LSAP HW4

1. Domain Analysis

Use below script to generate the results:

```
#!/bin/bash
SITES_FILE="sites.txt"
OUTPUT_CSV="task1_results.csv"

if [ ! -f "$SITES_FILE" ]; then
    exit 1
fi

echo "Website,A (IPv4),AAAA (IPv6),CNAME,MX,DNSSEC" > $OUTPUT_CSV


while read site; do
    if [ -z "$site" ]; then
        continue
    fi

    A_RECORD=$(dig +short $site A | tr '\n' ' ')
    AAAA_RECORD=$(dig +short $site AAAA | tr '\n' ' ')
    CNAME_RECORD=$(dig +short $site CNAME)
    MX_RECORD=$(dig +short $site MX | tr '\n' ' ')

    if dig +dnssec $site | grep -q "RRSIG"; then
        DNSSEC_STATUS="Enabled"
    else
        DNSSEC_STATUS="Disabled"
    fi

    echo "\"$site\",\"$A_RECORD\",\"$AAAA_RECORD\",\"$CNAME_RECORD\",\"$MX_RECORD\",\"$DNSSEC_STATUS\"" >> $OUTPUT_CSV

    sleep 0.5

done < "$SITES_FILE"
```

Website	A (IPv4)	AAAA (IPv6)	CNAME	MX	DNSSEC
google.com	142.250.196.206	2404:6800:4012:8::200e	nan	10 smtp.google.com.	Disabled
github.com	20.27.177.113	nan	nan	1 aspmx.l.google.com. 10 alt3.aspmx.l.google.com. 10 alt4.aspmx.l.google.com. 5 alt1.aspmx.l.google.com. 5 alt2.aspmx.l.google.com.	Disabled
wikipedia.org	103.102.166.224	2001:df2:e500:ed1a::1	nan	10 mx-in1001.wikimedia.org. 10 mx-in2001.wikimedia.org.	Disabled
youtube.com	142.250.196.206	2404:6800:4012:9::200e	nan	0 smtp.google.com.	Disabled
instagram.com	31.13.87.174	2a03:2880:f217:e5:face:b00c:0:4420	nan	10 mxa-00082601.gslb.pphosted.com. 10 mxb-00082601.gslb.pphosted.com.	Disabled
facebook.com	31.13.70.36	2a03:2880:f34c:1:face:b00c:0:25de	nan	10 smtpin.vvv.facebook.com.	Disabled
messenger.com	57.144.152.141	2a03:2880:f34c:8d:face:b00c:0:2	nan	10 mxb-00082601.gslb.pphosted.com. 10 mxa-00082601.gslb.pphosted.com.	Disabled

Website	A (IPv4)	AAAA (IPv6)	CNAME	MX	DNSSEC
apple.com	17.253.144.10	2620:149:af0::10	nan	10 mx-in.g.apple.com. 20 mx-in-ma.apple.com. 20 mx-in-rn.apple.com. 20 mx-in-sg.apple.com. 20 mx-in-hfd.apple.com. 20 mx-in-vib.apple.com.	Disabled
ani.gamer.com.tw	104.18.2.197 104.18.3.197	nan	nan	nan	Disabled
chatgpt.com	172.64.155.209 104.18.32.47	2a06:98c1:310b::ac40:9bd1 2a06:98c1:3100::6812:202f	nan	nan	Enabled

DNS lookup path：

```
Client
  ↓
Local DNS Resolver (ISP DNS / 1.1.1.1 / 8.8.8.8)
  ↓  asks
Root Name Server (.)
  ↓  directs to
TLD Server (.com / .org / .tw)
  ↓  directs to
Authoritative Name Server (ns1.google.com, ns2.google.com, etc.)
  ↓  returns
IP Address (A / AAAA Record)
```

2. DNS Resolution Time Measurement.

Use below script to generate the results:

```
#!/bin/bash
SITES_FILE="sites.txt"
OUTPUT_FILE="task2_dns_time.csv"

while read site; do
    if [ -z "$site" ]; then
        continue
    fi

    total=0
    count=5

    for i in $(seq 1 $count); do
        time=$(dig +stats $site | grep "Query time" | awk '{print $4}')
        total=$((total + time))
        sleep 0.3
    done

    avg=$((total / count))

    echo "$site,$avg" >> $OUTPUT_FILE
done < "$SITES_FILE"
```

Website	Average Query Time (ms)
google.com	78
github.com	51
wikipedia.org	51
youtube.com	52
instagram.com	76
facebook.com	48
messenger.com	53

