

Oliver Zhou

Bram Lewis

CS 372: Networks

28 January 2024

## Introduction to Network Programming

### Command used to run the program

```
python network_monitor.py
```

### Adding a ICMP service monitor for x.com

```
-----  
Enter any options in 'single quotes' to begin adding a service.  
----- Standard services -----  
  'HTTP','HTTPS','ICMP','DNS','NTP','TCP','UDP'  
----- Additional options -----  
'echo' - Test the custom TCP echo server.  
Return/enter - to start monitoring.  
'start' - also start monitoring.  
'view' - View all currently added services.  
'restart' - Clear added services and start over.  
'exit' - Exit the program.  
-----  
Enter option: icmp  
Enter server address for ICMP: x.com  
Enter ICMP option ('ping' or 'traceroute') (default = ping):  
Enter time the interval for checks: (default = 10):  
Service added.  
-----
```

*Below the prompt showing all the available options, 'icmp' was entered to choose the ICMP service for monitoring. Parameters such as the address and the ping tool and a 10 second interval was selected using default parameters.*

## Adding a DNS service with parameters configured by the user

```
-----
Enter option: dns
Enter the DNS server address: 8.8.8.8
Enter the DNS query lookup address: hotmail.com
Enter the DNS record type (A, MX, AAAA, CNAME, or manual) (default = A): MX
Enter the time interval for checks: (default = 10): 5
Service added.
-----
Enter any options in 'single quotes' to begin adding a service to monitor.
----- Standard services -----
'HTTP','HTTPS','ICMP','DNS','NTP','TCP','UDP'
----- Additional options -----
'echo' - Test the custom TCP echo server.
Return/enter - to start monitoring.
'start' - also start monitoring.
'view' - View all currently added services.
'restart' - Clear added services and start over.
'exit' - Exit the program.
-----
Enter option: view
Currently added services:
{'type': 'NTP', 'server': 'pool.ntp.org', 'interval': 5}
{'type': 'DNS', 'server': '8.8.8.8', 'query': 'hotmail.com', 'record_type': 'MX', 'interval': 5}
Enter option: █
```

*In this picture, the user configures the DNS monitor to query 8.8.8.8 (Google's DNS) for hotmail.com, with the MX record type for mail servers. Then an interval of monitoring every 5 seconds was configured.*

## Viewing added services using 'view'

```
-----
Enter any options in 'single quotes' to begin adding a service.
----- Standard services -----
'HTTP','HTTPS','ICMP','DNS','NTP','TCP','UDP'
----- Additional options -----
'echo' - Test the custom TCP echo server.
Return/enter - to start monitoring.
'start' - also start monitoring.
'view' - View all currently added services.
'restart' - Clear added services and start over.
'exit' - Exit the program.
-----
Enter option: view
Currently added services:
{'type': 'HTTP', 'url': 'http://www.google.com', 'interval': 10}
{'type': 'HTTPS', 'url': 'https://www.youtube.com', 'interval': 10}
{'type': 'ICMP', 'server': 'x.com', 'tool': 'ping', 'interval': 10}
{'type': 'DNS', 'server': '8.8.8.8', 'query': 'youtube.com', 'record_type': 'A', 'interval': 10}
{'type': 'NTP', 'server': 'pool.ntp.org', 'interval': 10}
{'type': 'TCP', 'server': 'google.com', 'port': 80, 'interval': 10}
{'type': 'UDP', 'server': '8.8.8.8', 'port': 53, 'timeout': 3, 'interval': 10}
Enter option: █
```

*The image shows a list of all the standard services that were previously added displayed in the terminal after typing 'view'. It contains all the configured parameters as well in the dictionary.*

### **Using 'restart' to clear the list**

```
-----
Enter option: http
Enter full URL (with http://) for HTTP service: http://www.google.com
Enter time the interval for checks: (default = 10): 5
Service added.
-----
Enter any options in 'single quotes' to begin adding a service.
----- Standard services -----
'HTTP','HTTPS','ICMP','DNS','NTP','TCP','UDP'
----- Additional options -----
'echo' - Test the custom TCP echo server.
Return/enter - to start monitoring.
'start' - also start monitoring.
'view' - View all currently added services.
'restart' - Clear added services and start over.
'exit' - Exit the program.
-----
Enter option: view
Currently added services:
{'type': 'HTTP', 'url': 'http://www.google.com', 'interval': 5}
Enter option: restart
Starting over.
Enter option: view
No services in the list.
Enter option: █
```

