

# EDI: First/Second Lab Report

Sepideh Hayati 515184

May 14, 2023

## Abstract

This report in part one investigates the performance of two websites, one with an Iranian domain and one with an Italian domain, using network monitoring tools such as ping, traceroute, and mtr. The objective of this report is to gain a deeper understanding of the technological infrastructure of these websites and to identify any possible errors that may be affecting their performance. In the second part, I examined the DNS servers for both domains and obtained basic information about these server domains. By comparing the results obtained from both websites, I was able to identify any significant differences in their performance and infrastructure. Overall, this report demonstrates the importance of network monitoring and DNS performance for optimizing website performance.

## 1 Monitoring

In this subject, we focused on computer Networks for lesson "*Enterprise Digital Infrastructure*", specifically on network monitoring using Linux commands in the Ubuntu distribution. Prior knowledge of the network and its infrastructures, as well as Linux commands, was helpful in better understanding the content and keeping up with the new educational material. We first reviewed the manual (*man* page) for the monitoring tools to effectively understand the concepts and usage of each command. However, before implementing the monitoring process, we designed a five-step plan to guide us in moving the monitoring process forward coherently. We then shared the numerical data and even errors encountered for analysis, and during the lab sessions, we discussed any differences in the outputs to identify the reasons behind them.

### 1.1 Methodology and experimental setup

To plan, design, and set up a monitoring project, it is necessary to specify the required details in five parts consisting of "*why, which/who, what, where, and how*". These steps may vary for different monitoring tools, and for each part, I have defined related plans and scenarios corresponding to the relevant tool.

I have chosen two websites with different domains, *it* and *ir* so that we will be able to compare some parameters. For the purpose of this implementation, a personal laptop was used, and the *Ubuntu 22.10 Linux distribution* was installed using *Oracle VM VirtualBox Manager 7.0.4*. However, due to some issues with working with certain monitoring tools in Ubuntu, the command prompt (cmd) in *Windows 11* was used instead; for example, to get the result for *tracert*. Furthermore, the monitoring was implemented over a home Wi-Fi internet connection. In cases where I used my mobile internet connection to observe the differences, I clearly indicated.

### 1.2 Experimental results

#### 1.2.1 ping

- **what is ping**

The *ping* command is used to test the reachability of a network host and measure the round-trip time (RTT) for packets sent from the source to the destination host. The output of the *ping* command provides information about the number of packets transmitted, received, lost, and the minimum, average, and maximum round-trip time.

- **scenario**

- WHY: Debugging and predicting failures

- WHICH: Server [www.carrefour.it](http://www.carrefour.it) and [www.canbo.ir](http://www.canbo.ir)
- WHAT: Reachability of the target, RTT between vantage point and target, Packet loss rate
- WHERE: Using a personal laptop with Ubuntu 22.10 Linux distribution also a home Wi-Fi internet connection.
- HOW: *ping*

## • Results

```
sepid@spdyt:~$ ping -c 10 canbo.ir
PING canbo.ir (185.255.88.212) 56(84) bytes of data:
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=1 ttl=40 time=198 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=2 ttl=40 time=201 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=3 ttl=40 time=230 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=4 ttl=40 time=211 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=5 ttl=40 time=136 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=6 ttl=40 time=159 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=7 ttl=40 time=138 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=8 ttl=40 time=162 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=9 ttl=40 time=145 ms
64 bytes from static.212.88.255.185.clients.irandns.com (185.255.88.212): icmp_seq=10 ttl=40 time=145 ms

--- canbo.ir ping statistics ---
10 packets transmitted, 9 received, 10% packet loss, time 9073ms
rtt min/avg/max/mdev = 135.928/175.564/230.267/32.932 ms
```

Figure 1: ping-www.canbo.ir

```
sepid@spdyt:~$ ping -c 10 carrefour.it
PING carrefour.it (45.60.47.6) 56(84) bytes of data:
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=1 ttl=49 time=20.5 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=2 ttl=49 time=43.3 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=3 ttl=49 time=35.7 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=4 ttl=49 time=30.6 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=5 ttl=49 time=49.5 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=6 ttl=49 time=58.9 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=7 ttl=49 time=62.8 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=8 ttl=49 time=70.6 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=9 ttl=49 time=29.6 ms
64 bytes from 45.60.47.6 (45.60.47.6): icmp_seq=10 ttl=49 time=84.5 ms

--- carrefour.it ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9046ms
rtt min/avg/max/mdev = 20.536/48.600/84.511/19.358 ms
```

Figure 2: ping-www.carrefour.it

The results for “*ping www.canbo.ir*”, shows that (F1):

- 10 packets were transmitted, but only 9 were received, so there is a 10 percent packet loss.
- The average round-trip time was 175.564 ms, with a minimum of 135.928 ms and a maximum of 230.267 ms.

The results for “*ping www.carrefour.it*”, shows that (F2):

- All 10 packets were transmitted and received with no packet loss.
- The average round-trip time was 48.600 ms, with a minimum of 20.536 ms and a maximum of 84.511 ms.

## • Conclusion

Overall, as you can see, the results show that there is a latency between the device and the website “*canbo.ir*” also the 10 percent packet loss shows that some packets are not being successfully transmitted, which could indicate a problem with the network or the website’s server. These differences could be due to various factors, such as the location of the host server, network congestion, and network infrastructure.

### 1.2.2 tracer

#### • what is *tracert*

*tracert* (“traceroute”) is a monitoring tool used to trace the route taken by an IP network on its way to a given host. It helps to identify the network latency, packet loss, and routing errors by showing the IP addresses of the routers through the path that the packets pass, and the time taken to reach each router.

- **scenario**

- WHY: Discover network topology, loops, misconfigurations, malfunctioning routers
- WHICH: Server [www.carrefour.it](http://www.carrefour.it) and [www.canbo.ir](http://www.canbo.ir)
- WHAT: Path between vantage point and target, RTTs between vantage point and all intermediate, and between vantage point and target
- WHERE: Using a personal laptop with Win 11 command prompt (cmd) also a home Wi-Fi internet connection.
- HOW: *tracert*

As I encountered some problems installing *traceroute* on Ubuntu, I ran this command on Windows.

