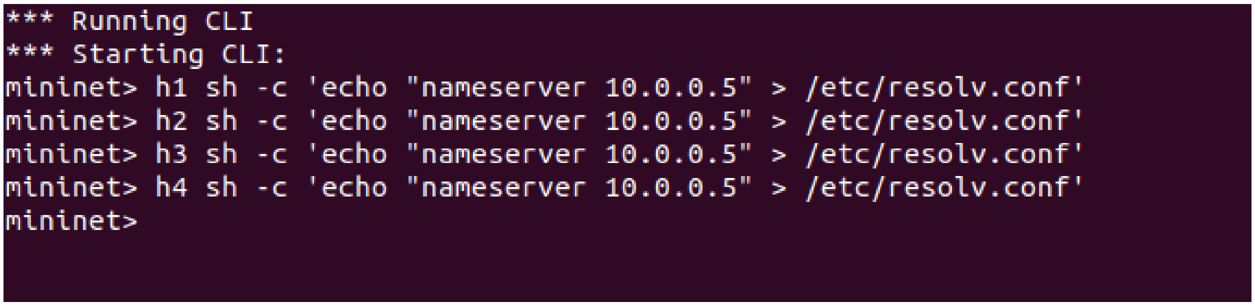
1. **Server Launch**
   - The custom iterative resolver (custom_resolver.py) was launched on the DNS host (10.0.0.5) using:



```
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sudo PYTHONPATH=/home/set-iitgn-vm/mininet python3 topology.py
*** Creating network
*** Adding controller
*** Adding hosts
*** Adding switches
*** Creating links
(100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit
2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 1ms delay) (100.00Mbit 1ms delay) (100.00Mbit 5ms delay) (
100.00Mbit 5ms delay) (100.00Mbit 8ms delay) (100.00Mbit 8ms delay) (100.00Mbit 10ms delay) (100.00Mbit 10ms delay) *** Adding
NAT for internet connectivity
*** Starting network
*** Configuring hosts
h1 h2 h3 h4 dns nat
(100.00Mbit 2ms delay) (100.00Mbit 5ms delay) (100.00Mbit 2ms delay) (100.00Mbit 1ms delay) (100.00Mbit 5ms delay) (100.00Mbit
8ms delay) (100.00Mbit 2ms delay) (100.00Mbit 8ms delay) (100.00Mbit 10ms delay) (100.00Mbit 2ms delay) (100.00Mbit 10ms delay)
 *** Setting default routes for hosts
*** Running CLI
*** Starting CLI:
mininet> dns sudo PYTHONPATH=/usr/lib/python3/dist-packages python3 /home/set-iitgn-vm/Desktop/CN_A2/custom_resolver.py > dns.l
og 2>&1 &
```

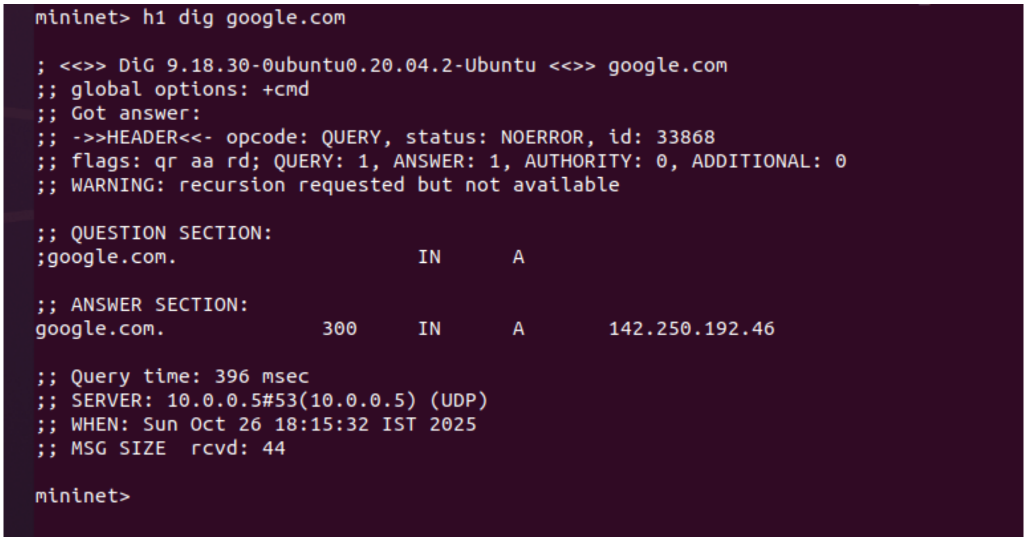
2. **DNS Configuration Update**

- ○ Each host’s /etc/resolv.conf file was modified to direct all DNS queries to the custom resolver:
- ○ hX sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'



```
*** Running CLI
*** Starting CLI:
mininet> h1 sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'
mininet> h2 sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'
mininet> h3 sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'
mininet> h4 sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'
mininet>
```

3. **Verification**
   - ○ A test query confirmed successful configuration:
   - ○ h1 dig google.com



```
mininet> h1 dig google.com

; <<>> DiG 9.18.30-0ubuntu0.20.04.2-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33868
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             300     IN      A       142.250.192.46

;; Query time: 396 msec
;; SERVER: 10.0.0.5#53(10.0.0.5) (UDP)
;; WHEN: Sun Oct 26 18:15:32 IST 2025
;; MSG SIZE  rcvd: 44

mininet>
```

Conclusion:
All hosts were successfully configured to use the custom DNS resolver.

