21BDS0340

Abhinav Dinesh Srivatsa

Computer Networks Lab

Assignment – IV

Question 1

Aim:

Implement the Stop and Wait protocol without timer

Output:

StopAndWaitServer.java

```
(base) abhi@Abhinavs-MacBook-Pro Assignment 4 % java StopAndWaitServer
Waiting for connection...
Received acknowledgement.

Sending packet 1... Received acknowledgement.

Ending connection.
```

StopAndWaitClient.java

```
(base) abhi@Abhinavs-MacBook-Pro Assignment 4 % java StopAndWaitClient
Initiating connection...
Sending acknowledgement.

Received packet... Sending acknowledgement.
Received packet... Destroyed connection.

Message: Hi my name is Abhinav Dinesh Srivatsa.
My registeration number is: 21BDS0340.
```

Code:

StopAndWaitServer.java

```java
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Arrays;

public class StopAndWaitServer {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public StopAndWaitServer(int port) throws IOException {
        this.port = port;
        this.socket = new DatagramSocket(port);
    }
```

```
    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsInt[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsInt[x] = (char) Integer.parseInt(elementsString[x]);

        return elementsInt;
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        System.out.println("Received acknowledgement.");

        this.address = packet.getAddress();
        this.port = packet.getPort();
        return stringToCharArray(Arrays.toString(data));
    }

    private void sendPacket(String dataString) throws IOException {
        byte dataByte[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(dataByte, dataByte.length,
                this.address, this.port);
        socket.send(packet);
    }

    private boolean checkAck(int expectedPacketNumber, char data[]) {
        return ((char) expectedPacketNumber) == (data[0] - '0');
    }

    private void sendData(String dataString) throws IOException {
        int packetNumber = 1;

        for (int x = 0; x <= dataString.length() / 1023; x++) {
            String data = Integer.toString(packetNumber);
            if (x + PACKET_SIZE - 1 < dataString.length() - 1)
                data += dataString.substring(x, x + PACKET_SIZE - 1);
            else
                data += dataString.substring(x, dataString.length());

            boolean ack = false;
            while (!ack) {
                System.out.print("Sending packet " + packetNumber + "... ");
```

```
                sendPacket(data);
                char response[] = receivePacket();
                ack = checkAck(packetNumber, response);
            }

            packetNumber++;
        }

        sendPacket("");
    }

    public static void main(String[] args) throws IOException {
        int port = 5000;
        String data = "Hi my name is Abhinav Dinesh Srivatsa.\nMy registeration
number is: 21BDS0340.";
        StopAndWaitServer saws = new StopAndWaitServer(port);
        System.out.println("Waiting for connection... ");
        saws.receivePacket();
        System.out.println();

        saws.sendData(data);
        System.out.println("\nEnding connection.");
    }
}
```

StopAndWaitClient.java

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Arrays;

public class StopAndWaitClient {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public StopAndWaitClient(InetAddress address, int port) throws SocketException
{
        this.port = port;
        this.address = address;
        socket = new DatagramSocket();
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
```

```
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsChar[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsChar[x] = (char) Integer.parseInt(elementsString[x]);

        return elementsChar;
    }

    public void sendPacket(String dataString) throws IOException {
        byte data[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(data, data.length, address,
port);

        System.out.println("Sending acknowledgement.");
        socket.send(packet);
    }

    public char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        return stringToCharArray(Arrays.toString(data));
    }

    public String[] receiveTransmission() throws IOException {
        String messages[] = new String[10];

        for (int x = 1; x < messages.length; x++) {
            char data[] = receivePacket();
            System.out.print("Received packet... ");

            if (data[0] == '\0')
                break;

            messages[x] = new String(data);
            String index = messages[x].substring(0, 1);
            messages[x] = messages[x].substring(1, messages[x].length() - 1);

            sendPacket(index);
        }

        System.out.println("Destroyed connection.\n");

        return messages;
    }
```

```
    public static void main(String[] args) throws IOException {
        InetAddress address = InetAddress.getByName("localhost");
        int port = 5000;
        StopAndWaitClient sawc = new StopAndWaitClient(address, port);
        System.out.println("Initiating connection...");
        sawc.sendPacket("initiate");
        System.out.println();

        String messages[] = sawc.receiveTransmission();
        for (int x = 1; x < messages.length; x++)
            if (messages[x] != null)
                System.out.println("Message: " + messages[x]);
    }
}
```

Question 2

Aim:

Implement the Stop and Wait protocol with Time ARQ (Automatic Repeat Query/ Request)

Output:

StopAndWaitARQServer.java

```
(base) abhi@Abhinavs-MacBook-Pro Assignment 4 % java StopAndWaitARQServer
Waiting for connection...
Acknowledgement not received.
Acknowledgement not received.
Acknowledgement not received.
Acknowledgement not received.
Acknowledgement not received.
Acknowledgement not received.
Acknowledgement not received.
Received acknowledgement.