- **Results**

```
C:\Windows\System32>tracert www.canbo.ir

Tracing route to canbo.ir [185.255.88.212]
over a maximum of 30 hops:

  1  2 ms  2 ms  5 ms  192.168.200.1
  2  43 ms  2 ms  2 ms  192.168.178.1
  3   7 ms  9 ms  6 ms  82.113.192.132
  4 128 ms  6 ms  7 ms  185.25.74.109
  5   8 ms  9 ms  9 ms  te-0-2-0-1.asr9kp-jn.network.twt.it [82.113.194.246]
  6  10 ms  9 ms  7 ms  81.25.202.185
  7  12 ms  8 ms  7 ms  ae-1.r21.mlanit02.it.bb.gin.ntt.net [129.250.3.160]
  8  16 ms 19 ms 16 ms  ae-6.r21.frnkge13.de.bb.gin.ntt.net [129.250.3.183]
  9  24 ms 23 ms 35 ms  ae-7.r21.amstnl07.nl.bb.gin.ntt.net [129.250.3.77]
 10  21 ms 23 ms 24 ms  ae-1.a00.amstnl07.nl.bb.gin.ntt.net [129.250.7.71]
 11  *      *      *      Request timed out.
 12  *      *      *      Request timed out.
 13  *      *      *      Request timed out.
 14  *      *      *      Request timed out.
 15 106 ms 102 ms 102 ms static.212.88.255.185.clients.irandns.com [185.255.88.212]

Trace complete.
```

Figure 3: tracert-www.canbo.ir

```
C:\Windows\System32>tracert www.carrefour.it

Tracing route to www.carrefour.it.cdn.cloudflare.net [104.16.50.34]
over a maximum of 30 hops:

  1  2 ms  2 ms  2 ms  192.168.200.1
  2   5 ms  5 ms  3 ms  192.168.178.1
  3  78 ms  8 ms  6 ms  82.113.192.132
  4 2183 ms 10 ms  8 ms  185.25.74.93
  5   8 ms  8 ms 10 ms  100.be100.ncs2-jn.network.twt.it [185.25.74.66]
  6   8 ms  7 ms  8 ms  hu-0-0-1-2.ncs55-2-jn.network.twt.it [185.25.74.134]
  7  10 ms 10 ms  7 ms  hu-49.7280-mix1.network.twt.it [185.25.74.138]
  8   9 ms  9 ms  7 ms  cloudflare.mix-it.net [217.29.66.167]
  9   8 ms  7 ms  7 ms  188.114.100.11
 10   9 ms  9 ms  7 ms  104.16.50.34

Trace complete.
```

Figure 4: tracert-www.carrefour.it

The results for "*tracert www.canbo.ir*", shows that (F3):

- The packets reach the destination in 19 hops, passing through routers belonging to Iran Data Communication Company.
- With varying latencies, including some packet loss and long delays at some hops.

The results for "tracert www.carrefour.it", shows that (F4):

- The packets reach the destination in 10 hops, passing through the routers belonging to Cloudflare.
- The average latency of around 12 milliseconds.

#### • Conclusion

Overall, the results show that the two websites are hosted on different networks and have different routing paths. The routing path for "www.canbo.ir" also includes some long delays and packet loss, indicating possible network congestion or connectivity issues.

### 1.2.3 mtr

#### • scenario

- WHY: Monitoring network connectivity problems
- WHICH: Server [www.carrefour.it](http://www.carrefour.it) and [www.canbo.ir](http://www.canbo.ir)
- WHAT: Measuring the network performance metrics including latency, jitter, packet loss rate, and hop count between the vantage point and the target servers.
- WHERE: Using a personal laptop with Win 11 command prompt (cmd) also a home Wi-Fi internet connection.
- HOW: *mtr*

#### • what is mtr command

*mtr* command combines the functionality of the *traceroute* and *ping* programs in a single network diagnostic tool. The output display the hostname or IP address of each network hop in the path to the target, the percentage of packet loss, the number of packets sent to each hop, the time taken for the last packet to reach each hop, the average time taken for all packets to reach each hop, the best time taken for a packet to reach each hop, the worst time taken for a packet to reach each hop.

#### • Results

Host		Packets		Plngs				
		Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	AS7777 gateway (10.0.2.2)	0.0%	221	0.5	0.4	0.2	1.2	0.1
2.	AS7777 192.168.200.1 (192.168.200.1)	2.7%	221	5.3	50.5	2.6	4021	341.5
3.	AS7777 192.168.178.1 (192.168.178.1)	1.4%	221	7.6	64.8	3.2	3952	359.0
4.	AS30848 82.113.192.132 (82.113.192.132)	0.9%	221	24.8	64.0	6.9	3946	356.7
5.	AS30848 185.25.74.93 (185.25.74.93)	5.0%	221	10.5	70.6	7.6	3854	352.7
6.	AS30848 100.be100.ncs2-jn.network.twt.it (185.25.74.66)	0.9%	221	26.2	63.8	7.9	3763	334.7
7.	AS30848 hu-0-0-1-2.ncs55-2-jn.network.twt.it (185.25.74.134)	1.8%	220	22.3	64.1	7.8	3671	325.9
8.	AS30848 hu-49-7208-mlx1.network.twt.it (185.25.74.138)	2.7%	220	19.9	81.3	7.1	4545	436.3
9.	AS7777 cloudflare.mlx1.it.net (217.29.66.107)	2.3%	220	27.2	86.0	7.5	4500	429.0
10.	AS13335 188.114.100.11 (188.114.100.11)	2.3%	220	64.5	83.1	8.5	4409	417.4
11.	AS13335 104.16.50.34 (104.16.50.34)	4.1%	220	27.4	76.1	7.3	4259	402.8

Figure 5: mtr-www.carrefour.ir

Host		Packets		Plngs				
		Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	AS7777 gateway (10.0.2.2)	0.0%	57	0.4	0.6	0.3	5.4	0.7
2.	AS7777 192.168.200.1 (192.168.200.1)	0.0%	57	3.5	80.9	3.0	2524	379.8
3.	AS7777 192.168.178.1 (192.168.178.1)	0.0%	57	4.6	81.7	4.0	2478	371.4
4.	AS30848 82.113.192.132 (82.113.192.132)	0.0%	57	43.5	86.2	7.2	2449	364.4
5.	AS30848 185.25.74.109 (185.25.74.109)	10.5%	57	11.7	91.6	8.0	2413	376.5
6.	AS30848 hu-0-0-1-2.ncs55-1-jn.network.twt.it (185.25.74.130)	0.0%	57	9.2	80.2	8.1	2363	347.5
7.	AS3257 xe-0-2-3-cr2-ml3.lp4.gtt.net (46.33.66.5)	0.0%	57	13.2	80.6	8.0	2337	342.9
8.	AS3257 et-8-3-0-cr0-sof1.lp4.gtt.net (213.200.113.46)	1.8%	57	37.0	99.6	32.5	2285	332.9
9.	AS3257 lp4.gtt.net (46.33.72.22)	0.0%	57	42.8	161.8	40.4	3297	530.1
10.	(waiting for reply)							
11.	AS29049 85.132.90.158 (85.132.90.158)	0.0%	57	85.2	199.6	84.1	3232	505.7
12.	(waiting for reply)							
13.	(waiting for reply)							
14.	(waiting for reply)							
15.	(waiting for reply)							
16.	(waiting for reply)							
17.	(waiting for reply)							
18.	(waiting for reply)							
19.	(waiting for reply)							
20.	AS61173 static.212.88.255.185.clients.trandns.com (185.255.88.212)	0.0%	56	100.8	183.4	96.3	2732	402.8

Figure 6: mtr-www.canbo.it

In this part, the command that is used to monitor is `mtr -b -z www.carrefour.it`, it sends a series of ICMP packets to the target host (`www.carrefour.it` or `www.canbo.it`) to determine the network path and measure network performance along the way. In this command that I have used for monitoring, the "-b" option enables batch mode, which continuously updates the results in the terminal, and the "-z" option is the display of numerical IP addresses in the output instead of domain names.

