

Network Diagnostic and Tool Package Project

Course: CEN322 – Network Programming

Project 01

Deadline: 24.10.2025 – 23:59

1. Project Overview

This project implements a modular **Network Diagnostic and Tool Package** that provides hands-on experience with socket programming and network diagnostics in Python.

It integrates multiple independent modules—machine information retrieval, echo testing, SNTP synchronization, chat communication, and error management—into a single functional system.

The system has two main components:

- **Server** — Accepts incoming connections and routes requests to appropriate modules.
- **Client** — Connects to the server and provides a user interface to execute modules.

2. Project Structure

```
proje_1/
|
└── client/
    ├── chat_logs/
    ├── error_logs/
    ├── main.py
    └── modules/
        ├── chat_module.py
        ├── connection.py
        ├── echo_test.py
        ├── error_manager.py
        ├── machine_info.py
        ├── sntp_client.py
        ├── set_server_address.py
        └── __init__.py
|
└── server/
    ├── chat_logs/
    ├── main.py
    └── modules/
        ├── chat_module.py
        ├── echo_test.py
        └── __init__.py
```

3. Environment and Dependencies

Requirements:

- Python 3.10 or later
- Tested on macOS and Linux (Windows compatible)

Libraries Used:

- socket
- threading
- datetime
- ntplib
- subprocess, re, os

To install the required dependency:

```
pip install ntplib
```

4. Running the Project

Important:

Both **server** and **client** must be started from the **project root directory (proje_1/)**, not from inside the client/ or server/ folders.

Step 1: Start the Server

```
python server/main.py
```

- The server will start listening on **port 5050**.
- It must be running before starting the client.

Step 2: Start the Client

In a new terminal window (still in the proje_1/ directory):

```
python client/main.py
```

When prompted for the server IP address:

- If testing on the **same machine**, enter: 127.0.0.1

- (This IP will **not appear** in the local IP list, but it works without issues.)
- If testing over LAN, choose an IP from the listed devices.

5. Modules Overview

A. Machine Information Module

Retrieves and displays:

- Hostname
- IP address(es)
- Network interfaces

B. Echo Test Module

Tests client-server communication:

- Sends a message to the server
- Server echoes the same message
- Confirms successful connection if data matches

C. SNTP Time Synchronization Module

Synchronizes local time with a network time server (pool.ntp.org) and displays:

- Network time
- Local time
- Offset in seconds

D. Chat Module

Establishes a two-way chat session between client and server:

- Messages are logged in `chat_logs/`
- Use `exit` to terminate the chat

E. Error Management and Settings Module

Demonstrates:

- Timeout settings
- Socket configuration
- Error handling and recovery

6. Practical Notes and Recommendations

1. Single Machine Testing

Use 127.0.0.1 as the IP address.

Even though it does **not** appear in the network IP list, it is valid and works properly.

2. Server Requirements

- The server must be started **before** the client.
- Default port: **5050**
- Ensure the port is available and not blocked by another process.

3. Sequential Module Execution

Running multiple modules back-to-back may sometimes cause socket or threading errors depending on the environment.

If this occurs:

- Stop both the **server** and **client** completely.
- Restart them and try again.

4. Logs

- Chat logs are stored under chat_logs/.
- Errors (if any) are stored under error_logs/ on the client side.

7. Example Usage

Example 1 – Echo Test

Server Output:

[MAIN] Server running on 0.0.0.0:5050

[MAIN] Connected by ('127.0.0.1', 53422)

[ECHO] Client says: Hello

Client Output:

Connecting to 127.0.0.1:5050 ...

Connected to server.

Sent: Hello

Received: ECHO_RESPONSE:Hello

Connection successful, data matches.

Example 2 – SNTP Module

[SNTP] Connecting to pool.ntp.org ...

--- SNTP Time Synchronization ---

NTP Server : pool.ntp.org

Network Time : 2025-10-24 20:45:18

Local Time : 2025-10-24 20:45:16

Time Offset : 2.000 seconds

8. Troubleshooting

Problem	Possible Cause	Recommended Action
Connection refused	Server not running	Start the server first
Address already in use	Port 5050 busy	Wait or choose another port
Client stuck at “Connecting...”	Wrong IP or firewall	Use 127.0.0.1 for local testing
Errors after multiple module runs	Thread/socket reuse	Restart both server and client

9. Submission

Student: Halit Şen

Student no: 2021555060

Course: CEN322 – Network Programming