



# UNIVERSITY OF DHAKA

Department of Computer Science and Engineering

CSE-3111 : Computer Networking Lab

Project Report : NetTacToe (Networked Tic Tac Toe  
Showdown)

**Submitted By:**

Name : Md Rifat Khan

Roll No : SH-38

Name : Md Istahak Islam

Roll No : AK-16

**Submitted On :**

May 13, 2024

**Submitted To :**

Dr. Md. Abdur Razzaque

Dr. Md Mamunur Rashid

Dr. Md. Ibrahim

Mr. Md. Redwan Ahmed Rizvee

## Contents

<b>1</b>	<b>Objectives</b>	<b>2</b>
<b>2</b>	<b>Methodology</b>	<b>2</b>

# Title

## Net NetTacToe (Networked Tic Tac Toe Showdown)

### 1 Objectives

1. **Implementing Socket Communication:** Develop a robust client-server communication system using Python socket programming to enable data exchange between networked devices.
2. **Implementing Networking Algorithms:** Incorporate key networking algorithms such as DNS resolution, TCP flow control, and congestion control into the project to demonstrate understanding and application of fundamental networking principles.
3. **Ensuring Reliable Data Transfer:** Implement mechanisms within the project to ensure reliable and efficient data transfer over the network, considering factors such as error handling, data integrity, and flow control.
4. **Addressing Real-World Networking Challenges:** Address real-world networking challenges such as network latency, packet loss, and bandwidth constraints through the implementation of appropriate algorithms and techniques within the project.
5. **Creating User-Friendly Interfaces:** Design and implement a graphical user interface (GUI) using Tkinter to provide users with an intuitive platform for interacting with networking functionalities and algorithms.

### 2 Methodology

We used python socket programming and Tkinter library to develop our project. The clients can connect with the servers with the help of DNS servers. The servers pair up the clients with each other with it's implemented algorithm. While sending and recieving data we we implemented many flaws of the network such as packet drop, timeout, buffer overflow etc. We solved this using various algorihtm such TCP flow control, TCP congestion control.

## 1. System Design and Implementation

1. **Socket Communication:** We implemented client and server modules using Python socket programming for establishing connections and sending/receiving data between networking devices. The clients connect with each other through the use of an intermediary server to communicate with each other.
2. **GUI Development:** Design and create a user-friendly interface using Tkinter library of python to interact with the users.

## 2. Networking Algorithms Integration

1. **DNS Resolution:** Implement DNS resolution functionality to resolve domain names to IP addresses of the servers so that clients can connect with the appropriate servers.
2. **TCP Flow Control:** Implement TCP flow control mechanisms to regulate data flow between sender and receiver, ensuring efficient data transfer.
3. **Congestion Control:** Implement congestion control algorithms to manage network congestion and optimize data transmission rates.