The results for "`mtr -b -z www.carrefour.it`" shows that (F5),

- There are some hostnames named "AS???". It means the IP address associated with that hop cannot be resolved to a hostname. Instead, the output will display the Autonomous System (AS) number associated with that hop
- The packets reach the destination in 11 hops, including an average 2.19 percent packet loss.

The results for "`mtr -b -z www.canbo.it`" shows that (F6),

- It appears that 20 hops have been made. However, you can see in some lines where the hops are listed, the message "*waiting for replay*" is displayed (9 rows). This shows that there is a problem with the connection at those hops, and they should not be counted in the total number of hops. Therefore, the total number of hops passed is 11.
- The packets reach the destination in 11 hops, including an average 1.07 percent packet loss.

- **Conclusion**

Overall, based on the output of the "`mtr -b -z`" command, we saw that both websites had hosts with the name AS (Autonomous System) and experienced packet loss. "`www.carrefour.it`" passed through 11 hops, and "`www.canbo.it`" also went through 11 hops but with 9 waiting for a reply as I mentioned above. For both websites, the average response time (avg) and the worst response time (wrst) were similar, but for Carrefour, better response times were observed in the last and best columns. As a whole, the Italian domain appears to have performed better in terms of responsiveness.

## 1.2.4 Speedtest

- **what is speedtest**

The *speedtest* command provides a quick and easy way to measure the network performance of an Internet connection. It measures the download and uploads speed, latency, jitter, and packet loss rate.

- **scenario**

- WHICH: using a personal laptop with Ubuntu 22.10 Linux distribution and a home Wi-Fi internet connection.
- WHERE: The personal laptop
- WHY: To diagnose network issues, identify performance bottlenecks, predict potential failures, and measure the speed and quality of an internet connection
- WHAT: Network speed and performance, download and upload speed, latency, jitter, packet loss rate
- HOW: *speedtest*

- **Results**

The two sets of results are from two different types of connections: the first one is over WiFi (with an access point in my Collegio) and the second one is over my mobile network (Telecom Italia Mobile-TIM).

In the first test over WiFi, the results for "`speed-cli`" shows that (F7),

- The download speed is 80.59 Mbit/s and the upload speed is 65.07 Mbit/s.
- The test was performed from TWT S.p.A. and the best server was selected based on a ping of 16.589 ms.
- The server was hosted by SWITCH in Zurich, which is approximately 220.45 km away from Pavia, SAN.GIOVANNI, where I am.

In the second test over my mobile network (TIM), the results for "*speed-cli*" shows that,

- The download speed is 32.28 Mbit/s and the upload speed is 18.05 Mbit/s.
- The test was performed from Telecom Italia Mobile and the best server was selected based on a ping of 50.332 ms.
- The server was hosted by iway AG in Zurich, which is also approximately 220.45 km away from Pavia, SAN.GIOVANNI, where I am.

```
sepideh@spdhyt:~$ speedtest-cli
Retrieving speedtest.net configuration...
Testing from TIM S.p.A. (185.90.68.63)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by SWITCH (Zurich) [220.45 km]: 16.589 ms
Testing download speed.....
Download: 80.59 Mbit/s
Testing upload speed.....
Upload: 65.07 Mbit/s
sepideh@spdhyt:~$ speedtest-cli
Retrieving speedtest.net configuration...
Testing from Telecom Italia Mobile (5.171.97.34)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by iway AG (Zürich) [220.45 km]: 50.332 ms
Testing download speed.....
Download: 32.28 Mbit/s
Testing upload speed.....
Upload: 18.05 Mbit/s
```

Figure 7: speedtest-My system

#### • Conclusion

Overall, Compared to the WiFi test, the test over my mobile network (TIM) has lower download and upload speeds, indicating that the mobile network connection is slower than the WiFi connection. The ping time is also higher in the mobile network, which means that the connection is slower to respond.

### 1.2.5 top

#### • what is top

The *top* command provides a dynamic real-time view of a running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel.

#### • scenario

- WHY: Monitoring system resource usage and identifying processes that may be consuming excessive resources.
- WHICH: Linux server on my laptop
- WHAT: CPU and memory usage of the database process
- WHERE: Using a personal laptop with Ubuntu 22.10 Linux distribution also a home Wi-Fi internet connection.
- HOW: *top*

