Yoshiaki Ryuzaki A00859112
Sukh Atwal A00907714

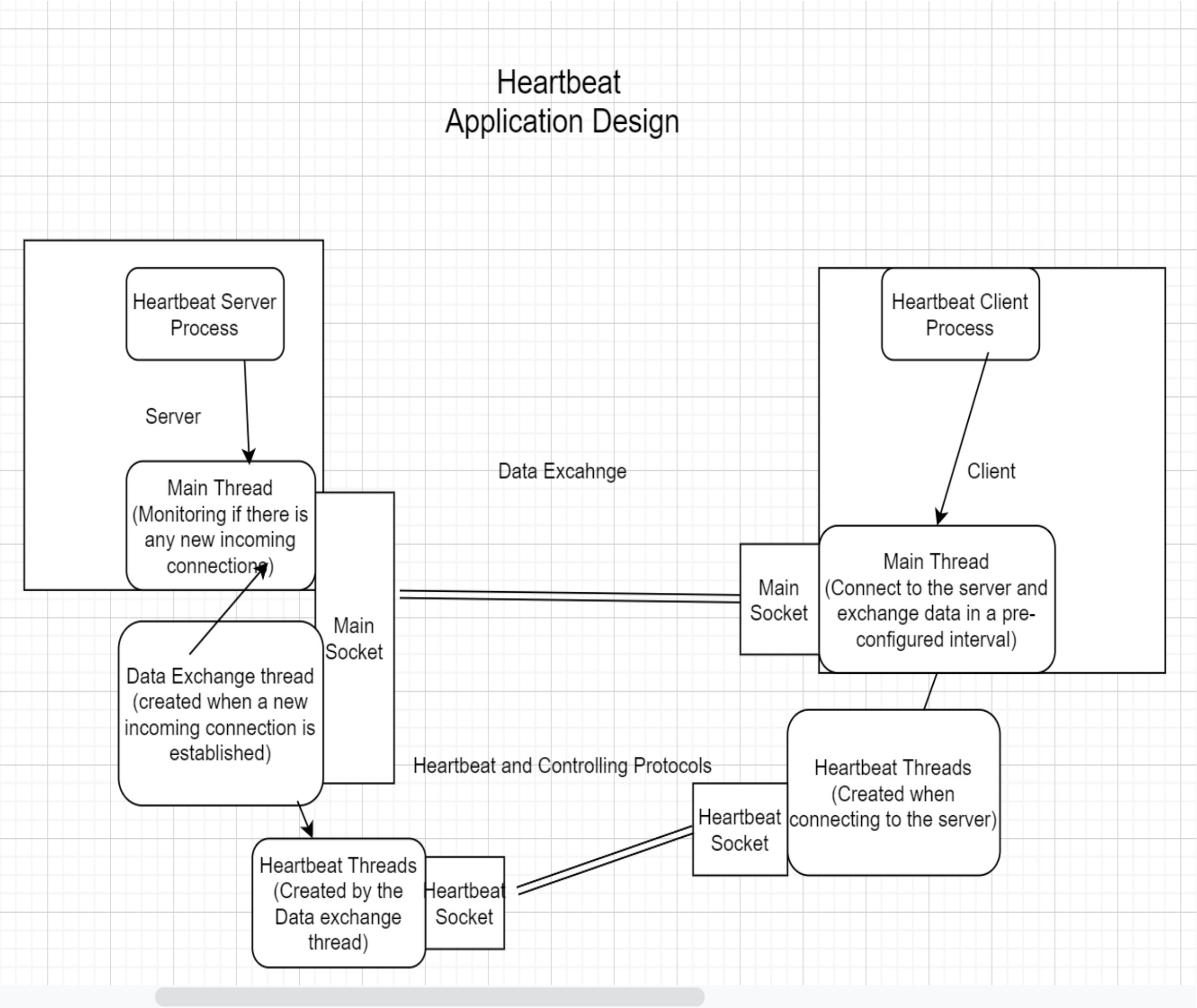
TCP Heartbeat Application Manual

Contents

Yoshiaki Ryuzaki A00859112
Sukh Atwal A00907714

Design

Heartbeat Application Design:


Heartbeat
Application Design
Heartbeat Server Process
Server
Main Thread (Monitoring if there is any new incoming connections)
Main Socket
Data Exchange thread (created when a new incoming connection is established)
Heartbeat Threads (Created by the Data exchange thread)
Heartbeat Socket
Data Excahnge
Heartbeat and Controlling Protocols
Heartbeat Client Process
Client
Main Socket
Main Thread (Connect to the server and exchange data in a pre-configured interval)
Heartbeat Socket
Heartbeat Threads (Created when connecting to the server)

Design description:

When a new client connects to the server, the main thread of the server starts a new Data Exchange thread to monitor the incoming traffic. The Data Exchange thread also starts a Heartbeat Thread which controls the Heartbeat protocol.

