Experiment 9

Aim: Using JAVA RMI Write a program to implement Basic Calculator

Procedure:

Interface //RMICalIntf.java

```
import java.rmi.*;
import java.rmi.server.*;
public interface RMICalIntf extends Remote
{                       //Declaring the methods to be implemented
        double add(double a,double b) throws RemoteException;
        double sub(double a,double b) throws RemoteException;
        double mul(double a,double b) throws RemoteException;
        double div(double a,double b) throws RemoteException;
}
```

Implementation //RMICalImpl.java

```
import java.rmi.*;
import java.io.*;
import java.rmi.server.*;

public class RMICalImpl extends UnicastRemoteObject implements RMICalIntf
{                              //Defining the methods declared in the interface
RMICalImpl() throws RemoteException
{
}
        public double add(double a, double b)throws RemoteException
{
return(a+b);
           }
        public double sub(double a,double b)throws RemoteException
{
                return(a-b);
                }
        public double mul(double a,double b)throws RemoteException
{
                return(a*b);
                }
        public double div(double a,double b)throws RemoteException
{
```

```
                return(a/b);
                }
}
```

<u>Server side</u>   //RMICalServer.java

```
import java.rmi.*;
public class RMICalServer
{
        public static void main(String args[])
{
                try
{                        //creating implementation object
RMICalImpl si = new RMICalImpl();
                        Naming.rebind("calserver",si);
                        }
catch(Exception e){}
                }
        }
```

<u>Client side</u>   //RMICalClient.java

```
import javax.swing.*;
import java.awt.*;
import java.rmi.*;
import java.awt.event.*;
import java.io.*;
public class RMICalClient extends JFrame implements ActionListener
{
        double n1 = 0.0;
        double  d1;
double n2 = 0.0;
        JButton jb[] = new JButton[21];
        JTextField tf;
        Container con;
        int button,i;
        String str;
        String num="";
        JPanel tp,bp;                         //declaring 2 panels for textfield and buttons
        public RMICalClient()
```

```
{
                setTitle("calculator");
                tp = new JPanel();
                bp = new JPanel();
                tf = new JTextField(22);
                tf.setEditable(false);
                con = getContentPane();
                bp.setLayout(new GridLayout(5,4));
                tp.add(tf);
                con.add(tp,"North");                //placing the textfield in the north
                for(int i=0;i<10;i++)               //inserting and numbering buttons
{
                    jb[i] = new JButton(""+i);
}
                jb[10] = new JButton("+");
                jb[11] = new JButton("-");
                jb[12] = new JButton("*");
                jb[13] = new JButton("/");
                jb[14] = new JButton("clear");
                jb[15] = new JButton(".");
                jb[16] = new JButton("=");
                for(int i = 0;i<17;i++)
{
                        jb[i].addActionListener(this);
                        bp.add(jb[i]);
                        }
                con.add(bp,"Center"); //placing the panel with the buttons in the center
                setDefaultCloseOperation(EXIT_ON_CLOSE);
                }
public void actionPerformed(ActionEvent ae)
{
str = ae.getActionCommand();            //get the text on the button
                System.out.println(str);
                for(int i=0;i<10;i++)
{
                      if(ae.getSource()==jb[i])
{
num = num+str;     //if the button clicked is a number,
                              tf.setText(num);     // concatenate it to the string “num”
                        }
                }
                if((ae.getSource())==jb[15])                //if the button pressed is "."
{
num = num+str;                          //concatenate it to num
                        tf.setText(num);
                        }
                for(int i=10;i<14;i++)
{
                        if(ae.getSource()==jb[i])           //if the button is an operator
{
button = i-9;           //store the operator in "button"
                                if(num!="")             //obtain the first operand
{
                                tf.setText("");
                                n1 = Double.parseDouble(num);    //convert string in to
                                 num="";                         //double & store it in "n1"
                                }
Else
{
                                                tf.setText("");
                                        }
}
                        }
                        if(ae.getSource()==jb[16])          //if the button pressed is "="
{
                                if(!(tf.getText().equals("")))
{
                                        tf.setText("");
n2 = Double.parseDouble(num);//store 2nd operand
//in n2
                                        num = "";
                                                try
{
                                                String url = "rmi://localhost/calserver";
                                                        RMICalIntf          a          =(RMICalIntf)
Naming.lookup(url);
                                                switch(button)
{
                                                        case 1:
                                                                d1 = a.add(n1,n2);
                                                                break;
                                                          case 2:
                                                                d1 = a.sub(n1,n2);
                                                                break;
                                                        case 3:
                                                                d1 = a.mul(n1,n2);
                                                                break;
                                                        case 4:
                                                                d1 = a.div(n1,n2);
                                                                break;
                                                          default:
                                                                d1 = 0.0;
                                                        }
str = String.valueOf(d1);          //convert the //value to string
                                                tf.setText(str);      //print the value
                                                        }
catch(Exception e){}
                                                }
Else
{
                                tf.setText("");
                                        }
                        }
                        if(ae.getSource()==jb[14]) //if button pressed is "CLEAR"
{
                                        tf.setText("");
                                        num = "";
                                        n1=0.0;
                                        n2=0.0;
                                        button=0;
                                }
                        }
public static void main(String args[])
{
                JFrame f = new RMICalClient();
                f.setSize(200,200);
                f.setVisible(true);
                }
}
```