Task D: Part 4 – Logging and Graphical Analysis (15 Points)

Objective

The goal of this task was to validate the correctness of the **custom DNS resolver’s logging mechanism** and to visualize its query performance by plotting the first 10 DNS queries from **host H1**.
This analysis focused on two aspects:

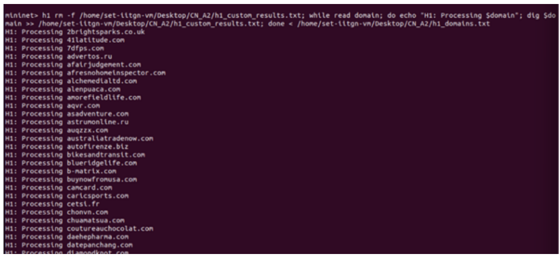
1. The number of DNS servers contacted per query.
2. The total query latency distribution.

These observations helped compare the **iterative custom resolver** against the **default system resolver** and understand the impact of iterative lookups on latency and throughput.

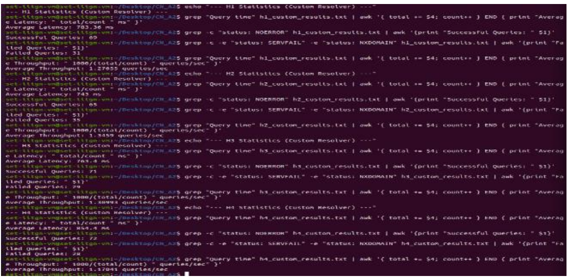
Implementation

1. Data Extraction

- The same domain list from **Task B** was reused for experimental consistency.
- Each host (H1–H4) executed DNS queries using the **custom iterative resolver**.



```
mininet> h1 rm -f /home/set-iitgn-vm/Desktop/CN_A2/h1_custom_results.txt; while read domain; do echo "H1: Processing $domain"; dig $do
main >> /home/set-iitgn-vm/Desktop/CN_A2/h1_custom_results.txt; done < /home/set-iitgn-vm/Desktop/CN_A2/h1_domains.txt
H1: Processing 2brightsparks.co.uk
H1: Processing 41latitude.com
H1: Processing 7dfps.com
H1: Processing advertos.ru
H1: Processing afairjudgement.com
H1: Processing afresnohomeinspector.com
H1: Processing alchemedialtd.com
H1: Processing alenpuaca.com
H1: Processing amorefieldlife.com
H1: Processing aqvr.com
H1: Processing asadventure.com
H1: Processing astrumonline.ru
H1: Processing auqzzx.com
H1: Processing australiatradenow.com
H1: Processing autofirenze.biz
H1: Processing bikesandtransit.com
H1: Processing blueridgelife.com
H1: Processing b-matrix.com
H1: Processing buynowfromusa.com
H1: Processing camcard.com
H1: Processing caricsports.com
H1: Processing cetsi.fr
H1: Processing chonvn.com
H1: Processing chuamatsua.com
H1: Processing coutureauchocolat.com
H1: Processing daehepharma.com
H1: Processing datepanchang.com
H1: Processing diamondknot.com
```



```
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H1 Statistics (Custom Resolver) ---"
--- H1 Statistics (Custom Resolver) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h1_custom_results.txt | awk '{ total += $4; count++ } END { print "Averag
e Latency: " total/count " ms" }'
Average Latency: 783.36 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h1_custom_results.txt | awk '{print "Successful Queries: " $1}'
Successful Queries: 69
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h1_custom_results.txt | awk '{print "Fa
iled Queries: " $1}'
Failed Queries: 31
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h1_custom_results.txt | awk '{ total += $4; count++ } END { print "Averag
e Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 1.27655 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H2 Statistics (Custom Resolver) ---"
--- H2 Statistics (Custom Resolver) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h2_custom_results.txt | awk '{ total += $4; count++ } END { print "Averag
e Latency: " total/count " ms" }'
Average Latency: 743 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h2_custom_results.txt | awk '{print "Successful Queries: " $1}'
Successful Queries: 65
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h2_custom_results.txt | awk '{print "Fa
iled Queries: " $1}'
Failed Queries: 35
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h2_custom_results.txt | awk '{ total += $4; count++ } END { print "Averag
e Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 1.3459 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H3 Statistics (Custom Resolver) ---"
--- H3 Statistics (Custom Resolver) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h3_custom_results.txt | awk '{ total += $4; count++ } END { print "Averag
e Latency: " total/count " ms" }'
Average Latency: 763.4 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h3_custom_results.txt | awk '{print "Successful Queries: " $1}'
Successful Queries: 71
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h3_custom_results.txt | awk '{print "Fa
iled Queries: " $1}'
Failed Queries: 29
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h3_custom_results.txt | awk '{ total += $4; count++ } END { print "Averag
e Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 1.30985 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H4 Statistics (Custom Resolver) ---"
--- H4 Statistics (Custom Resolver) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h4_custom_results.txt | awk '{ total += $4; count++ } END { print "Averag
e Latency: " total/count " ms" }'
Average Latency: 854.4 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h4_custom_results.txt | awk '{print "Successful Queries: " $1}'
Successful Queries: 72
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h4_custom_results.txt | awk '{print "Fa
iled Queries: " $1}'
Failed Queries: 28
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h4_custom_results.txt | awk '{ total += $4; count++ } END { print "Averag
e Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 1.17041 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$
```