The Heartbeat Thread does nothing unless the Data Exchange thread switches on the Starthb switch.

The Starthb switch is only switched on if there has been incoming traffic for a long time. If client's Heartbeat client thread does not response the Heartbeat, the server will close the socket after sending an End Connection message to the client, the End Connection message will make the client close the client sockets and client process. If client does response to the Heartbeat protocol, the Starthb switch will be turned off. And the Data Exchange thread will still be monitoring the incoming data.

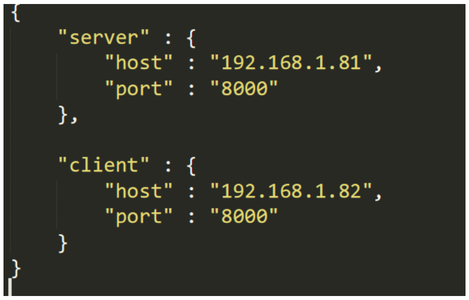
Yoshiaki Ryuzaki A00859112
Sukh Atwal A00907714

How to

The application includes 3 files:

1. **configuration.json:**

Configuration.json file is in Json format which includes all the configurations related to socket. Users can pre-configure the host IP address and port for both servers and clients. The configuration file should be copied to both the server and client under the same directory with server.py or client.py file.



```
{
    "server" : {
        "host" : "192.168.1.81",
        "port" : "8000"
    },

    "client" : {
        "host" : "192.168.1.82",
        "port" : "8000"
    }
}
```

2. **server.py:**

Server.py is a python script written in python 2. To execute it simply run ‘python server.py’. Once the script is executed, it prompts users to enter the following settings:

- Time interval in-between Heartbeats – Which is the time the Heartbeat protocol will wait before sending the next Heartbeat if the client does not reply
- Number of Heartbeats – Which is the number of Heartbeats the server will send before terminating the connection if the client does not reply
- Time to wait before starting Heartbeat protocol if no incoming data – Which is the number of seconds the Data Exchange thread will wait before turning on the Heartbeat protocol.
- setting cannot be lower than the total time of the Heartbeat protocol which is the Time interval in-between Heartbeats * Number of Heartbeats)

Users will have the option to enter new settings or keep using the default settings. After the user has entered all the settings. The server application will start running and monitoring the incoming traffic.

3. **client.py:**

The client.py is also a python script written in python2. 2. To execute it simply run ‘python client.py’. Once the script is executed, it prompts users to enter the following settings:

- Time interval in-between exchanging data – Which is how many seconds the client will wait before sending the next message to the server. (Heartbeat will be triggered if this number is set to too high)
- Time blocking the heartbeat protocol – This is the number of seconds before replying to the Heartbeat protocol. The default value is set to 0. Users should not change this value unless it is for testing purposes.

After the user entered all the settings. The client application will start running and exchanging data messages with the server.

Yoshiaki Ryuzaki A00859112
Sukh Atwal A00907714

Test Cases

All 4 cases listed are passed

Case Number	**Description**	**Expectation**
Case 1	**Normal data exchange at the configured rate**	**Heartbeat protocol should not kick in**
Case 2	**No or slow incoming data, but client does answer the heartbeat**	**Heartbeat protocol should kick in, but not terminate the connection**
Case 3	**No or slow incoming data, and client does not answer the heartbeat**	**Heartbeat protocol should kick in and terminate the connection.**
Case 4	**Client closes the connection**	**Server should close the connection without trigger Heartbeat protocol**

