Name: Mingda Zhang
Andrew ID: mingdaz

## 18-756 Coursework 1 Answer sheet

Please keep any code change descriptions and explanations (in this document) to under 10 lines – the code comments and structure should allow the TA to easily see what you have done. Also, remember to comment in your source code where you have made changes. Example:
On answer this sheet: “
Ex1)
**Explanation**: I changed the SONETRouter.receiveFrame to start a fire when it receives a frame. I did this using if(receive==true) and the .setIsOnFire(true)
**Code**:

```
//Added question 1.h) onFire code below
public SONETRouter.receiveFrame(){
// check that we did receive a frame
If(receive==true){
        // as we have received a frame set the datacenter on fire, and update the isOnFire variable
        this.startFire();
        this.setIsOnFire(true);

        // Do the next thing
        ….
}
}”
```

**ANSWERS**

1.a)
**Explanation**:  I changed the SONETRouter.**receiveFrame** to compare the frame frequency with router’s drop and destination frequency. I did this using dropFrequency.contains(wavelength) and destinationFrequencies.values().contains(wavelength)
**Code**:

```
public void receiveFrame(SONETFrame frame, int wavelength, OpticalNICTA nic)
{
          if(dropFrequency.contains(wavelength)){
     // check if  the frame is on the router drop frequency
               if(destinationFrequencies.values().contains(wavelength)){
     // if so, check if the frame is also the router destination frequency
                    sink(frame,wavelength);
               }
               else{
```

```
// take it off the line, don't know what to do now;
                                ;
                        }
                }
                else{
// if the frequency does not match any drop frequency then we forward it.
                        sendRingFrame(frame,wavelength,nic);
                }
        }
```

1.b)

**Explanation**: Because there are two lines link these two routers. And the data was sent through these two lines. Therefore router with address “88:77:66” receive  the same frame twice. This is an “1+1” protection behavior.

**Code**: sendRingFrame(frame,wavelength,nic);

1.c)

**Explanation**: Only router 1 get the message.  The router 1 source an frame and it equals to the drop frequency. Therefore it has been sent to the sink(frame,wavelength) operation branch. So the data has not been forward to router2.

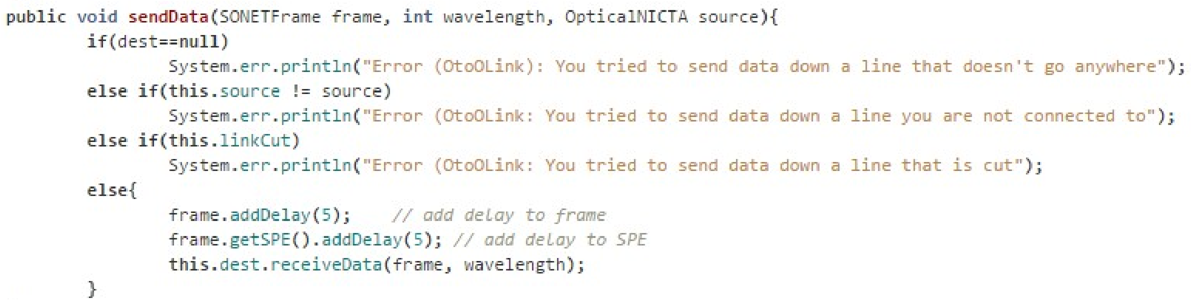
**Code**:

1.d)

**Explanation**: Modified **SPE.addDelay** and **SONETFrame.addDelay**. The realization for these two functions is the same. Also I change the **OtoOLink.sendData**

**Code**:

```
public void addDelay(int delay){
                this.delay += delay;
        }
```



```
public void sendData(SONETFrame frame, int wavelength, OpticalNICTA source){
        if(dest==null)
                System.err.println("Error (OtoOLink): You tried to send data down a line that doesn't go anywhere");
        else if(this.source != source)
                System.err.println("Error (OtoOLink: You tried to send data down a line you are not connected to");
        else if(this.linkCut)
                System.err.println("Error (OtoOLink: You tried to send data down a line that is cut");
        else{
                frame.addDelay(5);    // add delay to frame
                frame.getSPE().addDelay(5); // add delay to SPE
                this.dest.receiveData(frame, wavelength);
        }
```

1.e)

**Explanation**: Since the frame is sent to those NICs one by one, we can not let the sendFrame function directly modified fame instance. Therefore I use clone function return an deep copy of frame. In order to do that I also need to implement SPE clone() function.

**Code**:

```
public void sendRingFrame(SONETFrame frame, int wavelength, OpticalNICTA nic){
        ...
        NIC.sendFrame(frame.clone(), wavelength);
}

public SPE clone(){
                SPE newSPE = new SPE(this.getVTPointer());
                newSPE.delay = this.getDelay();
                return newSPE;
}

public SONETFrame clone(){
        SONETFrame newframe = new SONETFrame(this.getSPE().clone());
        newframe.delay = 0;
        newframe.setOAMFlags(this.getOAMFlags());
        return newframe;
}
```

1.f)

**Explanation**: I have change the order of the two if condition. If the destination drop frequency list does not contain the frequency of the packet then we do not forward it any more. If it coutains the frequency, then we judge if the current router frequency is the same with the packet frequency.

**Code**:

```
public void receiveFrame(SONETFrame frame, int wavelength, OpticalNICTA nic){
        if(destinationFrequencies.values().contains(wavelength)){
        // check if  the frame is on the router drop frequency
                if(dropFrequency.contains(wavelength)){
                // if so, check if the frame is also the router destination frequency
                        sink(frame,wavelength);
                }
                else{
                // if the frequency does not match any drop frequency then we forward it.
                        sendRingFrame(frame,wavelength,nic);
                }
        }
        else{
        // do not forward this message
                ;
        }
}
```

2.a)

**Explanation**: According to the structure given in the handout, I built 3 router each router has two interface.  Between every two routers there are two OtoOLink. Code is almost the same with ExampleTA.java a little change has made.

**Code**:

See q2a.java

2.b)

**Explanation**: Yes I get the same output.. The SPE delay is different because one go directly from “00:11:22” to “88:77:66”. The other go to “33:44:55” first. So that one need 10, the other only need 5.

**Code**:

2.c)

**Explanation**: I need to create four NIC for each router. And two uni-link between corresponding NIC.

**Code**: See code q2c.java

2.d)

**Explanation**: I have modified SONETRouter.java file. I change the receiveFrame function to parse the OAMFlags parameter. In order to select the shortest path. First, I need to send some message let the routers know their relation. It is not hard coded but by communications. The DestinationHopCount
array contains the corresponding hop number if the message sending to its destination by this NIC. I use the clockwise parameter as a flag to distinguish work line and protection line. They may be not really in the clockwise direction. At first I send data contain frequency and hop as flag in theform “frequency:hop”. For example router 1, which drop frequency is 1310,will send out “1310:1”. Each time a router send a frame to the ring it will add one hop number. For example this message is send to 33:44:55, and 88:77:66 firsts this message it will send this message to 33:44:55 with flag “1310:2”. Then 33:44:55 will set the hop count number of the NIC from which it receives “1310:2” to 2. After all router send out communication message, hop count treemap of a router would seems like:
<1310,{1,2,1,2}>
<1550,{2,1,2,1}>

**Code**: See code q2d.java, SONETRouter.java, Grade: / 40 /40 /20 /100