Sending packet 1... Received acknowledgement.

Ending connection.
```

StopAndWaitClient.java

```
(base) abhi@Abhinavs-MacBook-Pro Assignment 4 % java StopAndWaitClient
Initiating connection...
Sending acknowledgement.

Received packet... Sending acknowledgement.
Received packet... Destroyed connection.

Message: Hi my name is Abhinav Dinesh Srivatsa.
My registeration number is: 21BDS0340.
```

Code:

StopAndWaitARQServer.java

```java
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Arrays;

public class StopAndWaitARQServer {
    private static final int PACKET_SIZE = 1024, TIMEOUT = 3000;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public StopAndWaitARQServer(int port) throws IOException {
        this.port = port;
        this.socket = new DatagramSocket(port);
        this.socket.setSoTimeout(TIMEOUT);
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
```

```
        String elementsString[] = middle.split(", ");
        char elementsChar[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsChar[x] = (char) Integer.parseInt(elementsString[x]);

        return elementsChar;
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] receivePacket() {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        while (stringToCharArray(Arrays.toString(data))[0] == '\0')
            try {
                socket.receive(packet);
            } catch (IOException e) {
                System.out.println("Acknowledgement not received.");
            }
        System.out.println("Received acknowledgement.");

        this.address = packet.getAddress();
        this.port = packet.getPort();
        return stringToCharArray(Arrays.toString(data));
    }

    private void sendPacket(String dataString) throws IOException {
        byte dataByte[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(dataByte, dataByte.length,
                this.address, this.port);
        socket.send(packet);
    }

    private boolean checkAck(int expectedPacketNumber, char data[]) {
        return ((char) expectedPacketNumber) == (data[0] - '0');
    }

    private void sendData(String dataString) throws IOException {
        int packetNumber = 1;

        for (int x = 0; x <= dataString.length() / 1023; x++) {
            String data = Integer.toString(packetNumber);
            if (x + PACKET_SIZE - 1 < dataString.length() - 1)
                data += dataString.substring(x, x + PACKET_SIZE - 1);
            else
                data += dataString.substring(x, dataString.length());
```

```
            boolean ack = false;
            while (!ack) {
                System.out.print("Sending packet " + packetNumber + "... ");
                sendPacket(data);
                char response[] = {};
                response = receivePacket();
                ack = checkAck(packetNumber, response);
            }

            packetNumber++;
        }

        sendPacket("");
    }

    public static void main(String[] args) throws IOException {
        int port = 5000;
        String data = "Hi my name is Abhinav Dinesh Srivatsa.\nMy registeration
number is: 21BDS0340.";
        StopAndWaitARQServer saws = new StopAndWaitARQServer(port);
        System.out.println("Waiting for connection... ");
        saws.receivePacket();
        System.out.println();

        saws.sendData(data);
        System.out.println("\nEnding connection.");
    }
}
```

StopAndWaitClient.java

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Arrays;

public class StopAndWaitClient {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public StopAndWaitClient(InetAddress address, int port) throws SocketException
{
        this.port = port;
        this.address = address;
        socket = new DatagramSocket();
    }
```

```
    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsChar[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsChar[x] = (char) Integer.parseInt(elementsString[x]);

        return elementsChar;
    }

    public void sendPacket(String dataString) throws IOException {
        byte data[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(data, data.length, address,
port);

        System.out.println("Sending acknowledgement.");
        socket.send(packet);
    }

    public char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        return stringToCharArray(Arrays.toString(data));
    }

    public String[] receiveTransmission() throws IOException {
        String messages[] = new String[10];

        for (int x = 1; x < messages.length; x++) {
            char data[] = receivePacket();
            System.out.print("Received packet... ");

            if (data[0] == '\0')
                break;

            messages[x] = new String(data);
            String index = messages[x].substring(0, 1);
            messages[x] = messages[x].substring(1, messages[x].length() - 1);

            sendPacket(index);
        }

        System.out.println("Destroyed connection.\n");
```

```
        return messages;
    }

    public static void main(String[] args) throws IOException {
        InetAddress address = InetAddress.getByName("localhost");
        int port = 5000;
        StopAndWaitClient sawc = new StopAndWaitClient(address, port);
        System.out.println("Initiating connection...");
        sawc.sendPacket("initiate");
        System.out.println();

        String messages[] = sawc.receiveTransmission();
        for (int x = 1; x < messages.length; x++)
            if (messages[x] != null)
                System.out.println("Message: " + messages[x]);
    }
}
```