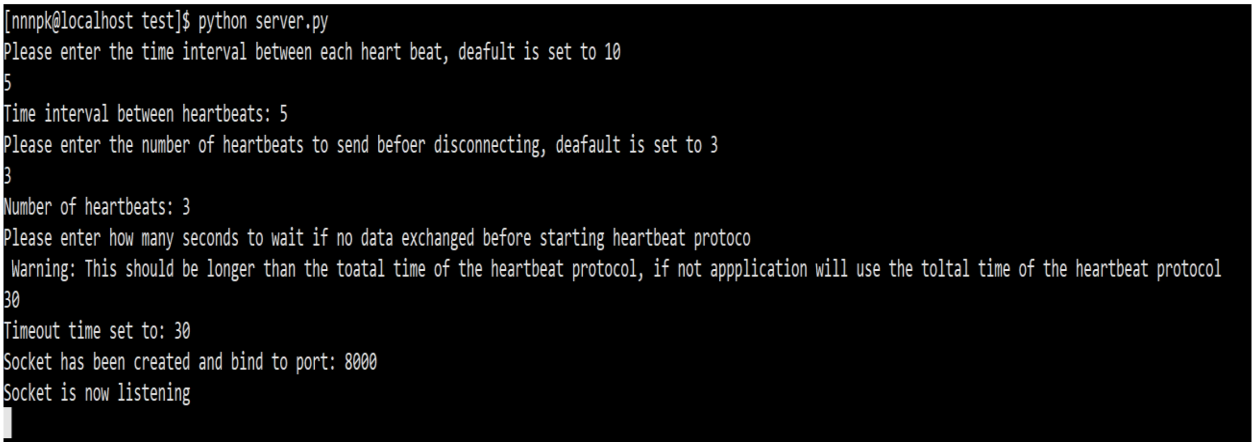
Test Case 1 – Normal data exchange

When data are being exchanged at the expected rate, Heartbeat should not be triggered.

These are the settings used for the test:

Server Setting:

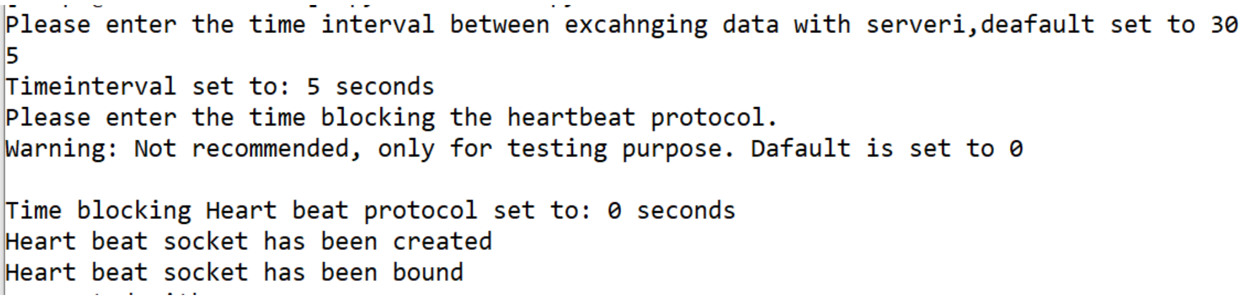
- Time Interval between Heartbeats – set to 5 seconds
- Number of Heartbeats – set to 3
- Expected maximum time interval before triggering heartbeat– set to 30 seconds



```
[nnnpk@localhost test]$ python server.py
Please enter the time interval between each heart beat, deafult is set to 10
5
Time interval between heartbeats: 5
Please enter the number of heartbeats to send befoer disconnecting, deafault is set to 3
3
Number of heartbeats: 3
Please enter how many seconds to wait if no data exchanged before starting heartbeat protoco
 Warning: This should be longer than the toatal time of the heartbeat protocol, if not appplication will use the toltal time of the heartbeat protocol
30
Timeout time set to: 30
Socket has been created and bind to port: 8000
Socket is now listening
```

Client Setting:

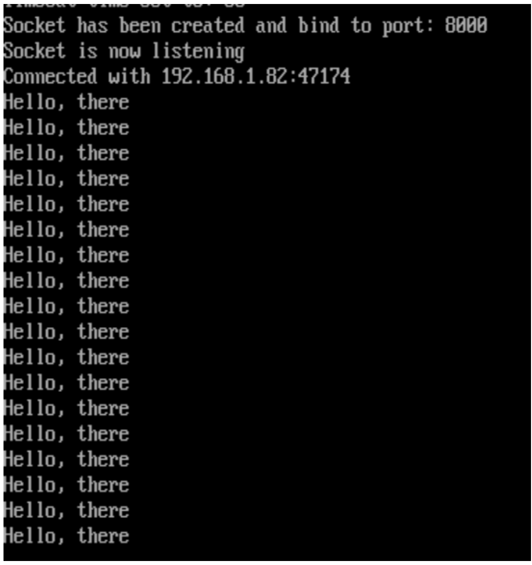
- Time interval between sending messages to the server – set to 5 seconds
- Time for blocking the heartbeat message (only used for testing purpose) – set to 0 seconds