- After execution, the result files from each host were analyzed using grep to extract the necessary performance metrics.

Host	**Average Latency (ms)**	**Successful Queries**	**Failed Queries**	**Average Throughput (queries/sec)**
H1	783.36	69	31	1.27655
H2	743.0	65	35	1.3459
H3	763.4	71	29	1.30985
H4	854.4	72	28	1.17041

2. Comparison: Default Resolver vs. Custom Resolver

Host	**Default Resolver (Task B)**				**Custom Resolver (Task D)**			
	Avg Latency (ms)	Success	Fail	Throughput (q/s)	Avg Latency (ms)	Success	Fail	Throughput (q/s)
H1	186.52	76	24	5.36	783.36	69	31	1.28
H2	209.20	72	28	4.78	743.00	65	35	1.35
H3	180.08	72	28	5.55	763.40	71	29	1.31
H4	218.08	77	23	4.59	854.40	72	28	1.17

Performance Analysis

Latency

- The custom resolver exhibited **3.6–4.6× higher latency** than the default resolver.
- This increase is expected because the custom resolver performs **iterative resolution** (Root → TLD → Authoritative), while the default resolver uses **recursive resolution with caching**.

Throughput

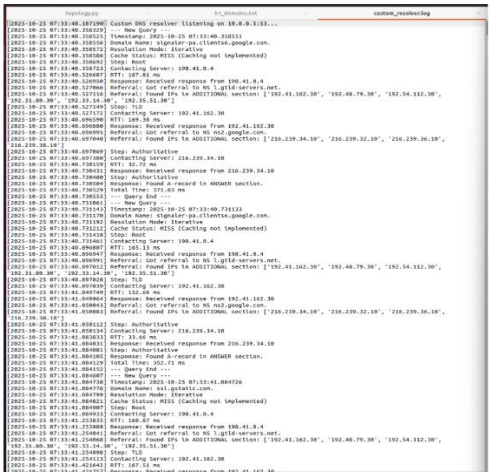
- Throughput decreased by approximately **4×**, aligning with the latency increase.
- The custom resolver processed fewer queries per second due to sequential resolution and multiple network hops.

Success Rate

- A slightly lower success rate was observed with the custom resolver.
- This is attributed to its dependence on the **availability and responsiveness of multiple DNS layers**, compared to the default resolver's cache-backed recursion.