#### • Results

As shown in the highlighted border on the image (F8),

- During the execution of the *top* command, only one user, *Sepideh*, was logged in to the system
- In the second line about tasks, you can see 233 tasks on the system, which includes both running and non-running tasks, 1 running task, 232 sleeping, 0 stopped, and 0 zombie (processes that have finished but not been cleaned up)
- The next line shows the percentage of CPU usage categorized into different types of activities.
- In the last column, the names of running programs that are using a percentage of the CPU can be seen. As noticeable in this column, during the execution of the *top* command, I was running *mtr* and *ping* commands on the Ubuntu terminal.

```
top - 09:41:10 up 37 min, 1 user, load average: 0.19, 0.23, 0.24
Tasks: 233 total, 1 running, 232 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.3 sy, 0.0 ni, 99.5 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
Mem: 1974.8 total, 155.7 free, 1210.6 used, 608.4 buff/cache
Swap: 0 total, 0 free, 0 used, 0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1747	sepideh	20	0	4878800	248112	64240	S	3.3	12.3	4:52.44	gnome-shell
2325	sepideh	20	0	572952	37528	23408	S	0.7	1.9	0:27.80	gnome-terminal-
2239	sepideh	20	0	227148	244	0	S	0.3	0.0	0:01.92	VBoxClient
2247	sepideh	20	0	227664	308	64	S	0.3	0.0	0:04.37	VBoxClient
2252	sepideh	20	0	160964	788	560	S	0.3	0.0	0:00.25	VBoxClient
4592	root	20	0	0	0	0	I	0.3	0.0	0:00.07	kworker/2:2-mm_percpu_wq
4639	sepideh	20	0	6760	5224	2312	S	0.3	0.3	0:00.18	mtr
4760	sepideh	20	0	22020	4116	3236	R	0.3	0.2	0:00.09	top
1	root	20	0	169468	9860	4676	S	0.0	0.5	0:03.51	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
14	root	20	0	0	0	0	S	0.0	0.0	0:00.08	ksoftirqd/0
15	root	20	0	0	0	0	I	0.0	0.0	0:01.48	rcu_preempt
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/0
17	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
21	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1
22	root	rt	0	0	0	0	S	0.0	0.0	0:00.17	migration/1
23	root	20	0	0	0	0	S	0.0	0.0	0:00.10	ksoftirqd/1
25	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri
26	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/2
27	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/2
28	root	rt	0	0	0	0	S	0.0	0.0	0:00.17	migration/2
29	root	20	0	0	0	0	S	0.0	0.0	0:00.08	ksoftirqd/2
31	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/2:0H-events_highpri
32	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/3
33	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/3

Figure 8: top-My system

## • Conclusion

Overall, the results show that most processes are sleeping and there are no stopped or zombie processes. The system has 155.7 MiB of free memory and 608.4 MiB of memory used for buffer/cache. The top process in terms of CPU usage is *gnome-shell*, which is using 3.3 percent of the CPU.

## 1.2.6 perfmon

### • what is perfmon

*Perfmon* (Performance Monitor) is a tool available in Windows operating systems that allows us to monitor the performance of our computer. It provides real-time monitoring of various performance metrics, including CPU usage, memory usage, disk usage, network activity, and more.

### • Results

As shown in the highlighted border on the pictures,

- I selected some parameters such as *Physical Disk*, *Memory*, *Network Interface*, *Processor Information* and *processor (User Time and Idle Time)* to monitor using the *perfmon* tool.
- As shown in the graph at 10:37:00 am (F9), I opened VirtualBox to use Ubuntu. After launching Ubuntu, I used some commands such as *top*, *mtr*, and *ping* in the terminal. As depicted in the graph, as soon as Linux was launched via VirtualBox, the "*Committed Byte in Use*" item, which is related to *Memory*, and the "*Byte Total/sec*" parameter, which is related to the *Network Interface*, started to increase.



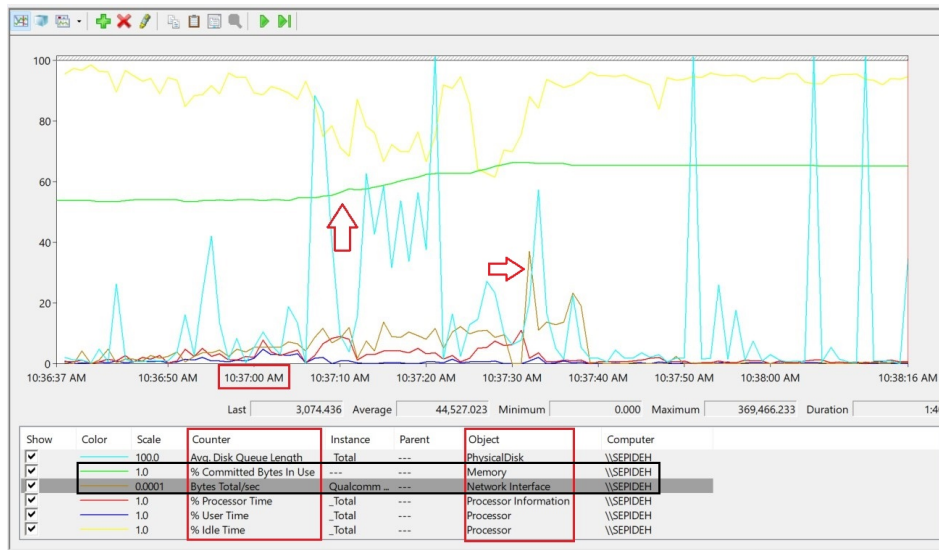


Figure 9: Windows Performance Monitor-My system 1

- After finishing using at 10:43:40 am, Linux and closing VirtualBox, these two lines experienced a downward slope in the graph.(F10)

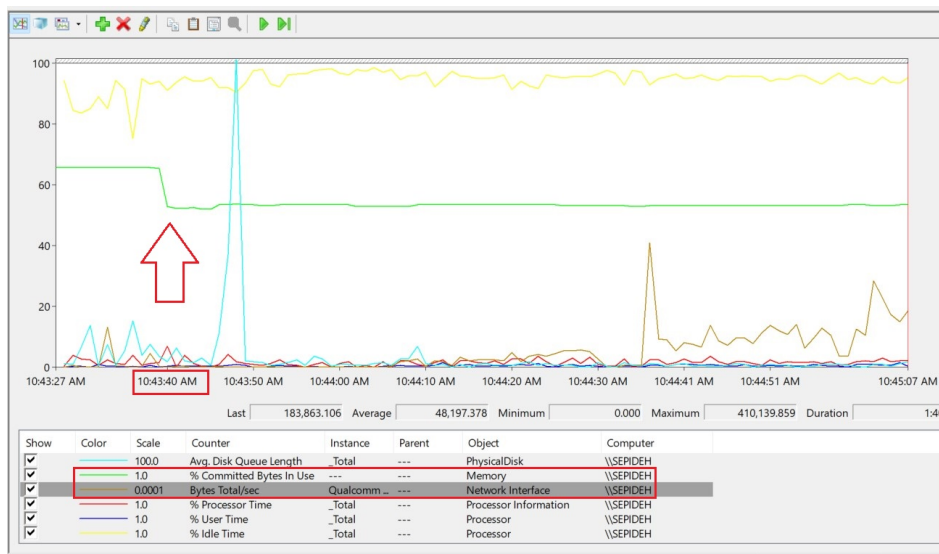


Figure 10: Windows Performance Monitor-My system 2

## 2 Doman Name System

In the DNS laboratory sessions, we utilized various name servers and examined the results of the DNS queries as well as the time taken to perform them. Additionally, the process of finding domain information such as name servers and identifying local servers was also explored throughout the series of DNS lab sessions.

### 2.1 Methodology and experimental setup

I used Ubuntu Linux and a Wi-Fi internet connection to run DNS-related commands. The command I used to access domain information such as name servers is "dig".



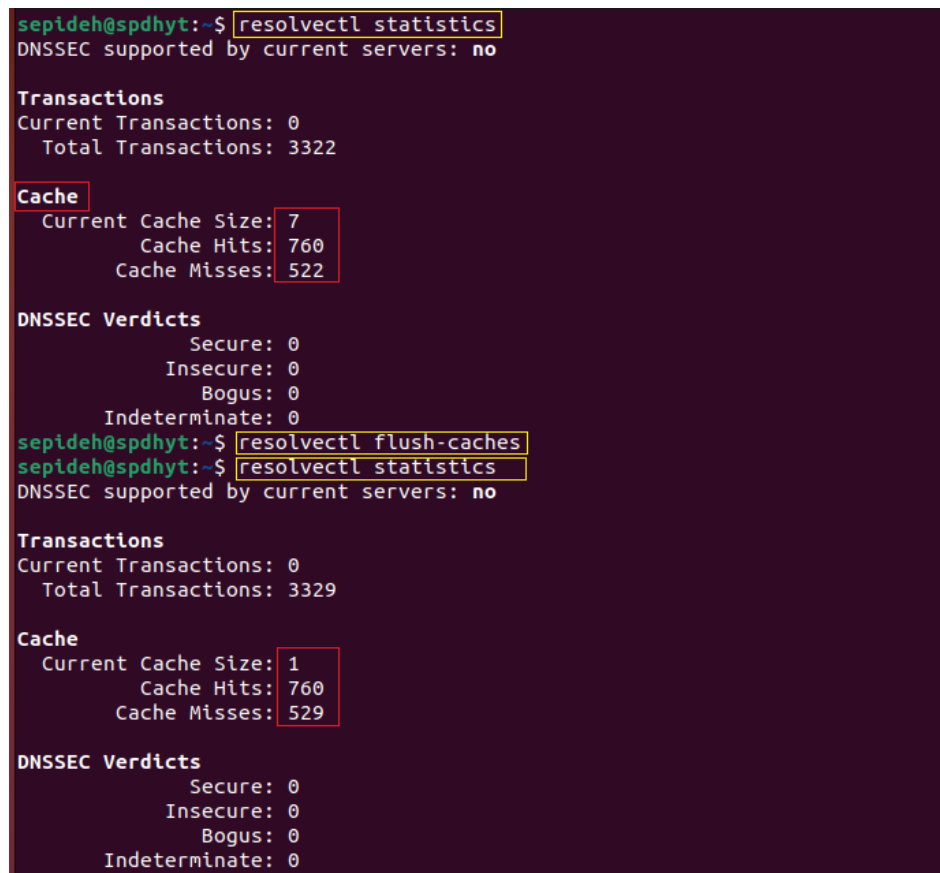
In this experiment, I examined DNS queries for several different websites, some with Italian domains and others for other countries.

## 2.2 Experimental results

### 2.2.1 Active experiments

First of all, I flushed the cache of my system, "*resolvectl flush-caches*" (F11).

- The first command "*resolvectl statistics*" displays the current DNS statistics on my system. In this picture, the "Cache" section displays the current size of the DNS cache on my system, as well as the number of cache hits and misses. In this case, the cache size is 7, and there have been 760 cache hits and 522 cache misses.
- Then I flushed my cache by using the second command "*resolvectl flush-caches*", which means all cached DNS responses will be deleted.
- After flushing, the cache section shows that the cache size has decreased to 1, and there have been 760 cache hits and 529 cache misses.

A terminal window with a dark purple background. The user 'sepideh' is at host 'spdhyt'. The first command is 'resolvectl statistics', which shows DNSSEC support as 'no', 0 current transactions, 3322 total transactions, a cache size of 7, 760 hits, and 522 misses. The second command is 'resolvectl flush-caches'. The third command is 'resolvectl statistics', which shows the same transaction counts but the cache size is now 1, with 760 hits and 529 misses. Red boxes highlight the 'Cache' section in both outputs, and yellow boxes highlight the commands.