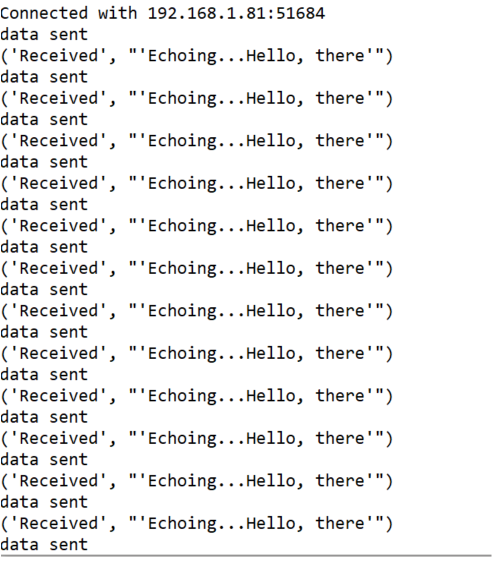
```
Please enter the time interval between excahnging data with serveri,deafault set to 30
5
Timeinterval set to: 5 seconds
Please enter the time blocking the heartbeat protocol.
Warning: Not recommended, only for testing purpose. Dafault is set to 0

Time blocking Heart beat protocol set to: 0 seconds
Heart beat socket has been created
Heart beat socket has been bound
```

Result of Test Case 1 - Passed



```
Socket has been created and bind to port: 8000
Socket is now listening
Connected with 192.168.1.82:47174
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
Hello, there
```



```
Connected with 192.168.1.81:51684
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
('Received', "'Echoing...Hello, there'")
data sent
```

Data were exchanged smoothly at the configured rate, and heartbeat protocol was not triggered. Which met the expectation we had set.

Case 2 – Slow or no incoming traffic, but client answers Heartbeat

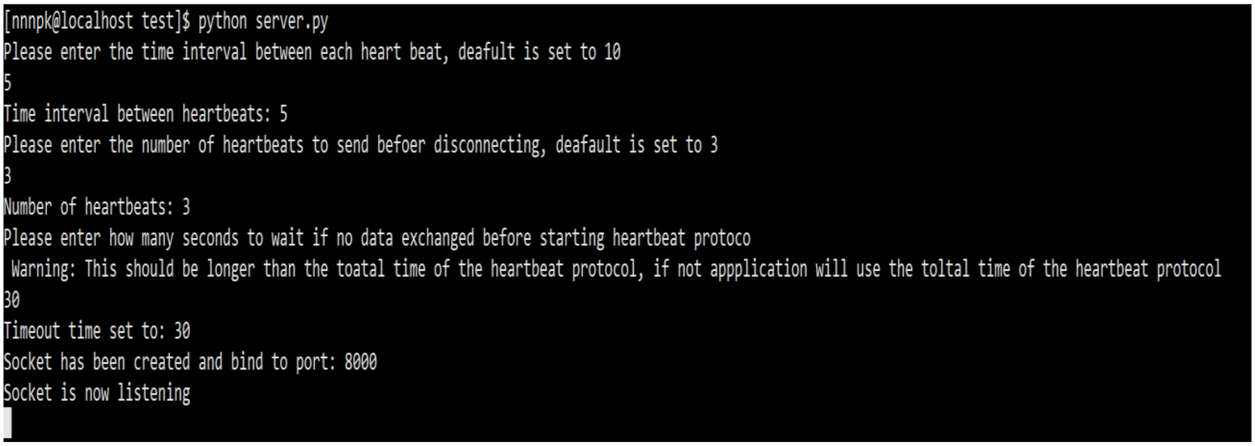
When there is no or slow incoming traffic, Heartbeat should be triggered. But as long as the client answers the Heartbeat, the connection will not be terminated.