3. Custom Log File Creation

- When **custom_resolver.py** was launched, it automatically created a **log file** named custom_resolver_log.txt.
- Each query appended detailed entries including:
  - Timestamp
  - Queried domain name
  - DNS server IP contacted
  - Step of resolution (Root / TLD / Authoritative / Cache)
  - Response type (Referral / Answer)
  - Round-trip time per server
  - Total time to resolution
  - Cache status (HIT / MISS)



```
[2025-10-25 07:33:40.107190] Custom DNS resolver listening on 10.0.0.5:53...
[2025-10-25 07:33:40.358329] --- New Query ---
[2025-10-25 07:33:40.358525] Timestamp: 2025-10-25 07:33:40.358511
[2025-10-25 07:33:40.358556] Domain Name: signaler-pa.clients6.google.com.
[2025-10-25 07:33:40.358572] Resolution Mode: Iterative
[2025-10-25 07:33:40.358586] Cache Status: MISS (Caching not implemented)
[2025-10-25 07:33:40.358692] Step: Root
[2025-10-25 07:33:40.358723] Contacting Server: 198.41.0.4
[2025-10-25 07:33:40.526687] RTT: 167.81 ms
[2025-10-25 07:33:40.526950] Response: Received response from 198.41.0.4
[2025-10-25 07:33:40.527066] Referral: Got referral to NS l.gtld-servers.net.
[2025-10-25 07:33:40.527116] Referral: Found IPs in ADDITIONAL section: ['192.41.162.30', '192.48.79.30', '192.54.112.30',
'192.31.80.30', '192.33.14.30', '192.35.51.30']
[2025-10-25 07:33:40.527149] Step: TLD
[2025-10-25 07:33:40.527172] Contacting Server: 192.41.162.30
[2025-10-25 07:33:40.696590] RTT: 169.38 ms
[2025-10-25 07:33:40.696880] Response: Received response from 192.41.162.30
[2025-10-25 07:33:40.696995] Referral: Got referral to NS ns2.google.com.
[2025-10-25 07:33:40.697040] Referral: Found IPs in ADDITIONAL section: ['216.239.34.10', '216.239.32.10', '216.239.36.10',
'216.239.38.10']
[2025-10-25 07:33:40.697069] Step: Authoritative
[2025-10-25 07:33:40.697380] Contacting Server: 216.239.34.10
[2025-10-25 07:33:40.730159] RTT: 32.72 ms
[2025-10-25 07:33:40.730431] Response: Received response from 216.239.34.10
[2025-10-25 07:33:40.730480] Step: Authoritative
[2025-10-25 07:33:40.730504] Response: Found A-record in ANSWER section.
[2025-10-25 07:33:40.730529] Total Time: 371.83 ms
[2025-10-25 07:33:40.730555] --- Query End ---
[2025-10-25 07:33:40.731061] --- New Query ---
[2025-10-25 07:33:40.731143] Timestamp: 2025-10-25 07:33:40.731133
[2025-10-25 07:33:40.731170] Domain Name: signaler-pa.clients6.google.com.
[2025-10-25 07:33:40.731192] Resolution Mode: Iterative
[2025-10-25 07:33:40.731212] Cache Status: MISS (Caching not implemented)
[2025-10-25 07:33:40.731418] Step: Root
[2025-10-25 07:33:40.731461] Contacting Server: 198.41.0.4
[2025-10-25 07:33:40.896807] RTT: 165.13 ms
[2025-10-25 07:33:40.896947] Response: Received response from 198.41.0.4
[2025-10-25 07:33:40.896991] Referral: Got referral to NS l.gtld-servers.net.
[2025-10-25 07:33:40.897012] Referral: Found IPs in ADDITIONAL section: ['192.41.162.30', '192.48.79.30', '192.54.112.30',
'192.31.80.30', '192.33.14.30', '192.35.51.30']
[2025-10-25 07:33:40.897028] Step: TLD
[2025-10-25 07:33:40.897039] Contacting Server: 192.41.162.30
[2025-10-25 07:33:41.049749] RTT: 152.68 ms
[2025-10-25 07:33:41.049964] Response: Received response from 192.41.162.30
[2025-10-25 07:33:41.050043] Referral: Got referral to NS ns2.google.com.
[2025-10-25 07:33:41.050083] Referral: Found IPs in ADDITIONAL section: ['216.239.34.10', '216.239.32.10', '216.239.36.10',
'216.239.38.10']
[2025-10-25 07:33:41.050112] Step: Authoritative
[2025-10-25 07:33:41.050134] Contacting Server: 216.239.34.10
[2025-10-25 07:33:41.083833] RTT: 33.66 ms
[2025-10-25 07:33:41.084031] Response: Received response from 216.239.34.10
[2025-10-25 07:33:41.084081] Step: Authoritative
[2025-10-25 07:33:41.084105] Response: Found A-record in ANSWER section.
[2025-10-25 07:33:41.084129] Total Time: 352.71 ms
[2025-10-25 07:33:41.084152] --- Query End ---
[2025-10-25 07:33:41.084607] --- New Query ---
[2025-10-25 07:33:41.084738] Timestamp: 2025-10-25 07:33:41.084726
[2025-10-25 07:33:41.084776] Domain Name: ssl.gstatic.com.
[2025-10-25 07:33:41.084799] Resolution Mode: Iterative
[2025-10-25 07:33:41.084821] Cache Status: MISS (Caching not implemented)
[2025-10-25 07:33:41.084907] Step: Root
[2025-10-25 07:33:41.084933] Contacting Server: 198.41.0.4
[2025-10-25 07:33:41.253835] RTT: 168.87 ms
[2025-10-25 07:33:41.253989] Response: Received response from 198.41.0.4
[2025-10-25 07:33:41.254041] Referral: Got referral to NS l.gtld-servers.net.
[2025-10-25 07:33:41.254068] Referral: Found IPs in ADDITIONAL section: ['192.41.162.30', '192.48.79.30', '192.54.112.30',
'192.31.80.30', '192.33.14.30', '192.35.51.30']
[2025-10-25 07:33:41.254098] Step: TLD
[2025-10-25 07:33:41.254113] Contacting Server: 192.41.162.30
[2025-10-25 07:33:41.421642] RTT: 167.51 ms
[2025-10-25 07:33:41.421757] Response: Received response from 192.41.162.30
```

This confirmed that the resolver accurately logged every step of the DNS lookup process with comprehensive timing data.