Website	Average Query Time (ms)
apple.com	47
ani.gamer.com.tw	49
chatgpt.com	23

3. DNS Load Balancing

chatgpt.com returns different IP addresses on repeated DNS queries, indicating DNS load balancing across multiple edge servers.

```
for i in {1..10}; do
    dig +short chatgpt.com A | head -n 1
    sleep 0.3
done
172.64.155.209
104.18.32.47
104.18.32.47
172.64.155.209
172.64.155.209
104.18.32.47
172.64.155.209
172.64.155.209
104.18.32.47
104.18.32.47
```

4. CDN Identification

```
#!/bin/bash
SITES_FILE="sites.txt"
OUTPUT="task4_cdn_results.csv"

echo "Website,CDN Provider" > $OUTPUT

while read site; do
    if [ -z "$site" ]; then continue; fi


    ip=$(dig +short $site A | head -n 1)

    whois_info=$(whois $ip | grep -Ei "cloudflare|fastly|akamai|google|facebook|amazon|edge|cdn" | head -n 1)

    if echo "$whois_info" | grep -qi "cloudflare"; then provider="Cloudflare"
    elif echo "$whois_info" | grep -qi "fastly"; then provider="Fastly"
    elif echo "$whois_info" | grep -qi "akamai"; then provider="Akamai"
    elif echo "$whois_info" | grep -qi "google"; then provider="Google Global CDN"
    elif echo "$whois_info" | grep -qi "facebook"; then provider="Meta Edge CDN"
    elif echo "$whois_info" | grep -qi "amazon"; then provider="AWS CloudFront"
    else provider="No CDN / Direct Hosting"
    fi

    echo "$site,$provider" >> $OUTPUT

    sleep 0.3
done < "$SITES_FILE"
```

Website	CDN Provider
google.com	Google Global CDN
github.com	No CDN / Direct Hosting
wikipedia.org	No CDN / Direct Hosting
youtube.com	Google Global CDN

Website	CDN Provider
instagram.com	Meta Edge CDN
facebook.com	Meta Edge CDN
messenger.com	Meta Edge CDN
apple.com	No CDN / Direct Hosting
ani.gamer.com.tw	Cloudflare
chatgpt.com	Cloudflare

5. Network Performance Monitoring

```
#!/bin/bash
SITES_FILE="sites.txt"
OUTPUT="task5_network_results.csv"

echo "Website,Avg_Latency(ms),Packet_Loss(%),Download_Throughput(Mbps)" > "$OUTPUT"

while read -r site; do
  [ -z "$site" ] && continue

  ip4=$(dig +short A "$site" | head -n 1)
  ip6=$(dig +short AAAA "$site" | head -n 1)

  target_ip=""
  ping_cmd=""
  if [ -n "$ip4" ]; then
    target_ip="$ip4"
    ping_cmd="ping -c 5 $target_ip"
  elif [ -n "$ip6" ]; then
    target_ip="$ip6"
    ping_cmd="ping6 -c 5 $target_ip"
  else
    echo "$site,N/A,N/A,N/A" >> "$OUTPUT"
    continue
  fi

  # ---- Latency / Loss ----
  ping_out=$($ping_cmd 2>/dev/null)
  packet_loss=$(echo "$ping_out" | grep -Eo '[0-9]+(\.[0-9]+)?% packet loss' | awk '{print $1}')
  [ -z "$packet_loss" ] && packet_loss="N/A"

  rtt_line=$(echo "$ping_out" | grep -E 'round-trip|rtt')
  if [ -n "$rtt_line" ]; then
    avg_latency=$(echo "$rtt_line" | awk -F'/' '{print $5}')
  else
    avg_latency="N/A"
  fi

  speed_bytes=$(curl -o /dev/null -L --silent --write-out '%{speed_download}\n' "https://$site")
  if [ -z "$speed_bytes" ]; then
    throughput="N/A"
  else
    throughput=$(echo "$speed_bytes / 125000" | bc -l)  # Bytes/s → Mbps
  fi

  echo "$site,$avg_latency,$packet_loss,$throughput" >> "$OUTPUT"
  sleep 0.4
done < "$SITES_FILE"
```

Website	Avg_Latency(ms)	Packet_Loss(%)	Download_Throughput(Mbps)
google.com	42.38	0.0%	0.308112

Website	Avg_Latency(ms)	Packet_Loss(%)	Download_Throughput(Mbps)
github.com	50.988	0.0%	10.1922
wikipedia.org	76.371	0.0%	1.02288
youtube.com	19.237	0.0%	12.097
instagram.com	18.307	0.0%	6.80838
facebook.com	306.074	0.0%	0.757952
messenger.com	147.171	0.0%	0.856728
apple.com	61.853	0.0%	2.99702
ani.gamer.com.tw	58.136	0.0%	10.4907
chatgpt.com	21.609	0.0%	0.605376