Question 3

Aim:

Implement the Sliding window protocol (General way)

Output:

SlidingWindowGeneralServer.java

```
(base) abhi@Abhinavs-MacBook-Pro Assignment 4 % java SlidingWindowGeneralServer
Waiting for connection...
Received acknowledgement.

Sending packet 1...
Ending connection.
```

SlidingWindowGeneralClient.java

```
(base) abhi@Abhinavs-MacBook-Pro Assignment 4 % java SlidingWindowGeneralClient
Initiating connection...
Sending acknowledgement.

Received packet... Sending acknowledgement.
Received packet... Destroyed connection.

Message: Hi my name is Abhinav Dinesh Srivatsa.
My registeration number is: 21BDS0340.
```

Code:

SlidingWindowGeneralServer.java

```java
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Arrays;

public class SlidingWindowGeneralServer {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public SlidingWindowGeneralServer(int port) throws IOException {
        this.port = port;
        this.socket = new DatagramSocket(port);
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsInt[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsInt[x] = (char) Integer.parseInt(elementsString[x]);
```

```
        return elementsInt;
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        System.out.println("Received acknowledgement.");

        this.address = packet.getAddress();
        this.port = packet.getPort();
        return stringToCharArray(Arrays.toString(data));
    }

    private void sendPacket(String dataString) throws IOException {
        byte dataByte[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(dataByte, dataByte.length,
                this.address, this.port);
        socket.send(packet);
    }

    private void sendData(String dataString) throws IOException {
        int packetNumber = 1;

        for (int x = 0; x <= dataString.length() / 1023; x++) {
            String data = Integer.toString(packetNumber);
            if (x + PACKET_SIZE - 1 < dataString.length() - 1)
                data += dataString.substring(x, x + PACKET_SIZE - 1);
            else
                data += dataString.substring(x, dataString.length());

            System.out.print("Sending packet " + packetNumber + "... ");
            sendPacket(data);

            packetNumber++;
        }

        sendPacket("");
    }

    public static void main(String[] args) throws IOException {
        int port = 5000;
        String data = "Hi my name is Abhinav Dinesh Srivatsa.\nMy registeration
number is: 21BDS0340.";
        SlidingWindowGeneralServer swgs = new SlidingWindowGeneralServer(port);
        System.out.println("Waiting for connection... ");
```

```
        swgs.receivePacket();
        System.out.println();

        swgs.sendData(data);
        System.out.println("\nEnding connection.");
    }
}
```

SlidingWindowGeneralClient.java

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Arrays;

public class SlidingWindowGeneralClient {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public SlidingWindowGeneralClient(InetAddress address, int port) throws
SocketException {
        this.port = port;
        this.address = address;
        socket = new DatagramSocket();
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsChar[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsChar[x] = (char) Integer.parseInt(elementsString[x]);

        return elementsChar;
    }

    public void sendPacket(String dataString) throws IOException {
        byte data[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(data, data.length, address,
port);
```

```
        System.out.println("Sending acknowledgement.");
        socket.send(packet);
    }

    public char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        return stringToCharArray(Arrays.toString(data));
    }

    public String[] receiveTransmission() throws IOException {
        String messages[] = new String[10];

        for (int x = 1; x < messages.length; x++) {
            char data[] = receivePacket();
            System.out.print("Received packet... ");

            if (data[0] == '\0')
                break;

            messages[x] = new String(data);
            String index = messages[x].substring(0, 1);
            messages[x] = messages[x].substring(1, messages[x].length() - 1);

            sendPacket(index);
        }

        System.out.println("Destroyed connection.\n");

        return messages;
    }

    public static void main(String[] args) throws IOException {
        InetAddress address = InetAddress.getByName("localhost");
        int port = 5000;
        SlidingWindowGeneralClient swgc = new SlidingWindowGeneralClient(address,
port);
        System.out.println("Initiating connection...");
        swgc.sendPacket("initiate");
        System.out.println();

        String messages[] = swgc.receiveTransmission();
        for (int x = 1; x < messages.length; x++)
            if (messages[x] != null)
                System.out.println("Message: " + messages[x]);
    }
}
```

Question 4

Aim:

Implement the go back N protocol

Output:

SlidingWindowGoBackNServer.java

```
(base) abhi@Abhinavs-MacBook-Pro Assignment 4 % java SlidingWindowGeneralServer
Waiting for connection...
Received acknowledgement.