<u>Compilation</u>

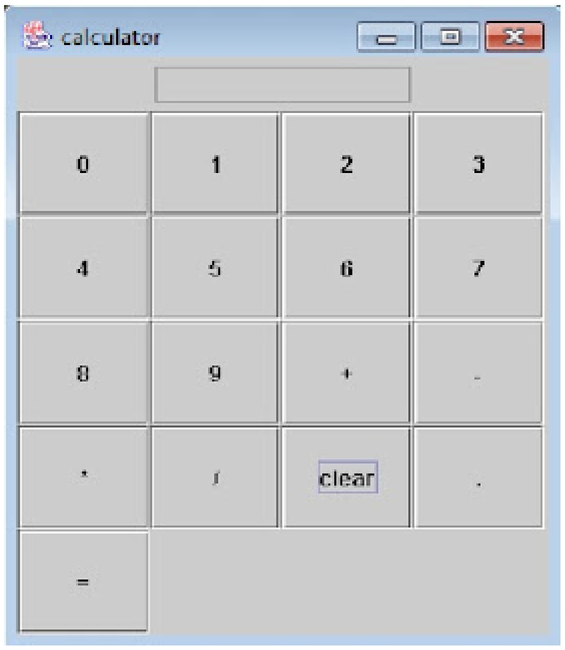
1.open command prompt
2.set the path to the specified file
3.type javac RMICalIntf.java //compiles interface
4.type javac RMICalImpl.java //compiles implementation

3.type javac RMICalServer.java //compiles server side
4.type javac RMICalClient.java //compiles client side

<u>Execution</u>

1.open command prompt
2.set the path to the specified file
3.type start rmiregistry
4.type java RMICalServer //executes server side
5. type java RMICalClient //executes client side

<u>Output</u>



Experiment 10

Aim: To work with Wire shark observe, Inspect HTTP Traffic. Apply filters to Inspect HTTP Traffic from a Given IP Address.

Requirements: Wireshark tool,Internet

Description/Procedure:

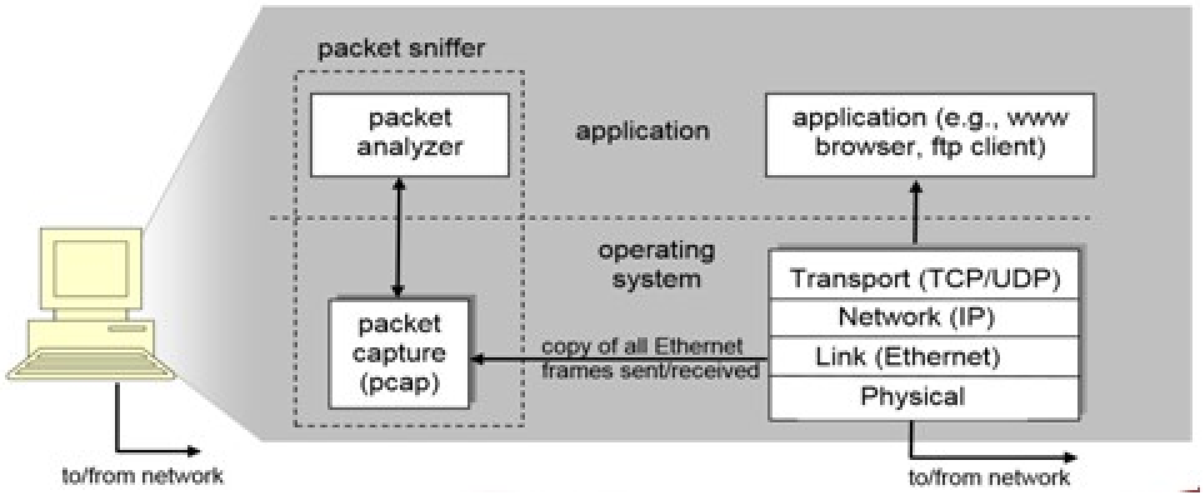
Introduction

The basic tool for observing the messages exchanged between executing protocolentities is called a **packet sniffer.**

The packet sniffer consists of 2 parts:

- The **packet capture** library receives a copy of every link layer frame that is sent from or received by your computer.