6. Network Routing Path Analysis

```
traceroute to goolge.com (142.250.196.196), 64 hops max, 40 byte packets
 1  192.168.68.1 (192.168.68.1)  13.576 ms  14.440 ms  9.648 ms
 2  192.168.1.1 (192.168.1.1)  10.498 ms  14.162 ms  10.356 ms
 3  * * *
 4  168-95-104-170.clpy-3331.hinet.net (168.95.104.170)  40.498 ms  51.315 ms  52.342 ms
 5  220-128-8-114.tyfo-3031.hinet.net (220.128.8.114)  49.548 ms
    220-128-8-38.tyfo-3031.hinet.net (220.128.8.38)  26.406 ms  17.687 ms
 6  220-128-8-17.tyfo-3335.hinet.net (220.128.8.17)  18.676 ms * *
 7  142.250.169.122 (142.250.169.122)  73.181 ms
    142.250.169.120 (142.250.169.120)  25.089 ms  22.027 ms
 8  * * 192.178.106.71 (192.178.106.71)  154.649 ms
 9  209.85.142.120 (209.85.142.120)  66.225 ms
    209.85.245.64 (209.85.245.64)  187.776 ms
    142.251.77.85 (142.251.77.85)  22.257 ms
10  nctsaa-ac-in-f4.1e100.net (142.250.196.196)  23.943 ms
    192.178.105.254 (192.178.105.254)  24.087 ms  22.820 ms
```

Hop	IP Address	Hostname	ISP / Organization	Country	Location	Avg Latency (ms)
1	192.168.68.1	Local Router	Local Network	N/A	Home LAN	~13
2	192.168.1.1	ISP Gateway	Local ISP	N/A	ISP Edge	~10
3	*	—	Router does not respond to ICMP	—	—	—
4	168.95.104.170	clpy-3331.hinet.net	Chunghwa Telecom (HiNet)	Taiwan	Regional router	~45
5	220.128.8.114 / 220.128.8.38	tyfo-3031.hinet.net	Chunghwa Telecom Backbone	Taiwan	Taiwan backbone core	~20–50
6	220.128.8.17	tyfo-3335.hinet.net	HiNet International Gateway	Taiwan	International exit router	~18
7	142.250.169.120 / 142.250.169.122	*.google.com	Google AS15169 Global Network	APAC	Google Edge PoP	~22
8	192.178.106.71	google backbone transit	Google Backbone	APAC	Internal transit hop	~150
9	209.85.142.120 / 142.251.77.85	google backbone transit	Google Backbone	Global	Inter-region routing	22–187
10	142.250.196.196	google.com (Final Server)	Google Service Node	APAC	Final destination	~23

```
Your Computer / LAN
 ↓
Home Router (192.168.68.1)
 ↓
ISP Gateway (192.168.1.1)
 ↓
HiNet Local Router (168.95.104.170)
 ↓
HiNet Backbone Node (220.128.8.38)
 ↓
HiNet International Gateway (220.128.8.17)
 ↓
Google APAC Edge PoP (142.250.169.120)
 ↓
Google Backbone Transit (192.178.106.71)
 ↓
Google Service Node (142.250.196.196)
```

7. Backend Server Investigation

```bash
#!/bin/bash
SITES="sites.txt"
OUTPUT="task7_server_results.csv"

echo "Website,Server" > $OUTPUT

while read site; do
    if [ -z "$site" ]; then
        continue
    fi

    SERVER=$(curl -I -s https://$site | grep -i "^server:" | awk -F ': ' '{print $2}' | tr -d '\r')

    if [ -z "$SERVER" ]; then
        SERVER=$(curl -I -s http://$site | grep -i "^server:" | awk -F ': ' '{print $2}' | tr -d '\r')
    fi

    if [ -z "$SERVER" ]; then
        SERVER="Unknown"
    fi

    echo "$site,$SERVER" >> $OUTPUT
done < "$SITES"
```

Website	Server
google.com	gws
github.com	github.com
wikipedia.org	mw-web.codfw.migration-5586c8d64d-kj52p
youtube.com	ESF
instagram.com	proxygen-bolt
facebook.com	proxygen-bolt
messenger.com	proxygen-bolt
apple.com	Unknown
ani.gamer.com.tw	cloudflare
chatgpt.com	cloudflare