*With the view option, you can see an HTTP service in the list. After typing 'restart', the console says "Starting over." and when viewing again, there are no services found in the list.*

## Starting the monitor using 'start'

```
Enter option: start
-----
Starting monitor!
2024-01-28 16:12:07 - TCP Echo server status: Server is up! Echo reply: Hello!
2024-01-28 16:12:07 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:12:07 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:12:07 - 104.244.42.1 status: Server is up! - ping: 46.35 ms
2024-01-28 16:12:07 - HTTP URL: http://www.google.com. Server status: Server is up! Status Code: 200
2024-01-28 16:12:07 - NTP Server pool.ntp.org status: Server is up! Time: Sun Jan 28 16:12:07 2024
2024-01-28 16:12:07 - HTTPS URL: https://www.youtube.com. Server status: Server is up! Status Code: 200, Description: Server is
up
Type 'exit' to end monitoring: █
```

*To start the monitor, the option 'start' was entered. Alternatively, if the user returns, then it will also start.*

## Require one service

```
Enter option: view
No services in the list.
Enter option: start
Please include at least one service to monitor.
Enter option: █
```

*The program won't let you monitor nothing (otherwise what is the point?)*

# Results:

## Running the monitor for 4-5 intervals

```
2024-01-28 16:12:07 - Server: 8.8.8.8, UDP Port: 53, UDP Status: Server is up! Description: Port 53 on 8.8.8.8 is open or no response received.
2024-01-28 16:12:17 - TCP Echo server status: Server is up! Echo reply: Hello!
2024-01-28 16:12:17 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:12:17 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:12:17 - 104.244.42.1 status: Server is up! - ping: 50.50 ms
2024-01-28 16:12:17 - HTTP URL: http://www.google.com. Server status: Server is up! Status Code: 200
2024-01-28 16:12:17 - NTP Server pool.ntp.org status: Server is up! Time: Sun Jan 28 16:12:17 2024
2024-01-28 16:12:18 - HTTPS URL: https://www.youtube.com. Server status: Server is up! Status Code: 200, Description: Server is up
2024-01-28 16:12:20 - Server: 8.8.8.8, UDP Port: 53, UDP Status: Server is up! Description: Port 53 on 8.8.8.8 is open or no response received.
2024-01-28 16:12:27 - TCP Echo server status: Server is up! Echo reply: Hello!
2024-01-28 16:12:27 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:12:27 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:12:27 - 104.244.42.1 status: Server is up! - ping: 44.97 ms
2024-01-28 16:12:27 - HTTP URL: http://www.google.com. Server status: Server is up! Status Code: 200
2024-01-28 16:12:27 - NTP Server pool.ntp.org status: Server is up! Time: Sun Jan 28 16:12:27 2024
2024-01-28 16:12:28 - HTTPS URL: https://www.youtube.com. Server status: Server is up! Status Code: 200, Description: Server is up
2024-01-28 16:12:33 - Server: 8.8.8.8, UDP Port: 53, UDP Status: Server is up! Description: Port 53 on 8.8.8.8 is open or no response received.
2024-01-28 16:12:37 - TCP Echo server status: Server is up! Echo reply: Hello!
2024-01-28 16:12:37 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:12:37 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:12:37 - 104.244.42.1 status: Server is up! - ping: 45.55 ms
2024-01-28 16:12:37 - HTTP URL: http://www.google.com. Server status: Server is up! Status Code: 200
2024-01-28 16:12:37 - NTP Server pool.ntp.org status: Server is up! Time: Sun Jan 28 16:12:37 2024
2024-01-28 16:12:39 - HTTPS URL: https://www.youtube.com. Server status: Server is up! Status Code: 200, Description: Server is up
2024-01-28 16:12:47 - TCP Echo server status: Server is up! Echo reply: Hello!
2024-01-28 16:12:47 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:12:47 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:12:47 - 104.244.42.1 status: Server is up! - ping: 44.22 ms
2024-01-28 16:12:47 - HTTP URL: http://www.google.com. Server status: Server is up! Status Code: 200
2024-01-28 16:12:47 - NTP Server pool.ntp.org status: Server is up! Time: Sun Jan 28 16:12:47 2024
2024-01-28 16:12:46 - Server: 8.8.8.8, UDP Port: 53, UDP Status: Server is up! Description: Port 53 on 8.8.8.8 is open or no response received.
```

*With intervals of 10 seconds between each of these records, this is a set of 4 iterations from the service monitor. Below will be a more complex one with different intervals.*