Following are the settings used for the test:

Server Setting:

- Time Interval between Heartbeats – set to 5 seconds
- Number of Heartbeats – set to 3
- Expected maximum time interval before triggering heartbeat– set to 30 seconds

Yoshiaki Ryuzaki A00859112
Sukh Atwal A00907714



```
[nnnpk@localhost test]$ python server.py
Please enter the time interval between each heart beat, deafult is set to 10
5
Time interval between heartbeats: 5
Please enter the number of heartbeats to send befoer disconnecting, deafault is set to 3
3
Number of heartbeats: 3
Please enter how many seconds to wait if no data exchanged before starting heartbeat protoco
 Warning: This should be longer than the toatal time of the heartbeat protocol, if not appplication will use the toltal time of the heartbeat protocol
30
Timeout time set to: 30
Socket has been created and bind to port: 8000
Socket is now listening
```

Client Setting:

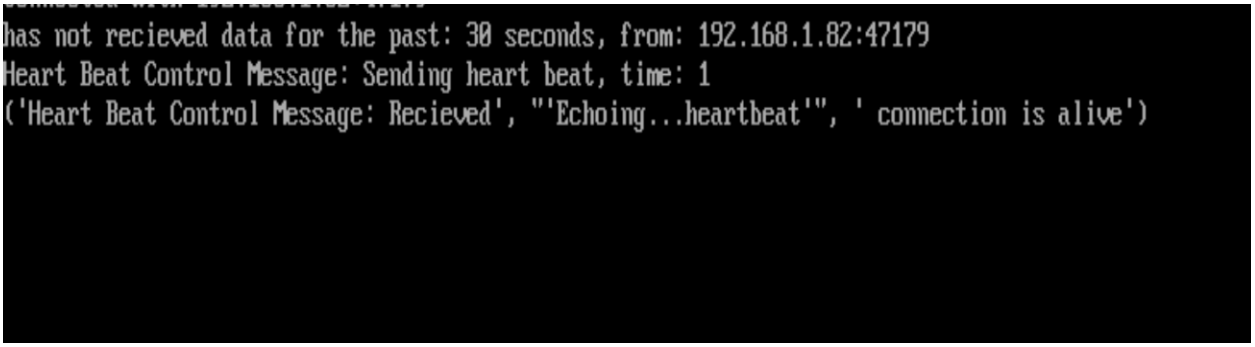
- Time interval between sending messages to the server – set to 50 seconds
- Time for blocking the heartbeat message (only used for testing purpose) – set to 0 seconds

```
[nnnpk@localhost beta]$ python client.py
Please enter the time interval between excahnging data with serveri,deafault set to 30
50
Timeinterval set to: 50 seconds
Please enter the time blocking the heartbeat protocol.
Warning: Not recommended, only for testing purpose. Dafault is set to 0

Time blocking Heart beat protocol set to: 0 seconds
Heart beat socket has been created
Heart beat socket has been bound
Connected with 192.168.1.81:51688
```