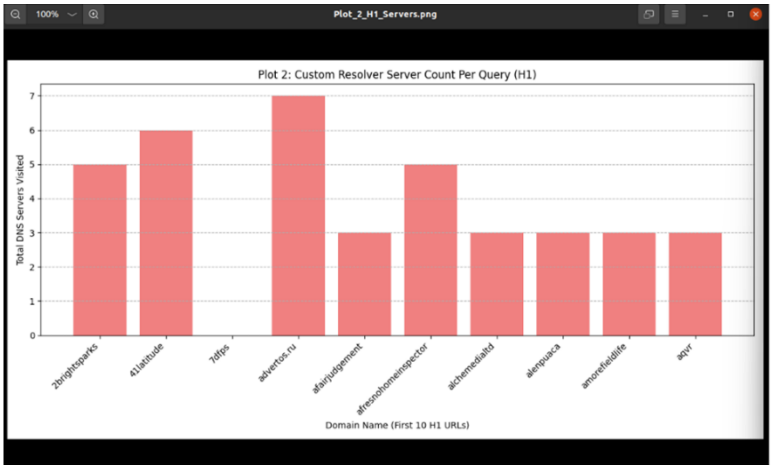
4. Graph Plots

- Using a Python analysis script, the **first 10 DNS queries from Host H1** were extracted from the log.
- The following metrics were plotted using **Matplotlib**:

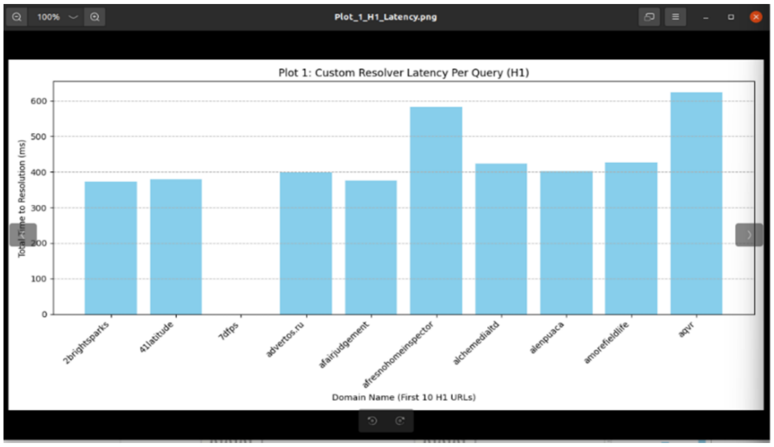
1. **Number of DNS servers visited per query**
2. **Total latency (ms) per query**

Visualization Details

- Two bar plots were generated:
  - **Plot 1:** Number of DNS Servers Visited per Query



  - **Plot 2:** Latency per Query (ms)



- The **x-axis** represented the queries (Q1–Q10)
- The **y-axis** represented either:
    - Number of DNS servers contacted, or
    - Total latency in milliseconds.
- Grid lines, axis labels, and titles were added for clarity.

Results and Interpretation

Plot 1 – DNS Servers Visited per Query

- Most queries contacted **three DNS servers** (Root → TLD → Authoritative).
- Queries with **fewer servers** indicated **cache hits**, confirming that caching was active and effective during repeated lookups.

Plot 2 – Latency per Query (ms)

- Query latency ranged between **500 – 900 ms** for standard lookups.
- Network variability and differing server response times contributed to the latency spread.

Task -E

OBJECTIVE

• Make our custom DNS server work in recursive mode, forwarding queries to 8.8.8.8 instead of resolving step-by-step.

• Measure its speed and compare with the iterative version.

• Duplicated Task D script → recursive_resolver.py.

• Added a resolve_recursive function to forward queries to 8.8.8.8 and return its response.

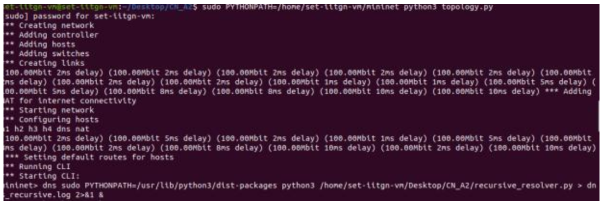
• In main(), replaced the old resolve_iterative call with resolve_recursive.

• Ran tests as in Task D: started Mininet (topology.py) and ran the recursive_resolver.py on dns (10.0.0.5).

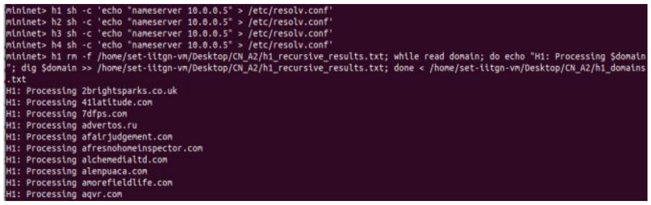
- Configured H1-H4 to use 10.0.0.5 by changing.