- The **packet analyzer** which displays the contents of all fields within a protocol message.

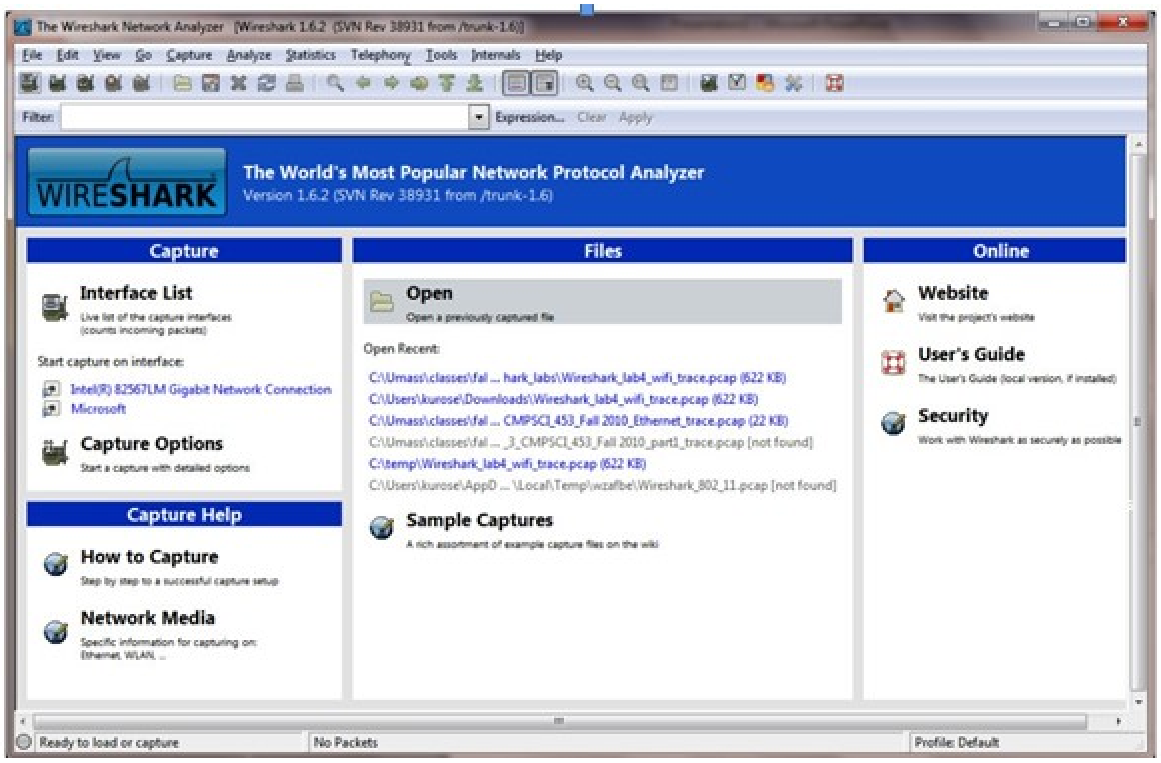


Getting Wireshark:

Wireshark is one of the best packet sniffer tools.

See http://www.wireshark.org/download.html

Running Wireshark:



The Wireshark Network Analyzer [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply

The World's Most Popular Network Protocol Analyzer
Version 1.6.2 (SVN Rev 38931 from /trunk-1.6)

Capture

Interface List
Live list of the capture interfaces (counts incoming packets)

Start capture on interface:
Intel(R) 82567LM Gigabit Network Connection
Microsoft

Capture Options
Start a capture with detailed options

Capture Help

How to Capture
Step by step to a successful capture setup

Network Media
Specific information for capturing on: Ethernet, WLAN, ...