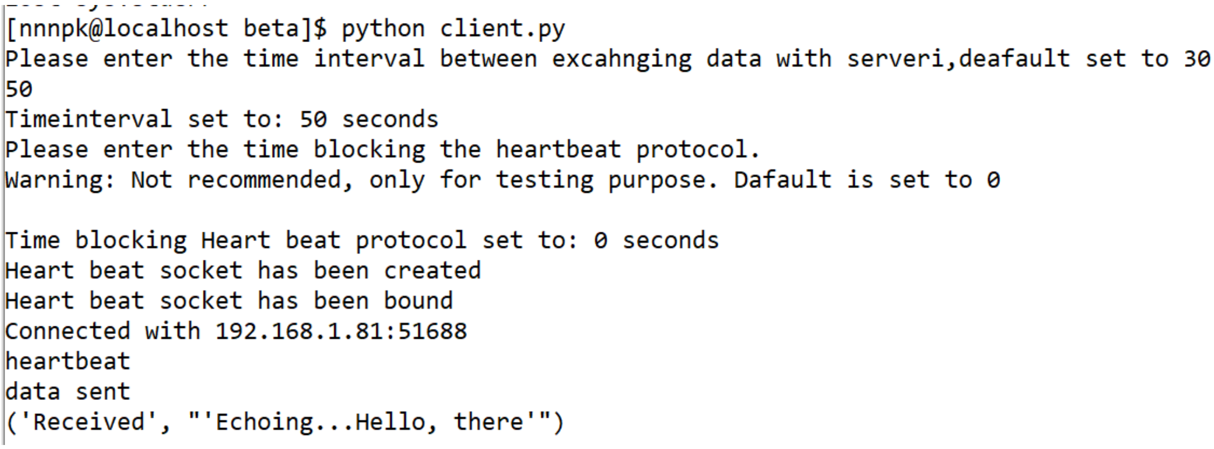
Result of Test Case 2 – Passed

Server Console:



```
Connected with 192.168.1.82:47179
has not recieved data for the past: 30 seconds, from: 192.168.1.82:47179
Heart Beat Control Message: Sending heart beat, time: 1
('Heart Beat Control Message: Recieved', "'Echoing...heartbeat'", ' connection is alive')
```

Client Console:



```
[nnnpk@localhost beta]$ python client.py
Please enter the time interval between excahnging data with serveri,deafault set to 30
50
Timeinterval set to: 50 seconds
Please enter the time blocking the heartbeat protocol.
Warning: Not recommended, only for testing purpose. Dafault is set to 0

Time blocking Heart beat protocol set to: 0 seconds
Heart beat socket has been created
Heart beat socket has been bound
Connected with 192.168.1.81:51688
heartbeat
data sent
('Received', "'Echoing...Hello, there'")
```

Heartbeat kicked in, but it confirmed the connection was still alive and did not terminate the connection. Which meets the expectation.

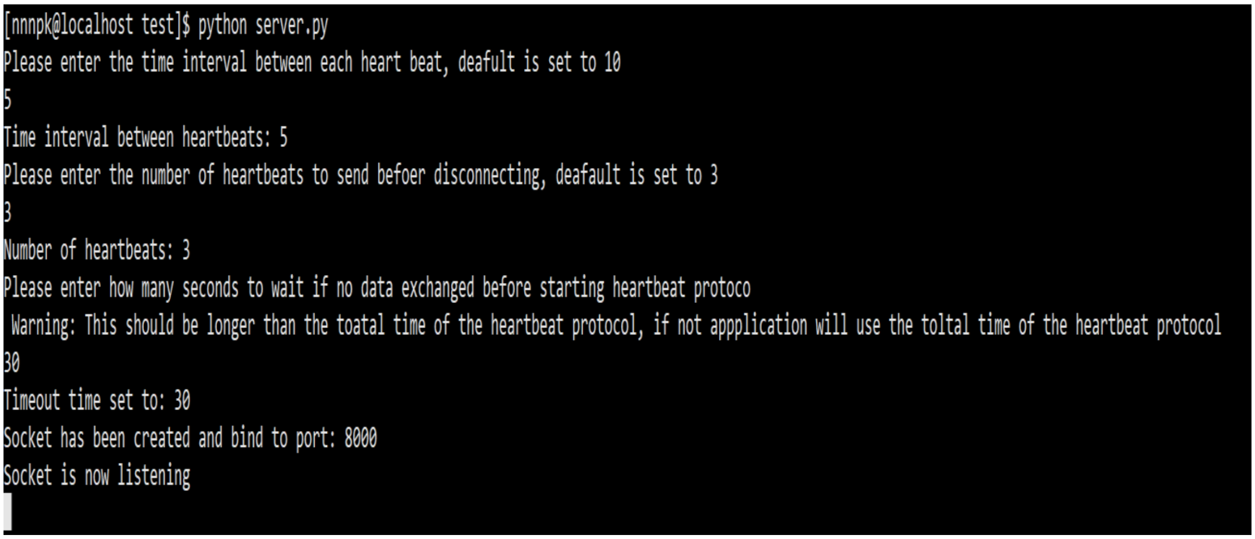
Test Case 3 – Slow or no incoming traffic, Heartbeat is not being answered.

When there is no or slow incoming traffic, Heartbeat should be triggered. And if client does not answer the Heartbeat, connection will be terminated.