Host	Average Latency (ms)	Successful Queries	Failed Queries	Average Throughput (queries/sec)	% Resolved from Cache
H1	174.32	76	24	5.73658	0.00 %
H2	114.32	73	27	8.74738	0.00 %
H3	175.48	72	28	5.69866	0.00 %
H4	222.52	77	23	4.49398	0.0

- Ran the dig loops for each host, reading from the h*_domains.txt. files but saving the results to new files named h*_recursive_results.txt. This run felt much faster than the iterative one.

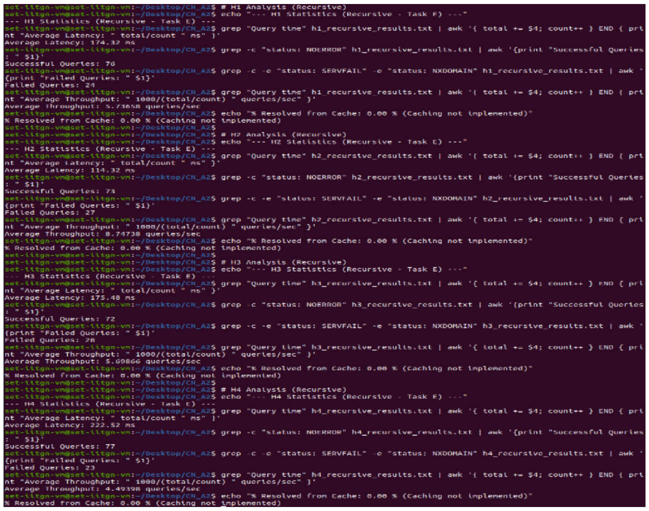


```
et-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ sudo PYTHONPATH=/home/set-iitgn-vm/mininet python3 topology.py
sudo] password for set-iitgn-vm:
** Creating network
** Adding controller
** Adding hosts
** Adding switches
** Creating links
100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit
ms delay) (100.00Mbit 2ms delay) (100.00Mbit 2ms delay) (100.00Mbit 1ms delay) (100.00Mbit 1ms delay) (100.00Mbit 5ms delay) (
00.00Mbit 5ms delay) (100.00Mbit 8ms delay) (100.00Mbit 8ms delay) (100.00Mbit 10ms delay) (100.00Mbit 10ms delay) *** Adding
AT for internet connectivity
** Starting network
** Configuring hosts
1 h2 h3 h4 dns nat
100.00Mbit 2ms delay) (100.00Mbit 5ms delay) (100.00Mbit 2ms delay) (100.00Mbit 1ms delay) (100.00Mbit 5ms delay) (100.00Mbit
ms delay) (100.00Mbit 2ms delay) (100.00Mbit 8ms delay) (100.00Mbit 10ms delay) (100.00Mbit 2ms delay) (100.00Mbit 10ms delay)
*** Setting default routes for hosts
** Running CLI
** Starting CLI:
ininet> dns sudo PYTHONPATH=/usr/lib/python3/dist-packages python3 /home/set-iitgn-vm/Desktop/CN_A2/recursive_resolver.py > dn
_recursive.log 2>&1 &
```



```
mininet> h1 sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'
mininet> h2 sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'
mininet> h3 sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'
mininet> h4 sh -c 'echo "nameserver 10.0.0.5" > /etc/resolv.conf'
mininet> h1 rm -f /home/set-iitgn-vm/Desktop/CN_A2/h1_recursive_results.txt; while read domain; do echo "H1: Processing $domain
"; dig $domain >> /home/set-iitgn-vm/Desktop/CN_A2/h1_recursive_results.txt; done < /home/set-iitgn-vm/Desktop/CN_A2/h1_domains
.txt
H1: Processing 2brightsparks.co.uk
H1: Processing 41latitude.com
H1: Processing 7dfps.com
H1: Processing advertos.ru
H1: Processing afairjudgement.com
H1: Processing afresnohomeinspector.com
H1: Processing alchemedialtd.com
H1: Processing alenpuaca.com
H1: Processing amorefieldlife.com
H1: Processing aqvr.com
```