## 3. Tools and Techniques

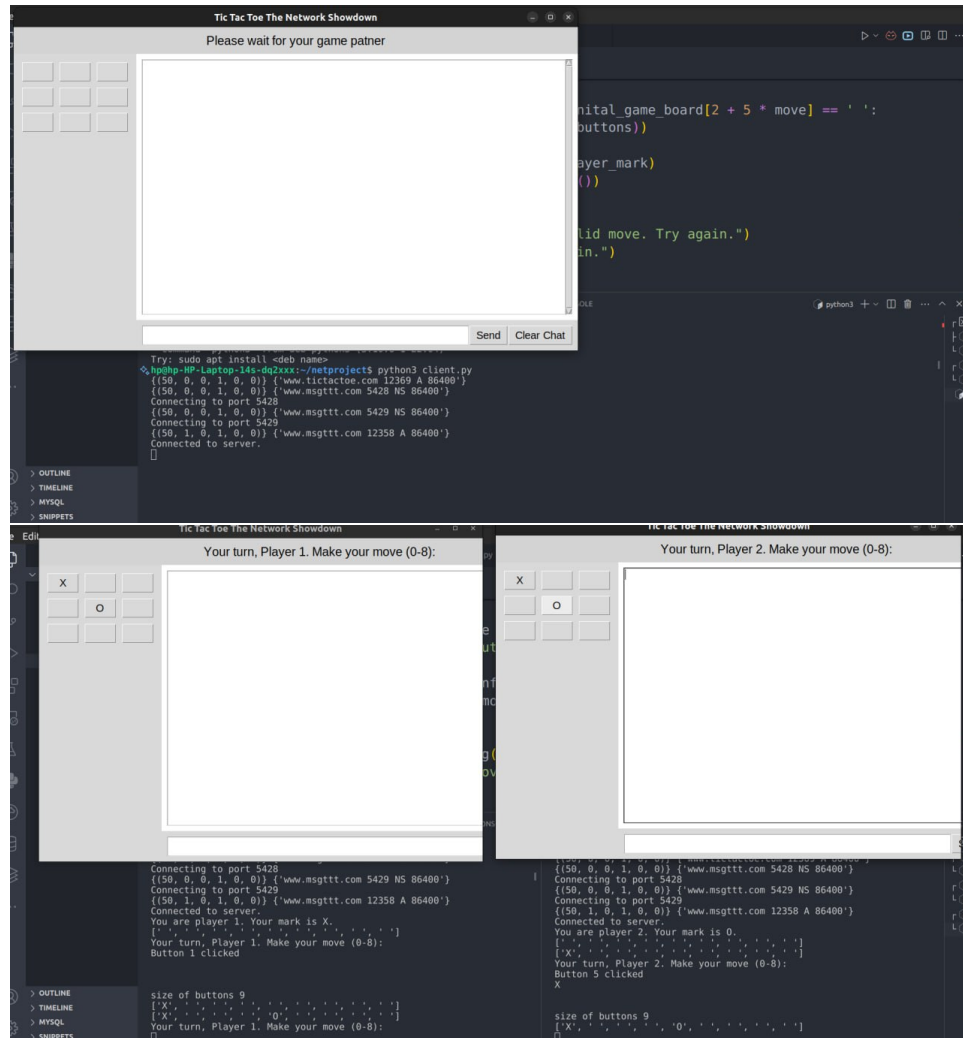
1. **Python:** Utilized for socket programming, algorithm implementations, and GUI development using Tkinter library.
2. **Tkinter:** Used to create a graphical user interface (GUI) for user interaction and visualization of networking functionalities.
3. **Networking Libraries:** Leveraged Python libraries such as socket and custom implementations for TCP flow control and congestion control algorithms.
4. **IDE:** Developed and tested code using an integrated development environment (IDE) VSCode for efficient coding and debugging.
5. **LaTeX:** Prepared project documentation and reports using LaTeX typesetting for professional and structured document formatting.

## 4. Outcomes

1. **Functional Networking System:** Developed a robust networking system using Python socket programming that facilitates communication between clients and servers. This system supports various networking tasks such as data transfer, DNS resolution, and network protocol implementations.
2. **Interactive GUI Design:** Designed and implemented a user-friendly graphical user interface (GUI) using Tkinter. The GUI provides intuitive controls for users to interact with networking functionalities and view real-time data related to DNS resolution, TCP flow control, and congestion control algorithms.
3. **Networking Algorithm Implementations:** Successfully integrated key networking algorithms into the system:
  - **DNS Resolution:** Implemented DNS resolution functionality to translate domain names to IP addresses, leveraging DNS protocols.
  - **TCP Flow Control:** Implemented mechanisms such as sliding window protocol to regulate data flow between communicating entities, ensuring efficient and reliable data transmission.
  - **Congestion Control:** Implemented congestion control algorithms, such as TCP Reno or similar, to manage network congestion and optimize data transfer rates.
4. **Reliable Data Transfer and Error Handling:** Implemented error handling mechanisms and data integrity checks to ensure reliable data transfer over the network. Techniques such as checksums and acknowledgment mechanisms were employed to detect and recover from transmission errors.
5. **Comprehensive Documentation:** Prepared detailed documentation covering system architecture, module functionalities, algorithm descriptions, testing methodologies, test results, performance analysis, challenges encountered, and recommendations for future enhancements.

## 5. Sample Input and Output

Here are some sample snapshots of our project.





## 6. Conclusion

In conclusion, this project has successfully achieved its objectives and demonstrated a comprehensive understanding and application of networking concepts, Python programming, and GUI development using Tkinter. Overall, this project has not only enhanced technical skills but also provided insights into real-world networking challenges and solutions. Future work may include expanding protocol support, optimizing algorithms for high-load scenarios, and incorporating advanced security features.