Following are the settings used for the test:

Server Setting:

- Time Interval between Heartbeats – set to 5 seconds
- Number of Heartbeats – set to 3
- Expected maximum time interval before triggering heartbeat– set to 30 seconds



```
[nnnpk@localhost test]$ python server.py
Please enter the time interval between each heart beat, deafult is set to 10
5
Time interval between heartbeats: 5
Please enter the number of heartbeats to send befoer disconnecting, deafault is set to 3
3
Number of heartbeats: 3
Please enter how many seconds to wait if no data exchanged before starting heartbeat protoco
 Warning: This should be longer than the toatal time of the heartbeat protocol, if not appplication will use the toltal time of the heartbeat protocol
30
Timeout time set to: 30
Socket has been created and bind to port: 8000
Socket is now listening
```

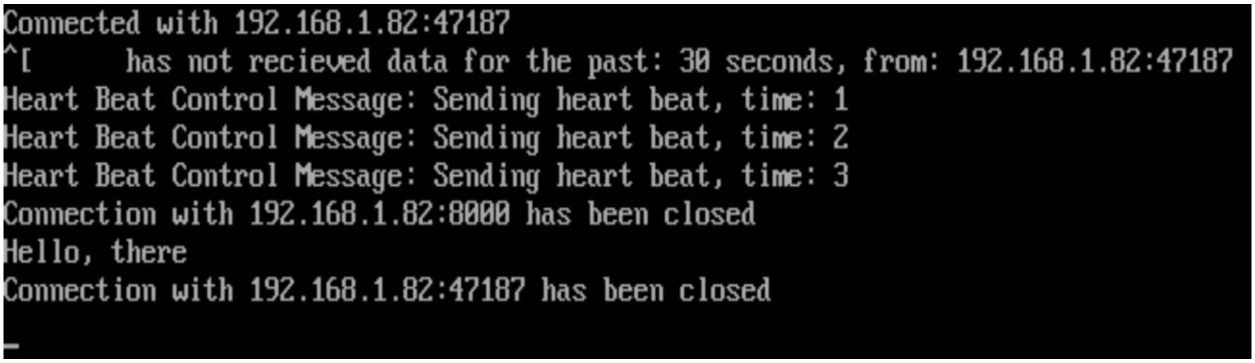
Client Setting:

Yoshiaki Ryuzaki A00859112
Sukh Atwal A00907714

- Time interval between sending messages to the server – set to 50 seconds
- Time for blocking the heartbeat message (only used for testing purpose) – set to 180 seconds (Which blocks the client from reply the Heartbeat)

```
[nnnpk@localhost beta]$ python client.py
Please enter the time interval between excahnging data with serveri,deafault set to 30
50
Timeinterval set to: 50 seconds
Please enter the time blocking the heartbeat protocol.
Warning: Not recommended, only for testing purpose. Dafault is set to 0
30
Time blocking Heart beat protocol set to: 30 seconds
Heart beat socket has been created
Heart beat socket has been bound
Connected with 192.168.1.81:51710
```

Result of Test Case 3 – passed



```
Connected with 192.168.1.82:47187
^[      has not recieved data for the past: 30 seconds, from: 192.168.1.82:47187
Heart Beat Control Message: Sending heart beat, time: 1
Heart Beat Control Message: Sending heart beat, time: 2
Heart Beat Control Message: Sending heart beat, time: 3
Connection with 192.168.1.82:8000 has been closed
Hello, there
Connection with 192.168.1.82:47187 has been closed
```

```
[nnnpk@localhost beta]$ python client.py
Please enter the time interval between excahnging data with serveri,deafault set to 30
50
Timeinterval set to: 50 seconds
Please enter the time blocking the heartbeat protocol.
Warning: Not recommended, only for testing purpose. Dafault is set to 0
30
Time blocking Heart beat protocol set to: 30 seconds
Heart beat socket has been created
Heart beat socket has been bound
Connected with 192.168.1.81:51710
heartbeat
data sent
('Received', "'Echoing...Hello, thereEnd Connection'")
Connection closed by server
[nnnpk@localhost beta]$
```

Heartbeat was triggered, and it terminated the connection since the client did not reply. Which meets the expectation.