Files

Open
Open a previously captured file

Open Recent:
C:\Umass\classes\fal ... hark_labs\Wireshark_lab4_wifi_trace.pcap (622 KB)
C:\Users\kurose\Downloads\Wireshark_lab4_wifi_trace.pcap (622 KB)
C:\Umass\classes\fal ... CMPSCI_453_Fall 2010_Ethernet_trace.pcap (22 KB)
C:\Umass\classes\fal ... _3_CMPSCI_453_Fall 2010_part1_trace.pcap [not found]
C:\temp\Wireshark_lab4_wifi_trace.pcap (622 KB)
C:\Users\kurose\AppD ... \Local\Temp\wzafbe\Wireshark_802_11.pcap [not found]

Sample Captures
A rich assortment of example capture files on the wiki

Online

Website
Visit the project's website

User's Guide
The User's Guide (local version, if installed)

Security
Work with Wireshark as securely as possible

Ready to load or capture | No Packets | Profile: Default

Filters

Table 6.4. Display Filter comparison operators

English	C-like	Description and example
eq	==	**Equal** `ip.src==10.0.0.5`
ne	!=	**Not equal** `ip.src!=10.0.0.5`
gt	>	**Greater than** `frame.len > 10`
lt	<	**Less than** `frame.len < 128`
ge	>=	**Greater than or equal to** `frame.len ge 0x100`
le	<=	**Less than or equal to** `frame.len <= 0x20`

Testing Wireshark:

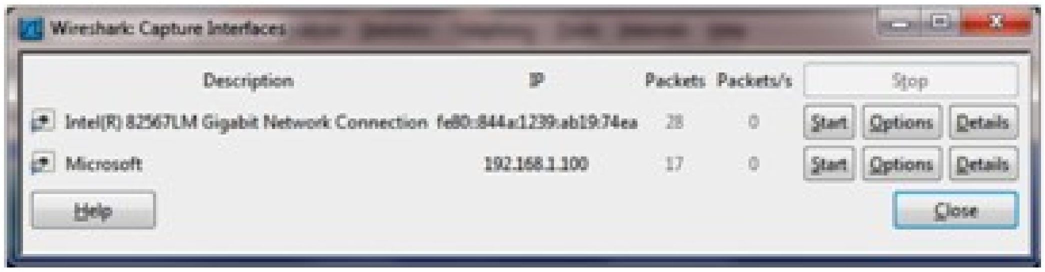
Table 6.6. Display Filter Logical Operations

English	C-like	Description and example
and	&&	**Logical AND** `ip.src==10.0.0.5 and tcp.flags.fin`
or	||	**Logical OR** `ip.scr==10.0.0.5 or ip.src==192.1.1.1`
xor	^^	**Logical XOR** `tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29`
not	!	**Logical NOT** `not llc`

1. Start up your favorite web browser, which will display your selected homepage.

2. Start up the Wireshark software. You will initially see a window similar to that shown in slide 5. Wireshark has not yet begun capturingpackets.

3. To begin packet capture, select the Capture pull down menu and select Interfaces. This will cause the “Wireshark: Capture Interfaces”window to be displayed, as shown in the figure below



Description	IP	Packets	Packets/s
Intel(R) 82567LM Gigabit Network Connection	fe80::844a:1239:ab19:74ea	28	0
Microsoft	192.168.1.100	17	0

4. Click on Start for the interface on which you want to begin packet capture (in thecase, the Gigabit network Connection). Packet capture will now begin - Wireshark is now capturing all packets being sent/received from/by your computer.

5. By selecting Capture pulldown menu and selecting Stop, you can stop packet capture. But don’t stop packet capture yet. Let’s capture some interesting packetsfirst.

6. While Wireshark is running, enter the URL: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html and have that page displayed in your browser. In order to display this page, your browser will contact the HTTP server at gaia.cs.umass.edu and exchange HTTP messages with the server in order to download this page

7. Stop Wireshark packet capture by selecting stop in the Wireshark capture window.You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the gaia.cs.umass.edu web server should appear somewhere in the listing of packets captured.

8. Type in “http” (without the quotes, and in lower case - all protocol names are in lower casein Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select Apply (to the right of where you entered “http”). This will cause only HTTP message to be displayed in the packet-listing window.

9. Find the HTTP GET message that was sent from your computer to the gaia.cs.umass.edu HTTP server. (Look for an HTTP GET message in the “listing of captured packets” portion ofthe Wireshark window (see Figure 3) that shows “GET” followed by the gaia.cs.umass.edu URL that you entered. When you select the HTTP GET message, the

Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window. By clicking on ‘+’ and ‘-‘ right-pointing and down-pointing arrowheads to the left side of the packet details window, minimize information displayed.

10. Result: Hence working with wireshark tool to observe the traffic and analysis of traffic in a network is successfully completed.