

INTERNET TECHNOLOGIES LAB



A Java Based Mailing App

Using ICMP and POP3 Protocols

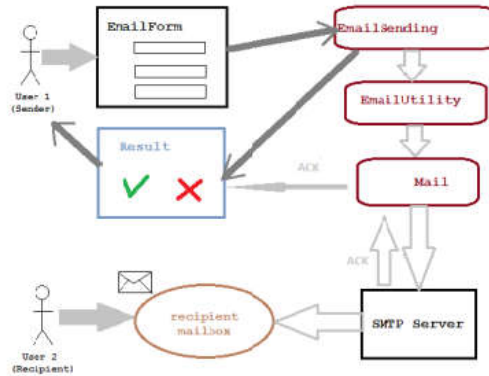
Abhijit Dutta

B.C.S.E. IV (1st sem)

001410501004

Jadavpur University

TASK



Design a Mail server and client program that implements SMTP and POP3 protocols for sending and receiving emails. An acknowledgement is sent to the sender whenever the recipient views the email. Recipient should find email notifications whenever (s)he logs in. Two possible use cases are shown above. For the details of POP3 and SMTP protocols you can refer to RFC 1939 (<http://tools.ietf.org/html/rfc1939>) and RFC 821 (<http://tools.ietf.org/html/rfc821>), respectively

Observe corresponding packet flow using Wireshark and report on traffic analysis.

CODE

MailBox.java

```
package mailapp;

import javax.mail.MessagingException;

/**
 *
 * @author Abhijit
 */
public class MailBox extends javax.swing.JFrame {

    private MailUtil mail;

    public MailBox() {
        initComponents();
        mail=new MailUtil();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
```

```
jLabel6 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jPasswordField1 = new javax.swing.JPasswordField();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jTextField2 = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

jTextField3 = new javax.swing.JTextField();

jLabel8 = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jLabel10 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();


jLabel6.setText("jLabel6");


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


jLabel1.setFont(new java.awt.Font("Times New Roman", 3, 18)); // NOI18N
jLabel1.setText("MailBox");
jLabel1.setAlignmentX(0.5F);


jTextField1.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        jTextField1ActionPerformed(evt);

    }

});
```

```
jLabel2.setFont(new java.awt.Font("Trajan Pro 3", 0, 12)); // NOI18N
```

```
jLabel2.setText("Sender");
```

```
jLabel3.setText("Email Id");
```

```
jLabel4.setText("Password");
```

```
jLabel5.setFont(new java.awt.Font("Trajan Pro 3", 0, 12)); // NOI18N
```

```
jLabel5.setText("Receiver");
```

```
jLabel7.setText("Email Id");
```

```
jLabel8.setText("Subject");
```

```
jLabel9.setFont(new java.awt.Font("Trajan Pro 3", 0, 12)); // NOI18N
```

```
jLabel9.setText("Mail");
```

```
jTextArea1.setColumns(20);
```

```
jTextArea1.setRows(5);
```

```
jScrollPane1.setViewportView(jTextArea1);
```

```
jLabel10.setText("Text");
```

```
jButton1.setText("Send Mail");
```

```
jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton1ActionPerformed(evt);
```

```
    }
```

```
});
```

```
jButton2.setText("Authenticate");
```

```

jButton2.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        jButton2ActionPerformed(evt);

    }

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(layout.createSequentialGroup()

                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .add(layout.createSequentialGroup()

                            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                                .add(layout.createSequentialGroup()

                                    .addGap(18, 18, 18)

                                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 83, javax.swing.GroupLayout.PREFERRED_SIZE)

                                    .addGap(0, 0, Short.MAX_VALUE)

                                )

                                .add(layout.createSequentialGroup()

                                    .addGroup(layout.createSequentialGroup()

                                        .addContainerGap()

                                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                                                    .addComponent(jLabel4)

                                                    .addComponent(jLabel3, javax.swing.GroupLayout.Alignment.TRAILING)

                                                    .addComponent(jLabel8, javax.swing.GroupLayout.Alignment.TRAILING)

                                                    .addComponent(jLabel7))

                                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                                                        .addComponent(jTextField3))

                                                    .addGroup(layout.createSequentialGroup()

                                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addComponent(jLabel5)

        .addGap(94, 94, 94))

    .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup())

    .addGap(30, 30, 30)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

            .addComponent(jButton2)

            .addGap(0, 0, Short.MAX_VALUE))

        .addComponent(jTextField2)

        .addComponent(jTextField1)

        .addComponent(jPasswordField1))))))

    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())

    .addGap(0, 34, Short.MAX_VALUE)

    .addComponent(jLabel10)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 261,
javax.swing.GroupLayout.PREFERRED_SIZE)))

    .addContainerGap())

    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())

    .addGap(0, 0, Short.MAX_VALUE)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())

            .addComponent(jLabel2)

            .addGap(106, 106, 106))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())

            .addComponent(jButton1)

            .addGap(32, 32, 32))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())

            .addComponent(jLabel9)

            .addGap(116, 116, 116))))))

);

layout.setVerticalGroup(