# Results:

## Complex monitor with different intervals

```
Enter option: view
Currently added services:
{'type': 'ICMP', 'server': 'youtube.com', 'tool': 'ping', 'interval': 5}
{'type': 'DNS', 'server': '8.8.8.8', 'query': 'youtube.com', 'record_type': 'A', 'interval': 8}
{'type': 'TCP', 'server': 'google.com', 'port': 80, 'interval': 15}
{'type': 'NTP', 'server': 'pool.ntp.org', 'interval': 25}
Enter option:
-----
Starting monitor!
2024-01-28 16:28:35 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:28:35 - 142.251.215.238 status: Server is up! - ping: 22.65 ms
2024-01-28 16:28:35 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:28:35 - NTP Server pool.ntp.org status: Server is up! Time: Sun Jan 28 16:28:36 2024
2024-01-28 16:28:40 - 142.251.215.238 status: Server is up! - ping: 18.54 ms
2024-01-28 16:28:43 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:28:45 - 142.251.215.238 status: Server is up! - ping: 21.33 ms
2024-01-28 16:28:50 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:28:50 - 142.251.215.238 status: Server is up! - ping: 19.62 ms
2024-01-28 16:28:51 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.251.215.238']
2024-01-28 16:28:55 - 142.251.215.238 status: Server is up! - ping: 18.52 ms
2024-01-28 16:28:59 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:29:01 - 142.251.215.238 status: Server is up! - ping: 17.33 ms
2024-01-28 16:29:00 - NTP Server pool.ntp.org status: Server is up! Time: Sun Jan 28 16:29:01 2024
2024-01-28 16:29:05 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:29:06 - 142.251.215.238 status: Server is up! - ping: 18.84 ms
2024-01-28 16:29:08 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:29:11 - 142.251.215.238 status: Server is up! - ping: 19.85 ms
2024-01-28 16:29:16 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.251.215.238']
2024-01-28 16:29:16 - 142.251.215.238 status: Server is up! - ping: 16.20 ms
2024-01-28 16:29:20 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:29:21 - 142.251.215.238 status: Server is up! - ping: 17.04 ms
2024-01-28 16:29:24 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:29:26 - NTP Server pool.ntp.org status: Server is up! Time: Sun Jan 28 16:29:26 2024
2024-01-28 16:29:26 - 142.251.215.238 status: Server is up! - ping: 26.64 ms
2024-01-28 16:29:31 - 142.251.215.238 status: Server is up! - ping: 121.52 ms
2024-01-28 16:29:32 - DNS Server: 8.8.8.8, Status: Server is up! A Records Results: ['142.250.217.110']
2024-01-28 16:29:36 - Server: google.com, TCP Port: 80, TCP Status: Server is up! Description: Port 80 on google.com is open.
2024-01-28 16:29:36 - 142.251.215.238 status: Server is up! - ping: 27.13 ms
Type 'exit' to end monitoring: exit
```

*At the top of the output here, you can see the intervals set at the end of each service entry. This gives you an idea of the gaps between outputs in the monitor output below. For example, NTP has appropriately the longest gap between outputs because it has the highest interval.*



### After typing 'exit' to stop monitoring

```
2024-01-28 16:15:58 - Server: 8.8.8.8, UDP Port: 53, UDP s
sponse received.
Type 'exit' to end monitoring: exit
Please allow all threads to finish. Exiting...
Exiting.
```

*This is how exiting the monitor looks after typing 'exit' in the monitor command line. Depending on set interval times, this may take a while as all the threads handling each of the services being monitored need to have their stop events complete and threads finish before the program exits.*

### Adding the echo monitor

```
{'type': 'UDP', 'server': '8.8.8.8', 'port': 53, 'timeout': 3, 'interval': 10}
Enter option: echo
Enter the echo server address (recommendation: localhost/127.0.0.1): localhost
Enter port for echo service (default = 12345):
Enter message to echo (default = Hello!):
Enter the time interval for checks: (default = 10):
Service added.
-----
```

*This example shows the local echo server from the echo requirement being added.*

### Turning off the echo server while the monitor is running

```
-----
Starting monitor!
2024-01-28 16:14:40 - TCP Echo server status: Server is up! Echo reply: Test!
2024-01-28 16:14:50 - TCP Echo server status: Server is up! Echo reply: Test!
2024-01-28 16:15:00 - TCP Echo server status: Server is down! (or listening on wrong port).
Type 'exit' to end monitoring: █
```

*The server was turned off after the second monitor socket connection was made, and then when the next monitor check came and the server was off, it showed that the server was down because a connection could not be made.*