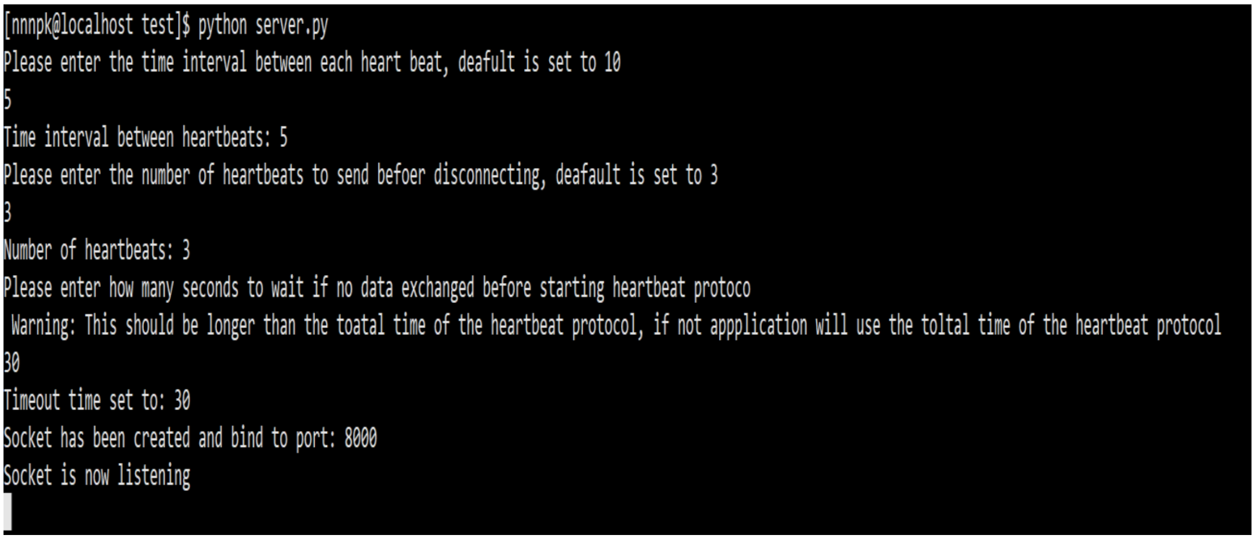
Yoshiaki Ryuzaki A00859112
Sukh Atwal A00907714

Test Case 4 – Client closes the connection

When the client terminates the connection, serer should close the socket without starting the Heartbeat.

Server Setting:

- Time Interval between Heartbeats – set to 5 seconds
- Number of Heartbeats – set to 3
- Expected maximum time interval before triggering heartbeat– set to 30 seconds



```
[nnnpk@localhost test]$ python server.py
Please enter the time interval between each heart beat, deafult is set to 10
5
Time interval between heartbeats: 5
Please enter the number of heartbeats to send befoer disconnecting, deafault is set to 3
3
Number of heartbeats: 3
Please enter how many seconds to wait if no data exchanged before starting heartbeat protoco
 Warning: This should be longer than the toatal time of the heartbeat protocol, if not appplication will use the toltal time of the heartbeat protocol
30
Timeout time set to: 30
Socket has been created and bind to port: 8000
Socket is now listening
```

Client Setting:

- Time interval between sending messages to the server – set to 5 seconds

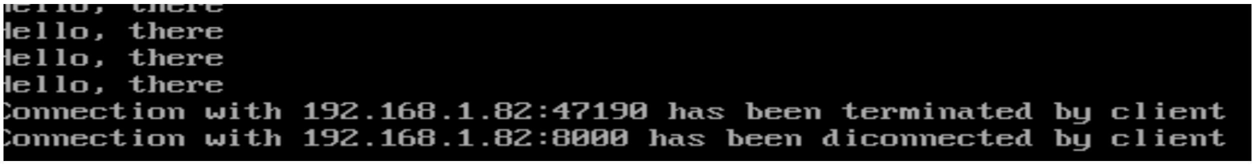
Time for blocking the heartbeat message (only used for testing purpose) – set to 0 seconds

```
[nnnpk@localhost beta]$ python client.py
Please enter the time interval between excahnging data with serveri,deafault set to 30
5
Timeinterval set to: 5 seconds
Please enter the time blocking the heartbeat protocol.
Warning: Not recommended, only for testing purpose. Dafault is set to 0

Time blocking Heart beat protocol set to: 0 seconds
Heart beat socket has been created
Heart beat socket has been bound
Connected with 192.168.1.81:51750
```

Yoshiaki Ryuzaki A00859112
Sukh Atwal A00907714

Result of test case 4 - Passed



```
Hello, there
Hello, there
Hello, there
Hello, there
Connection with 192.168.1.82:47190 has been terminated by client
Connection with 192.168.1.82:8000 has been diconnected by client
```

When client terminated the connection, server closed the sockets without triggering the Heartbeat. Which meets the expectation.

Source Code:

For source code, please check the attached source code files.