```

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel2)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel3)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel4))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel7))

.addGap(18, 18, 18)

.addComponent(jLabel9)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel8))

```



```

        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jLabel10)

            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 116,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jButton1))

    );

    pack();
} // </editor-fold>

```

```

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    //System.out.println("Password is"+new String(jPasswordField1.getPassword()));

    mail.authenticateSender(jTextField1.getText(),new String(jPasswordField1.getPassword()));

}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    try{

        mail.sendMail(jTextField2.getText(),jTextField3.getText(),jTextArea1.getText());

    }catch(Exception e){System.out.println("Error= "+e);}

}

```

```

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {

```

```

/* Set the Nimbus look and feel */

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
 * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
 */

try {
    for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(MailBox.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(MailBox.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(MailBox.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(MailBox.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new MailBox().setVisible(true);
    }
});
}

// Variables declaration - do not modify

```

```
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
// End of variables declaration
}
```

MailUtil.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package mailapp;

import java.util.Date;
import java.util.Properties;
import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

/**
 *
 * @author Abhijit
 */
public class MailUtil {

    final String SSL_FACTORY = "javax.net.ssl.SSLSocketFactory";

    private Properties props;

    private Session session;

    private String sender;

    private String receiver;

    private String password;
```

```

public MailUtil()
{
    props = System.getProperties();
    props.setProperty("mail.smtp.host", "smtp.gmail.com");
    props.setProperty("mail.smtp.socketFactory.class", SSL_FACTORY);
    props.setProperty("mail.smtp.socketFactory.fallback", "false");
    props.setProperty("mail.smtp.port", "465");
    props.setProperty("mail.smtp.socketFactory.port", "465");
    props.put("mail.smtp.auth", "true");
    props.put("mail.debug", "true");
    props.put("mail.store.protocol", "pop3");
    props.put("mail.transport.protocol", "smtp");
}

public void authenticateSender(String username,String password)
{
    session = Session.getDefaultInstance(props,new Authenticator(){
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(username, password);
        }
    });

    this.sender=username;
    this.password=password;
}

public void sendMail(String receiver,String subject,String text) throws Exception
{
    this.receiver=receiver;

    Message msg = new MimeMessage(session);

    msg.setFrom(new InternetAddress(sender));
    msg.setRecipients(Message.RecipientType.TO,