Sending packet 1...
Ending connection.
```

SlidingWindowGoBackNClient.java

```
(base) abhi@Abhinavs-MacBook-Pro Assignment 4 % java SlidingWindowGeneralClient
Initiating connection...
Sending acknowledgement.

Received packet... Sending acknowledgement.
Received packet... Destroyed connection.

Message: Hi my name is Abhinav Dinesh Srivatsa.
My registeration number is: 21BDS0340.
```

Code:

SlidingWindowGoBackNServer.java

```java
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Arrays;

public class SlidingWindowGoBackNServer {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public SlidingWindowGoBackNServer(int port) throws IOException {
        this.port = port;
        this.socket = new DatagramSocket(port);
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsInt[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsInt[x] = (char) Integer.parseInt(elementsString[x]);
```

```
        return elementsInt;
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        System.out.println("Received acknowledgement.");

        this.address = packet.getAddress();
        this.port = packet.getPort();
        return stringToCharArray(Arrays.toString(data));
    }

    private void sendPacket(String dataString) throws IOException {
        byte dataByte[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(dataByte, dataByte.length,
                this.address, this.port);
        socket.send(packet);
    }

    private void sendData(String dataString) throws IOException {
        int packetNumber = 1;

        for (int x = 0; x <= dataString.length() / 1023; x++) {
            String data = Integer.toString(packetNumber);
            if (x + PACKET_SIZE - 1 < dataString.length() - 1)
                data += dataString.substring(x, x + PACKET_SIZE - 1);
            else
                data += dataString.substring(x, dataString.length());

            System.out.print("Sending packet " + packetNumber + "... ");
            sendPacket(data);

            packetNumber++;
        }

        sendPacket("");
    }

    public static void main(String[] args) throws IOException {
        int port = 5000;
        String data = "Hi my name is Abhinav Dinesh Srivatsa.\nMy registeration
number is: 21BDS0340.";
        SlidingWindowGoBackNServer swgs = new SlidingWindowGoBackNServer(port);
        System.out.println("Waiting for connection... ");
```

```
        swgs.receivePacket();
        System.out.println();

        swgs.sendData(data);
        System.out.println("\nEnding connection.");
    }
}
```

SlidingWindowGoBackNClient.java

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Arrays;

public class SlidingWindowGoBackNClient {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public SlidingWindowGoBackNClient(InetAddress address, int port) throws
SocketException {
        this.port = port;
        this.address = address;
        socket = new DatagramSocket();
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsChar[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsChar[x] = (char) Integer.parseInt(elementsString[x]);

        return elementsChar;
    }

    public void sendPacket(String dataString) throws IOException {
        byte data[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(data, data.length, address,
port);
```

```
        System.out.println("Sending acknowledgement.");
        socket.send(packet);
    }

    public char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        return stringToCharArray(Arrays.toString(data));
    }

    public String[] receiveTransmission() throws IOException {
        String messages[] = new String[10];

        for (int x = 1; x < messages.length; x++) {
            char data[] = receivePacket();
            System.out.print("Received packet... ");

            if (data[0] == '\0')
                break;

            messages[x] = new String(data);
            String index = messages[x].substring(0, 1);
            messages[x] = messages[x].substring(1, messages[x].length() - 1);

            sendPacket(index);
        }

        System.out.println("Destroyed connection.\n");

        return messages;
    }

    public static void main(String[] args) throws IOException {
        InetAddress address = InetAddress.getByName("localhost");
        int port = 5000;
        SlidingWindowGoBackNClient swgc = new SlidingWindowGoBackNClient(address,
port);
        System.out.println("Initiating connection...");
        swgc.sendPacket("initiate");
        System.out.println();

        String messages[] = swgc.receiveTransmission();
        for (int x = 1; x < messages.length; x++)
            if (messages[x] != null)
                System.out.println("Message: " + messages[x]);
    }
}
```