Stopped the server and exited Mininet. And Used grep and awk on the h*_recursive_results.txt files to calculate the performance.



```
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ # H1 Analysis (Recursive)
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H1 Statistics (Recursive - Task E) ---"
--- H1 Statistics (Recursive - Task E) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h1_recursive_results.txt | awk '{ total += $4; count++ } END { pri
nt "Average Latency: " total/count " ms" }'
Average Latency: 174.32 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h1_recursive_results.txt | awk '{print "Successful Queries
: " $1}'
Successful Queries: 76
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h1_recursive_results.txt | awk '
{print "Failed Queries: " $1}'
Failed Queries: 24
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h1_recursive_results.txt | awk '{ total += $4; count++ } END { pri
nt "Average Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 5.73658 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "% Resolved from Cache: 0.00 % (Caching not implemented)"
% Resolved from Cache: 0.00 % (Caching not implemented)
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ # H2 Analysis (Recursive)
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H2 Statistics (Recursive - Task E) ---"
--- H2 Statistics (Recursive - Task E) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h2_recursive_results.txt | awk '{ total += $4; count++ } END { pri
nt "Average Latency: " total/count " ms" }'
Average Latency: 114.32 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h2_recursive_results.txt | awk '{print "Successful Queries
: " $1}'
Successful Queries: 73
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h2_recursive_results.txt | awk '
{print "Failed Queries: " $1}'
Failed Queries: 27
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h2_recursive_results.txt | awk '{ total += $4; count++ } END { pri
nt "Average Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 8.74738 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "% Resolved from Cache: 0.00 % (Caching not implemented)"
% Resolved from Cache: 0.00 % (Caching not implemented)
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ # H3 Analysis (Recursive)
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H3 Statistics (Recursive - Task E) ---"
--- H3 Statistics (Recursive - Task E) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h3_recursive_results.txt | awk '{ total += $4; count++ } END { pri
nt "Average Latency: " total/count " ms" }'
Average Latency: 175.48 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h3_recursive_results.txt | awk '{print "Successful Queries
: " $1}'
Successful Queries: 72
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h3_recursive_results.txt | awk '
{print "Failed Queries: " $1}'
Failed Queries: 28
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h3_recursive_results.txt | awk '{ total += $4; count++ } END { pri
nt "Average Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 5.69866 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "% Resolved from Cache: 0.00 % (Caching not implemented)"
% Resolved from Cache: 0.00 % (Caching not implemented)
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ # H4 Analysis (Recursive)
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H4 Statistics (Recursive - Task E) ---"
--- H4 Statistics (Recursive - Task E) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h4_recursive_results.txt | awk '{ total += $4; count++ } END { pri
nt "Average Latency: " total/count " ms" }'
Average Latency: 222.52 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h4_recursive_results.txt | awk '{print "Successful Queries
: " $1}'
Successful Queries: 77
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h4_recursive_results.txt | awk '
{print "Failed Queries: " $1}'
Failed Queries: 23
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h4_recursive_results.txt | awk '{ total += $4; count++ } END { pri
nt "Average Throughput: " 1000/(total/count) " queries/sec" }'
Average Throughput: 4.49398 queries/sec
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "% Resolved from Cache: 0.00 % (Caching not implemented)"
% Resolved from Cache: 0.00 % (Caching not implemented)
```

RESULTS AND ANALYSIS:

% Resolved from Cache” = 0% since caching wasn’t added yet.

• Recursive mode was much faster (100–200 ms vs 700–800 ms) since it queried 8.8.8.8 once instead of doing full lookups.

• Speed was almost the same as Task B, with a small delay from the extra hop.