```

```
        InternetAddress.parse(receiver,false));

    msg.setSubject(subject);

    msg.setText(text);

    msg.setSentDate(new Date());

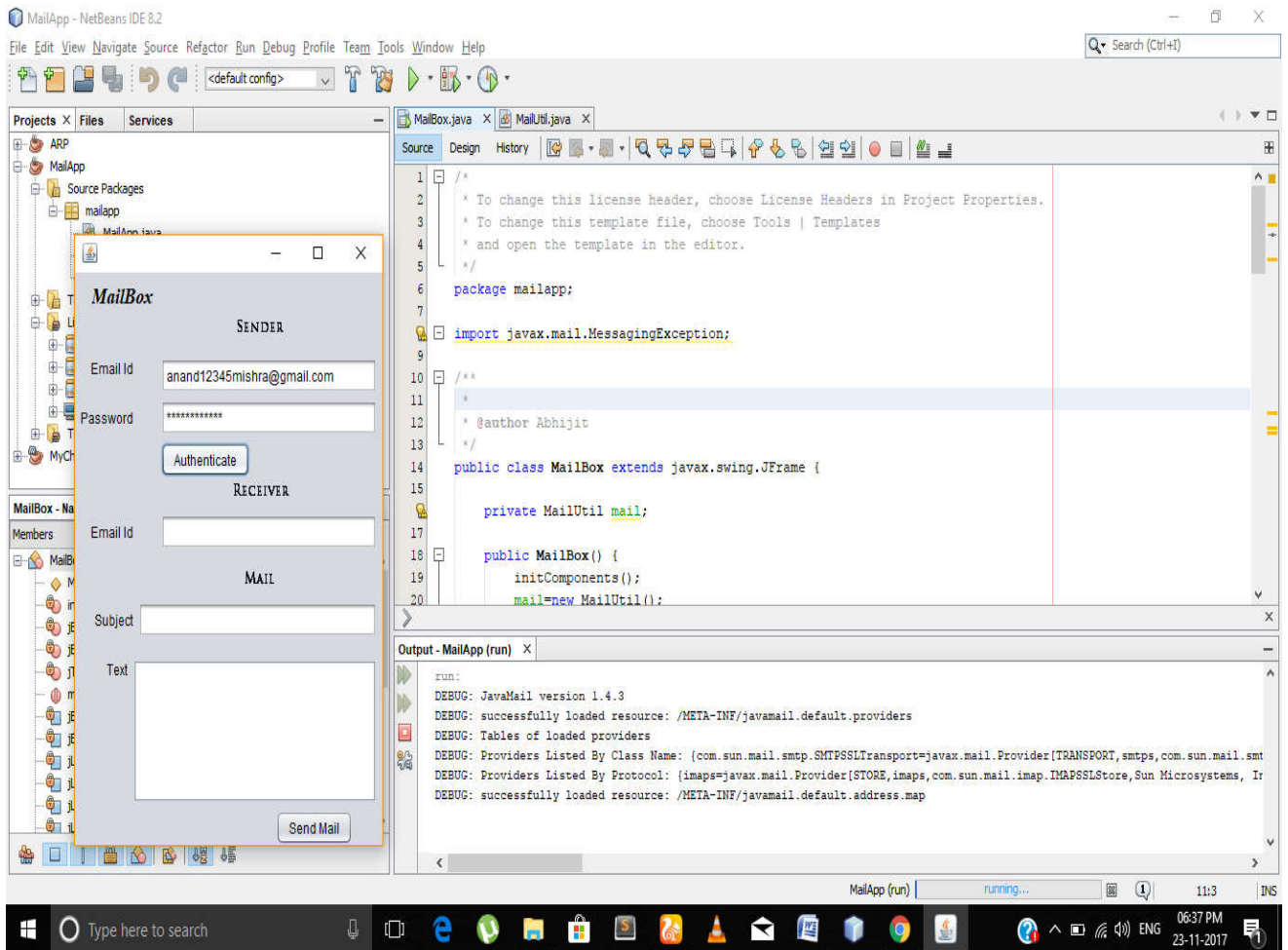
    Transport.send(msg);

    System.out.println("Message sent.");
}

}
```

OUTPUT

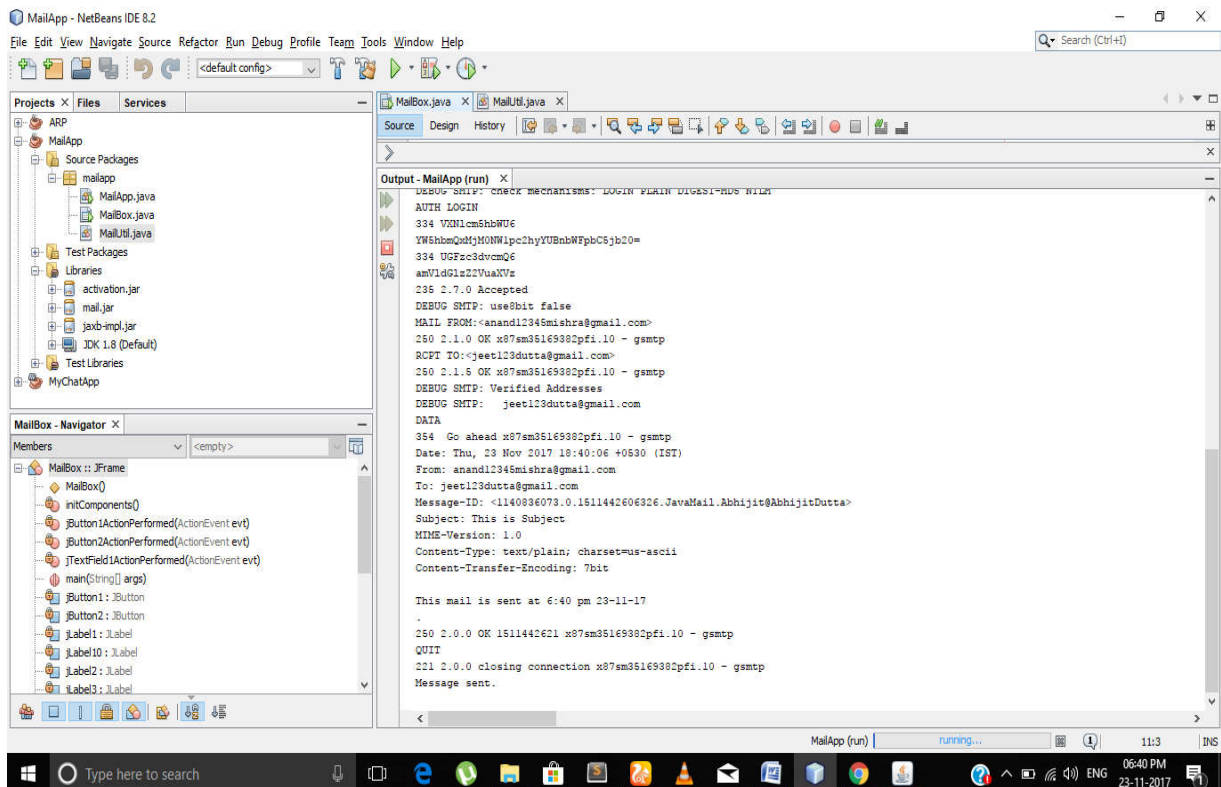
Running the MailBox.java and authenticating the Email Id of Sender (Sender Login)