```
sepideh@spdhyt:~$ resolvectl statistics
DNSSEC supported by current servers: no

Transactions
Current Transactions: 0
Total Transactions: 3322

Cache
Current Cache Size: 7
Cache Hits: 760
Cache Misses: 522

DNSSEC Verdicts
Secure: 0
Insecure: 0
Bogus: 0
Indeterminate: 0

sepideh@spdhyt:~$ resolvectl flush-caches
sepideh@spdhyt:~$ resolvectl statistics
DNSSEC supported by current servers: no

Transactions
Current Transactions: 0
Total Transactions: 3329

Cache
Current Cache Size: 1
Cache Hits: 760
Cache Misses: 529

DNSSEC Verdicts
Secure: 0
Insecure: 0
Bogus: 0
Indeterminate: 0
```

Figure 11: cache flush

- the IP addresses of "*achille.unipv.it*" and another hostname TLD *.it*

To query the DNS for the IP addresses of "*achille.unipv.it*" and another hostname in the TLD domain *.it*, I have used the *dig* command in the Ubuntu terminal. *dig* is a tool for querying DNS. It uses to obtain information about domain names, such as IP addresses associated with a domain name, name server information, and time-to-live (TTL).

The results for "*dig achille.unipv.it*" (F12), and "*dig enel.it*" (F13) shows that,

```

sepideh@spdhyt:~$ dig achille.unipv.it

; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> achille.unipv.it
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59474
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;achille.unipv.it.                IN      A

;; ANSWER SECTION:
achille.unipv.it.                300     IN      A      193.204.34.164

;; Query time: 20 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fri May 12 06:30:04 CEST 2023
;; MSG SIZE rcvd: 61

```

Figure 12: dig-achille.unipv.it

```

sepideh@spdhyt:~$ dig enel.it

; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> enel.it
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38434
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;enel.it.                        IN      A

;; ANSWER SECTION:
enel.it.                        7200    IN      A      192.230.80.63
enel.it.                        7200    IN      A      192.230.83.63

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fri May 12 06:45:31 CEST 2023
;; MSG SIZE rcvd: 68

```

Figure 13: dig-enel.it

- As you can see in the pictures, the query time for "achille.unipv.it" was 20 msec, while the query time for "enel.it" was 12 msec. This shows that the DNS server for "enel.it" responded faster than the DNS server for "achille.unipv.it".
  - The "achille.unipv.it" query returned one IP address, while the "enel.it" query returned two IP addresses. This means that "enel.it" has multiple IP addresses, that can be used to load balance traffic or provide redundancy.
  - The TTL of the "achille.unipv.it" was 300, which means that the DNS can be cached by a resolver for 300 seconds before it expires. Also, the TTL value for the "enel.it" query was 7200.
  - As you can see in the picture, the dig command for both domains didn't return any authoritative answer, because the authority value is zero, therefore I did not receive an official answer. This means that the DNS server for the domain itself responded to the query and the answer was obtained from a cache or by forwarding a query to another DNS.
- **The name(s) of the mail servers associated with the domain "universitadipavia.it" and "harvard.edu"**  
In this case, the specific RR (Resource Record) associated with the query is the MX (Mail Exchange) Record type. MX records are used to identify the mail servers that are responsible for handling incoming emails for the domain. Therefore I have used "dig MX universitadipavia.it" and "dig MX harvard.edu".

The results for "dig MX universitadipavia.it" (F14), and "dig MX harvard.edu" (F15) shows that,

- In the case of "universitadipavia.it", there is a list of 7 MX RRs (google.com) and for "harvard.edu", there are only 2 MX RRs (pphosted.com) listed. Each mail server has a different priority level, but for "harvard.edu" you can see the same priority level of 100, this means that incoming emails to "harvard.edu" will be handled by either of these two servers, with no preference for one over the other.
- There is no authoritative or official answer.
- To obtain the IP addresses of each DNS Mail server, we can use the command "dig -t A aspmx.google.com". For example, by running "dig -t A aspmx.google.com", I received the IP address; 64.233.167.27.

```
sepideh@spdhyt:~$ dig MX universitadipavia.it

; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> MX universitadipavia.it
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31418
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;universitadipavia.it.      IN      MX

;; ANSWER SECTION:
universitadipavia.it.  3569    IN      MX      1 aspmx.l.google.com.
universitadipavia.it.  3569    IN      MX      10 aspmx2.googlemail.com.
universitadipavia.it.  3569    IN      MX      5 alt2.aspmx.l.google.com.
universitadipavia.it.  3569    IN      MX      10 aspmx3.googlemail.com.
universitadipavia.it.  3569    IN      MX      10 aspmx5.googlemail.com.
universitadipavia.it.  3569    IN      MX      5 alt1.aspmx.l.google.com.
universitadipavia.it.  3569    IN      MX      10 aspmx4.googlemail.com.

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fri May 12 12:45:28 CEST 2023
;; MSG SIZE rcvd: 228
```

Figure 14: dig MX universitadipavia.it

```
sepideh@spdhyt:~$ dig MX harvard.edu

; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> MX harvard.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55691
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;harvard.edu.             IN      MX

;; ANSWER SECTION:
harvard.edu.             180     IN      MX      100 mx0a-00171101.pphosted.com.
harvard.edu.             180     IN      MX      100 mx0b-00171101.pphosted.com.