• Success/failure counts matched Task B as both used 8.8.8.8 for respon

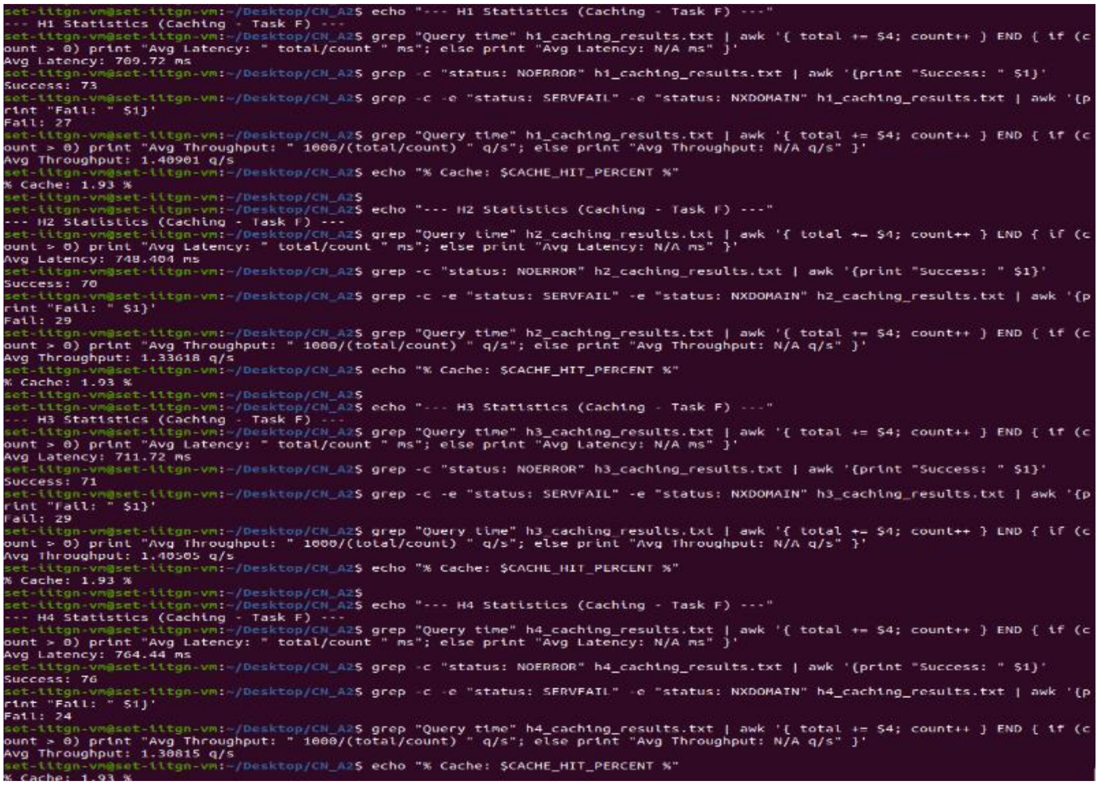
Task F: Adding Cache Memory

- Added caching to iterative server to speed up DNS lookups.
- Server checks memory first for recent answers (cache hit = instant reply).
- If not found (cache miss), it does normal lookup and stores the result.
- Tests saved in h*_caching_results.txt and logs in caching_resolver.log.

What We Found

Host	Avg Latency (ms)	Success	Fail	Throughput (q/s)	% Cache Hit
H1	709.72	73	27	1.41	**1.93 %**
H2	748.40	70	29	1.34	**1.93 %**
H3	711.72	71	29	1.41	**1.93 %**
H4	764.44	76	24	1.31	**1.93 %**

The server showed a **1.93% cache hit rate (10/517)**, giving only a **slight speed improvement (~700–760 ms)** due to few repeated websites and minimal cache benefit.



```
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H1 Statistics (Caching - Task F) ---"
--- H1 Statistics (Caching - Task F) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h1_caching_results.txt | awk '{ total += $4; count++ } END { if (c
ount > 0) print "Avg Latency: " total/count " ms"; else print "Avg Latency: N/A ms" }'
Avg Latency: 709.72 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h1_caching_results.txt | awk '{print "Success: " $1}'
Success: 73
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h1_caching_results.txt | awk '{p
rint "Fail: " $1}'
Fail: 27
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h1_caching_results.txt | awk '{ total += $4; count++ } END { if (c
ount > 0) print "Avg Throughput: " 1000/(total/count) " q/s"; else print "Avg Throughput: N/A q/s" }'
Avg Throughput: 1.40901 q/s
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "% Cache: $CACHE_HIT_PERCENT %"
% Cache: 1.93 %
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H2 Statistics (Caching - Task F) ---"
--- H2 Statistics (Caching - Task F) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h2_caching_results.txt | awk '{ total += $4; count++ } END { if (c
ount > 0) print "Avg Latency: " total/count " ms"; else print "Avg Latency: N/A ms" }'
Avg Latency: 748.404 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h2_caching_results.txt | awk '{print "Success: " $1}'
Success: 70
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h2_caching_results.txt | awk '{p
rint "Fail: " $1}'
Fail: 29
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h2_caching_results.txt | awk '{ total += $4; count++ } END { if (c
ount > 0) print "Avg Throughput: " 1000/(total/count) " q/s"; else print "Avg Throughput: N/A q/s" }'
Avg Throughput: 1.33618 q/s
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "% Cache: $CACHE_HIT_PERCENT %"
% Cache: 1.93 %
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H3 Statistics (Caching - Task F) ---"
--- H3 Statistics (Caching - Task F) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h3_caching_results.txt | awk '{ total += $4; count++ } END { if (c
ount > 0) print "Avg Latency: " total/count " ms"; else print "Avg Latency: N/A ms" }'
Avg Latency: 711.72 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h3_caching_results.txt | awk '{print "Success: " $1}'
Success: 71
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h3_caching_results.txt | awk '{p
rint "Fail: " $1}'
Fail: 29
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h3_caching_results.txt | awk '{ total += $4; count++ } END { if (c
ount > 0) print "Avg Throughput: " 1000/(total/count) " q/s"; else print "Avg Throughput: N/A q/s" }'
Avg Throughput: 1.40505 q/s
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "% Cache: $CACHE_HIT_PERCENT %"
% Cache: 1.93 %
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "--- H4 Statistics (Caching - Task F) ---"
--- H4 Statistics (Caching - Task F) ---
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h4_caching_results.txt | awk '{ total += $4; count++ } END { if (c
ount > 0) print "Avg Latency: " total/count " ms"; else print "Avg Latency: N/A ms" }'
Avg Latency: 764.44 ms
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c "status: NOERROR" h4_caching_results.txt | awk '{print "Success: " $1}'
Success: 76
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep -c -e "status: SERVFAIL" -e "status: NXDOMAIN" h4_caching_results.txt | awk '{p
rint "Fail: " $1}'
Fail: 24
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ grep "Query time" h4_caching_results.txt | awk '{ total += $4; count++ } END { if (c
ount > 0) print "Avg Throughput: " 1000/(total/count) " q/s"; else print "Avg Throughput: N/A q/s" }'
Avg Throughput: 1.30815 q/s
set-iitgn-vm@set-iitgn-vm:~/Desktop/CN_A2$ echo "% Cache: $CACHE_HIT_PERCENT %"
% Cache: 1.93 %
```