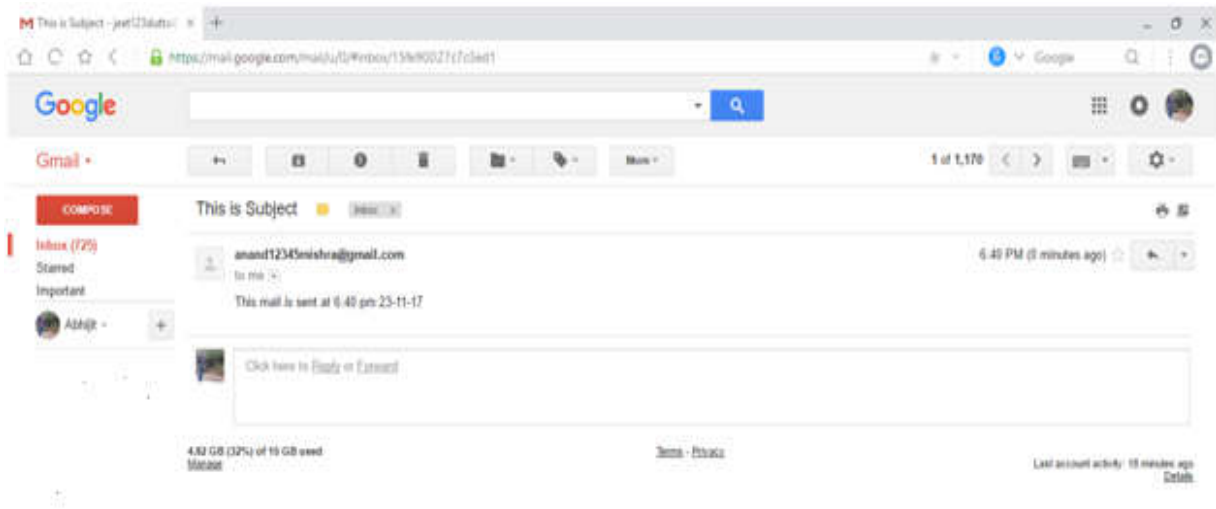
Compose the Mail to be sent .