;; Query time: 28 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fri May 12 14:24:29 CEST 2023
;; MSG SIZE rcvd: 112
```

Figure 15: dig MX harvard.edu

- The IP address of a Web server located outside Europe

- In this case, first of all, I used the command `"dig snapp.ir"` to obtain the IP address of a web server located outside Europe. As shown in the output, there are two A records associated with `"snapp.ir"`. However, in the authoritative section, the value is zero, indicating that the response is not authoritative. (F16)

```
sepid@spdh:~$ dig snapp.ir

; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> snapp.ir
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5719
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;snapp.ir.                IN      A

;; ANSWER SECTION:
snapp.ir.                 60      IN      A      185.143.234.120
snapp.ir.                 60      IN      A      185.143.233.120

;; Query time: 28 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fri May 12 20:28:39 CEST 2023
;; MSG SIZE rcvd: 69
```

Figure 16: dig Snapp.ir

- After obtaining the IP address, I used the command `"dig +dnssec snapp.ir +trace"`. This command is used to perform a DNS lookup and also trace the DNS query from the root name servers to the authoritative name servers for the domain. The response to this query shows 2 A records and 13 NS records. (F17)
- By using the `+trace` option in the command `"dig+dnssec snapp.ir +trace"`, we can obtain information about the name server domain, which can vary across different domains.

```
sepid@spdh:~$ dig +dnssec snapp.ir +trace

; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> +dnssec snapp.ir +trace
;; global options: +cmd
.                518400   IN      NS      a.root-servers.net.
.                518400   IN      NS      b.root-servers.net.
.                518400   IN      NS      c.root-servers.net.
.                518400   IN      NS      d.root-servers.net.
.                518400   IN      NS      e.root-servers.net.
.                518400   IN      NS      f.root-servers.net.
.                518400   IN      NS      g.root-servers.net.
.                518400   IN      NS      h.root-servers.net.
.                518400   IN      NS      i.root-servers.net.
.                518400   IN      NS      j.root-servers.net.
.                518400   IN      NS      k.root-servers.net.
.                518400   IN      NS      l.root-servers.net.
.                518400   IN      NS      m.root-servers.net.
;; Received 239 bytes from 127.0.0.53#53(127.0.0.53) in 20 ms

snapp.ir.         0        IN      A      185.143.234.120
snapp.ir.         0        IN      A      185.143.233.120
;; Received 69 bytes from 199.7.83.42#53(l.root-servers.net) in 12 ms
```

Figure 17: dig dig+dnssec snapp.ir +trace

- the IP addresses of the Name Servers of a company located outside Europe

- To obtain the IP addresses of a company's Name Servers located outside Europe, I used the *dig* command in Ubuntu to execute two queries. The first query, "*dig NS nike.com*", was used to obtain a list of Name Servers associated with the company's domain name, which showed 4 Name Servers in this query and all of these Name Servers belong to "*nike.com*"[F18]. Then, I selected one of the Name Servers from the list and ran the second query, "*dig ns-n1.nike.com*", to retrieve its IP address [F19]. To obtain an authoritative answer, I also executed a third query, "*dig 205.251.197.8*", using the IP address obtained from the previous query [F20]. This helped me verify the information obtained from the Name Servers.

```
septideh@spdhyt:~$ dig NS nike.com

; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> NS nike.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26314
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;nike.com.                IN      NS

;; ANSWER SECTION:
nike.com.                 3506    IN      NS      ns-n1.nike.com.
nike.com.                 3506    IN      NS      ns-n2.nike.com.
nike.com.                 3506    IN      NS      ns-n3.nike.com.
nike.com.                 3506    IN      NS      ns-n4.nike.com.

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 11:36:53 CEST 2023
;; MSG SIZE rcvd: 117
```

Figure 18: IP on NS of nike.com-Query1

```
septideh@spdhyt:~$ dig ns-n1.nike.com

; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> ns-n1.nike.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 394
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ns-n1.nike.com.          IN      A

;; ANSWER SECTION:
ns-n1.nike.com.           300     IN      A       205.251.197.8

;; Query time: 20 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 11:37:06 CEST 2023
;; MSG SIZE rcvd: 59
```

Figure 19: IP on NS of nike.com-Query2

```

sepldeh@spdhyt:~$ dig 205.251.197.8
; <<> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<> 205.251.197.8
; global options: +cmd
; Got answer:
; ->HEADER<- opcode: QUERY, status: NXDOMAIN, id: 53866
; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 65494
; QUESTION SECTION:
; 205.251.197.8.                IN      A
;
; AUTHORITY SECTION:
.                3600    IN      SOA      a.root-servers.net. nstld.verisign-grs.com. 2023051300 1800 900 604800 86400
;
; Query time: 16 msec
; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
; WHEN: Sat May 13 11:37:19 CEST 2023
; MSG SIZE rcvd: 117

```

Figure 20: IP on NS of nike.com-Query3

- To obtain information about the primary name server, we can use the command `"dig SOA nike.com"`, which reveals that `"ns-n1.nike.com"` is the primary name server.[F21]

```

sepldeh@spdhyt:~$ dig SOA nike.com
; <<> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<> SOA nike.com
; global options: +cmd
; Got answer:
; ->HEADER<- opcode: QUERY, status: NOERROR, id: 34922
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 65494
; QUESTION SECTION:
; nike.com.                    IN      SOA
;
; ANSWER SECTION:
nike.com.        3449    IN      SOA      ns-n1.nike.com. hostmaster.nike.com. 202101501 10800 3600 1814400 900
;
; Query time: 24 msec
; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
; WHEN: Sat May 13 11:40:35 CEST 2023
; MSG SIZE rcvd: 90

```

Figure 21: dig SOA nike.com

- Used the `"whois nike.com"` command on Linux to find the name of the person who registered the domain (F), which is `"Nike, Inc"` and their email address. Additionally, I was able to determine the registration expiration date of the domain from the output; `2024-03-05`. [F22]

```

The Registry database contains ONLY .COM, .NET, .EDU domains and Registrars.
Domain Name: nike.com
Registry Domain ID: 7258 DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2023-02-28T23:48:24+0000
Creation Date: 1995-03-04T05:00:00+0000
Registrar Registration Expiration Date: 2024-03-05T00:00:00+0000
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.20860851750
Domain Status: clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
Domain Status: clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)
Domain Status: clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
Domain Status: serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)
Domain Status: serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)
Domain Status: serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)
Registry Registrant ID:
Registrant Name: Internet Domain Administrator
Registrant Organization: Nike, Inc.
Registrant Street: One Bowerman Drive, DF/4
Registrant City: Beaverton
Registrant State/Province: OR
Registrant Postal Code: 97005
Registrant Country: US

```

Figure 22: whois nike.com

- Query one of the Name Servers identified in the previous experiment to obtain the IP address of the Name Servers of the domains `unipiv.it` and `cloudflare.com`
- To obtain the IP addresses of the Name Servers for the domains `"unipiv.it"` and `"cloudflare.com"`, I used the Name Server `"ns-n2.nike.com"`, which was identified in the previous experiment. [F23]



```

sepideh@spdhyt:~$ dig nike.com NS

;<<> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<> nike.com NS
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 4847
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;nike.com.                IN      NS

;; ANSWER SECTION:
nike.com.                 3376    IN      NS      ns-n1.nike.com.
nike.com.                 3376    IN      NS      ns-n2.nike.com.
nike.com.                 3376    IN      NS      ns-n3.nike.com.
nike.com.                 3376    IN      NS      ns-n4.nike.com.

