TOR CLI Routing Project Lab Solution

Prerequisites:

- A system running **Kali** or another Debian-based Linux distribution.
- **Root** access (sudo) to install and configure software.

1. Install Dependencies

```
sudo apt update
sudo apt install python3 python3-pip
```

This will install Python and the pip package manager for Python.

2. Set Up a Python Virtual Environment

Navigate to the project directory and set up a Python virtual environment for managing your Python packages:

```
cd ./TOR_CLI-Routing-Project/TOR_CLI-Routing-Presentation/
python -m venv venv
source venv/bin/activate
```

Once the environment is activated, install the required Python libraries:

```
pip install -r requirements.txt
```

This will install the dependencies listed in the `requirements.txt` file (it will install `PySocks`, `pycryptodome`, `stem`).

note!

in case you incounter issues with sourcesing of the venv you can always spesify the path instead
instead of python you would use ./venv/bin/python
instead of pip you would use ./venv/bin/pip

3. Install TOR and Start the Service

```
sudo apt install tor
```

Start the TOR service:

```
sudo systemctl start tor
```

Enable TOR to start automatically on boot:

```
sudo systemctl enable tor
```

Check the status of the TOR service to ensure it is running:

```
sudo systemctl status tor
```

4. Configure TOR for Control Access

Edit the TOR configuration file to enable the control port. Open the TOR config file:

```
sudo vim /etc/tor/torrc
```

Add or uncomment the following line to enable the **ControlPort**:

```
ControlPort 9051
```

```
diff torrc.back torrc
```

Restart the TOR service to apply the changes:

```
sudo systemctl restart tor.service
```

5. Verify TOR Setup

Use `curl` to verify that your system is properly routing traffic through the TOR network. Run the following command:

```
curl --socks5 127.0.0.1:9050 https://checkip.amazonaws.com
```

This should return the public IP address assigned by the TOR network, indicating that TOR is working correctly.

and you can compare it with your own pablic IP

```
curl https://checkip.amazonaws.com
```

diff file

```
curl https://checkip.amazonaws.com > your_ip

curl --socks5 127.0.0.1:9050 https://checkip.amazonaws.com > tor_ip

diff your_ip tor_ip > diff_of_ips.txt
```

6. Running the Client-Server Application

Run the Server

On one network (machine), run the **server.py** script. This server will be listening for incoming connections:

```
python server.py
```

Run the Client On another network (machine), run the **client.py** script. The client will connect to the server, using the TOR network to route the traffic:

note!

In case of lack of another mashine you can use services like AWS

```
python client.py
```

7. Install Nyx for TOR Monitoring

To monitor the TOR network and its circuits, install **Nyx**:

```
sudo apt install nyx
```

Launch **Nyx** to visualize the TOR circuit status:

```
nyx
```

Make sure that step `6.` was performed

client.py will show the Tor circuit, in Nyx, you can use the following commands to get additional information about the TOR network:

- To check the status of your TOR circuit:

  ```
  GETINFO circuit-status
  ```

- To view general information about your TOR instance:

  ```
  /info sendos
  ```

apply this commands to the circuit that was used by the client.py