;; Query time: 16 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 14:15:39 CEST 2023
;; MSG SIZE rcvd: 117

```

Figure 23: To select one of the NS records of "nike.com"

- To retrieve the Name Servers for "unipv.it", I executed the command "dig @ns-n2.nike.com unipv.it NS"[F24], and for "cloudflare.com", I used "dig @ns-n2.nike.com cloudflare.com NS". [F25]

```

sepideh@spdhyt:~$ dig ns-n2.nike.com unipv.it NS

;<<> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<> ns-n2.nike.com unipv.it NS
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 47731
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ns-n2.nike.com.                IN      A

;; ANSWER SECTION:
ns-n2.nike.com.                 300     IN      A      205.251.198.251

;; Query time: 16 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 14:16:01 CEST 2023
;; MSG SIZE rcvd: 59

;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 42434
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;unipv.it.                     IN      NS

;; ANSWER SECTION:
unipv.it.                      104     IN      NS      ipv36.unipv.it.
unipv.it.                      104     IN      NS      ipv512.unipv.it.

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 14:16:01 CEST 2023
;; MSG SIZE rcvd: 78

```

Figure 24: To retrieve the NS for Unipv.it using Ns record of nike.com

```

septideh@spdhyt:~$ dig ns-n2.nike.com cloudflare.com NS
;; <<> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<> ns-n2.nike.com cloudflare.com NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37916
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ns-n2.nike.com.                IN      A

;; ANSWER SECTION:
ns-n2.nike.com.                300     IN      A      205.251.198.251

;; Query time: 23 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 15:46:35 CEST 2023
;; MSG SIZE rcvd: 59

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42717
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;cloudflare.com.                IN      NS

;; ANSWER SECTION:
cloudflare.com.                205     IN      NS      ns3.cloudflare.com.
cloudflare.com.                205     IN      NS      ns4.cloudflare.com.
cloudflare.com.                205     IN      NS      ns5.cloudflare.com.
cloudflare.com.                205     IN      NS      ns6.cloudflare.com.
cloudflare.com.                205     IN      NS      ns7.cloudflare.com.

;; Query time: 15 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 15:46:35 CEST 2023
;; MSG SIZE rcvd: 133

```

Figure 25: To retrieve the NS for Cloudflare using Ns record of nike.com

- After retrieving the NS records for "unipv.it" and "cloudflare.com", I selected one NS and used the command "dig NS of unipv unipv.it NS" and "dig NS of unipv unipv.it NS" to obtain the IP address of the NS for both domains. I observed one IP address for the NS of "unipv.it" [F26] and two IP addresses for "cloudflare.com" [F27], because "cloudflare.com" has configured multiple IP address for the NS "ns3.cloudflare.com"

```

septideh@spdhyt:~$ dig ipv36.unipv.it unipv.it NS
;; <<> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<> ipv36.unipv.it unipv.it NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9930
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ipv36.unipv.it.                IN      A

;; ANSWER SECTION:
ipv36.unipv.it.                125     IN      A      193.204.35.27

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 14:18:17 CEST 2023
;; MSG SIZE rcvd: 59

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10515
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;unipv.it.                      IN      NS

;; ANSWER SECTION:
unipv.it.                      164     IN      NS      ipv36.unipv.it.
unipv.it.                      164     IN      NS      ipv512.unipv.it.

;; Query time: 16 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 14:18:17 CEST 2023
;; MSG SIZE rcvd: 78

```

Figure 26: To obtain the IP address of the NS record of Unipv.it

```

septideh@spdyht:~$ dig ns3.cloudflare.com cloudflare.com NS
;; <<>> DiG 9.18.12-0ubuntu0.22.10.1-Ubuntu <<>> ns3.cloudflare.com cloudflare.com NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41966
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 65494
;; QUESTION SECTION:
;ns3.cloudflare.com.      IN      A
;; ANSWER SECTION:
ns3.cloudflare.com.      1947    IN      A      162.159.7.226
ns3.cloudflare.com.      1947    IN      A      162.159.0.33
;; Query time: 11 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 15:47:11 CEST 2023
;; MSG SIZE rcvd: 79
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21554
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 65494
;; QUESTION SECTION:
;cloudflare.com.         IN      NS
;; ANSWER SECTION:
cloudflare.com.          169     IN      NS      ns6.cloudflare.com.
cloudflare.com.          169     IN      NS      ns4.cloudflare.com.
cloudflare.com.          169     IN      NS      ns5.cloudflare.com.
cloudflare.com.          169     IN      NS      ns7.cloudflare.com.
cloudflare.com.          169     IN      NS      ns3.cloudflare.com.
;; Query time: 3 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat May 13 15:47:11 CEST 2023
;; MSG SIZE rcvd: 133

```

Figure 27: To obtain the IP address of the NS record of cloudflare

## 2.2.2 Performance experiments

- To assess the performance of DNS servers for *unipv.it* and *nike.com*, I created a DNS list named *dnsservers.txt* and used a bulk ping utility to send an arbitrary DNS query to a given list of DNS servers [F28].
- This process evaluates the performance of each DNS server in responding to the DNS query. The result of running the command "*dnseval -t A -f dnsservers.txt -c10 cloudflare*" [F29] and "*dnseval -t A -f dnsservers.txt -c10 nike.com*" [F30] is shown on the picture. This command is used to evaluate the responsiveness of DNS servers for the domain "*nike.com*" using the *dnseval* tool. So, by running this command, *dnseval* performs 10 DNS record searches for the A record type for the domain "*nike.com*" for each of the DNS servers listed in the *dnsservers.txt* file and displays the results. This allows for evaluating the quality and performance of different DNS servers in responding to DNS requests.

```

GNU nano 6.4
1. 1.1.1
8.8.8.8
205.251.198.251
193.204.35.100
162.159.7.226
162.159.0.33
193.204.35.27
1.0.0.1
4.2.2.1
4.2.2.2
4.2.2.3
4.2.2.4
9.9.9.9
8.8.4.4

```

Figure 28: To create a list of DNS servers

```
sepi@sepihyt:~$ dnseval -t A -f dns_servers.txt -c10 cloudflare.com
```

server	avg(ms)	min(ms)	max(ms)	stddev(ms)	lost(%)	ttd	flags	response
1.1.1.1	14.497	11.581	16.590	1.734	NO	52	QR -- -- RD RA -- --	NOERROR
8.8.8.8	14.996	12.390	20.627	2.994	N/A	221	QR -- -- RD RA -- --	NOERROR
205.251.198.251	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
193.204.35.100	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
162.159.7.226	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
162.159.0.33	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
193.204.35.27	16.735	11.246	28.124	4.071	NO	213	QR -- -- RD RA -- --	NOERROR
9.9.9.9	22.258	12.922	45.972	12.291	N/A	155	QR -- -- RD RA -- --	NOERROR
4.2.2.1	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
4.2.2.2	15.750	11.003	18.868	1.843	NO	51	QR -- -- RD RA -- --	NOERROR
4.2.2.3	16.148	11.312	28.204	2.493	N/A	241	QR -- -- RD RA -- --	NOERROR
4.2.2.4	14.537	12.978	19.113	2.094	N/A	207	QR -- -- RD RA -- --	NOERROR
9.9.9.9	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
8.8.4.4	15.556	12.405	21.965	2.743	NO	265	QR -- -- RD RA -- --	NOERROR

```
sepi@sepihyt:~$ nc
```

```
sepi@sepihyt:~$ dnseval -t A -f dns_servers.txt -c10 cloudflare.com
```

server	avg(ms)	min(ms)	max(ms)	stddev(ms)	lost(%)	ttd	flags	response
1.1.1.1	13.454	11.488	16.197	1.531	NO	243	QR -- -- RD RA -- --	NOERROR
8.8.8.8	13.001	10.386	17.298	2.239	NO	186	QR -- -- RD RA -- --	NOERROR
205.251.198.251	17.139	12.031	27.002	2.241	NO	59	QR -- -- RD RA -- --	NOERROR
193.204.35.100	15.815	11.682	22.150	3.203	NO	141	QR -- -- RD RA -- --	NOERROR
162.159.7.226	13.798	10.794	18.835	2.485	NO	51	QR -- -- RD RA -- --	NOERROR
162.159.0.33	11.427	10.009	15.140	1.507	NO	388	QR -- -- RD RA -- --	NOERROR
193.204.35.27	13.073	10.933	15.142	1.689	NO	279	QR -- -- RD RA -- --	NOERROR
9.9.9.9	12.997	10.948	16.688	1.723	NO	227	QR -- -- RD RA -- --	NOERROR
4.2.2.1	14.141	12.387	16.195	1.384	NO	196	QR -- -- RD RA -- --	NOERROR
4.2.2.2	14.037	11.384	17.891	2.149	NO	227	QR -- -- RD RA -- --	NOERROR
4.2.2.3	14.034	9.681	19.877	2.740	NO	8	QR -- -- RD RA -- --	NOERROR
4.2.2.4	16.448	13.411	21.110	2.362	NO	8	QR -- -- RD RA -- --	NOERROR
9.9.9.9	15.325	11.889	19.888	1.990	NO	127	QR -- -- RD RA -- --	NOERROR
8.8.4.4	13.477	8.555	16.478	2.159	NO	118	QR -- -- RD RA -- --	NOERROR

Figure 29: dnseval cloudflare.com

```
sepi@sepihyt:~$ dnseval -t A -f dns_servers.txt -c10 nike.com
```

server	avg(ms)	min(ms)	max(ms)	stddev(ms)	lost(%)	ttd	Flags	response
1.1.1.1	15.029	11.842	24.577	4.026	NO	29	QR -- -- RD RA -- --	NOERROR
8.8.8.8	13.226	11.062	18.587	2.124	NO	21	QR -- -- RD RA -- --	NOERROR
205.251.198.251	14.178	12.040	17.016	1.744	NO	42	QR -- -- RD RA -- --	NOERROR
193.204.35.100	17.282	11.346	43.584	9.457	NO	19	QR -- -- RD RA -- --	NOERROR
162.159.7.226	13.712	11.987	15.899	1.345	NO	38	QR -- -- RD RA -- --	NOERROR
162.159.0.33	16.551	10.526	26.369	5.038	NO	29	QR -- -- RD RA -- --	NOERROR
193.204.35.27	14.288	12.023	19.156	1.993	NO	24	QR -- -- RD RA -- --	NOERROR
9.9.9.9	14.254	10.998	19.628	1.116	NO	29	QR -- -- RD RA -- --	NOERROR
4.2.2.1	13.811	11.628	15.989	1.333	NO	28	QR -- -- RD RA -- --	NOERROR
4.2.2.2	14.498	11.524	17.831	1.570	NO	28	QR -- -- RD RA -- --	NOERROR
4.2.2.3	14.474	10.829	19.352	2.501	NO	41	QR -- -- RD RA -- --	NOERROR
4.2.2.4	14.580	12.230	21.272	2.555	NO	40	QR -- -- RD RA -- --	NOERROR
9.9.9.9	15.943	11.943	19.352	2.640	NO	36	QR -- -- RD RA -- --	NOERROR
8.8.4.4	15.348	11.725	24.395	1.741	NO	27	QR -- -- RD RA -- --	NOERROR

```
sepi@sepihyt:~$ dnseval -t A -f dns_servers.txt -c10 ntk.com
```

server	avg(ms)	min(ms)	max(ms)	stddev(ms)	lost(%)	ttd	Flags	response
1.1.1.1	14.045	11.288	20.349	3.734	N/A	NO	QR -- -- RD RA -- --	NOERROR
8.8.8.8	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
205.251.198.251	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
193.204.35.100	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
162.159.7.226	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
162.159.0.33	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
193.204.35.27	12.769	12.769	12.769	0.000	N/A	24	QR -- -- RD RA -- --	NOERROR
9.9.9.9	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
4.2.2.1	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
4.2.2.2	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
4.2.2.3	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
4.2.2.4	0.000	0.000	0.000	0.000	N/A	N/A	-- -- -- -- -- -- -- --	No Response
9.9.9.9	15.575	12.190	23.876	3.498	NO	16	QR -- -- RD RA -- --	NOERROR
8.8.4.4	13.808	11.934	15.458	2.009	NO	11	QR -- -- RD RA -- --	NOERROR

Figure 30: dnseval nike.com

- I used `dnstraceroute -a nike.com` to trace the path and measure the time taken by the query to reach my local NS for the domain `"nike.com"`. The `-a` flag helped me identify the autonomous systems (AS) responsible for each hop. In my case, the query took `3.13 ms` to reach my local name server. Then, I switched to the open name server `9.9.9.9` and ran `dnstraceroute -s 9.9.9.9 -a nike.com` to measure the time taken for the same query to reach this destination. The query first went to another gateway in `1.1 ms`, then it reached `"9.9.9.9"`. Thus, changing the destination caused the query to go through a different path from my vantage point to the destination, resulting in an increased time.

### 3 Conclusion

In these two experiments, we gained familiarity with various commands for *"monitoring network traffic"* on routers, as well as working with commands in Linux, Windows, and tools such as *perfmon*. We examined the *"DNS"* servers for several different domains and obtained basic information about these server domains. Furthermore, we observed the differences in command execution before and after caching. Overall, these experiments allowed us to gain a deeper understanding of Monitoring, DNS performance, and the importance of DNS server selection for optimizing network performance.