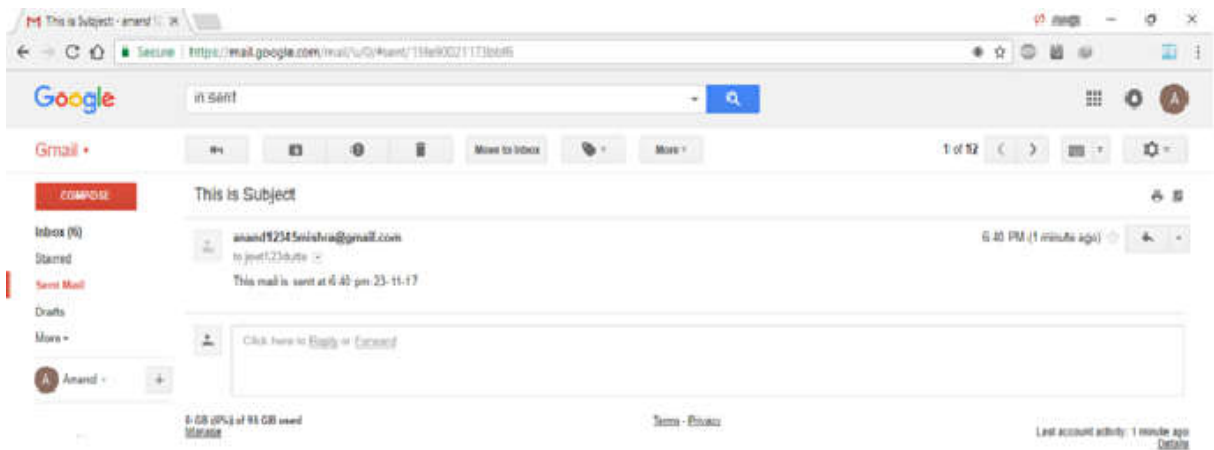
Console Log that the Mail has been sent..



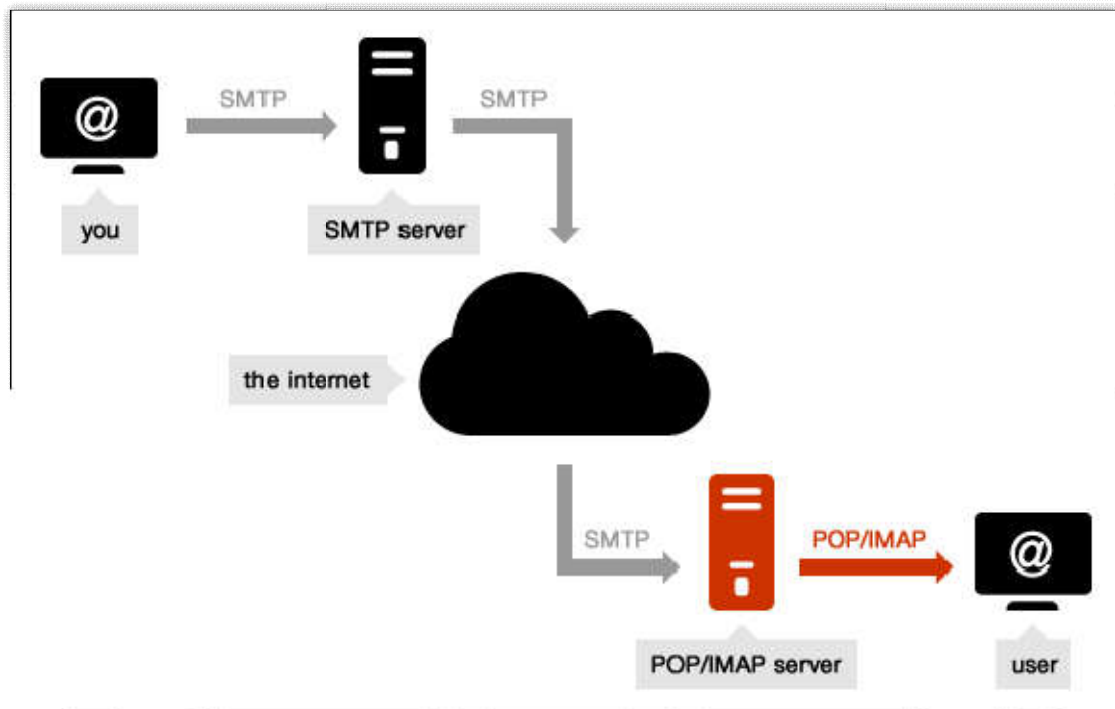
Message Received Confirmation (Receiver Email Id)



Acknowledgement Received - Message Sent Confirmation (Sender Email Id)



WORKING



SMTP and POP3 are two of the most important technical aspects of the whole process of email dispatch and delivery. To put it simply, SMTP (acronym for Simple Mail Transfer Protocol) handles the delivery, while POP (acronym for Post Office Protocol) takes care of the email retrieval and download. (Another very popular protocol that you can use in place of POP is IMAP – that is, Internet Message Access Protocol).

SMTP (Simple Mail Transfer Protocol) and POP (Post Office Protocol) are both standards used for emailing. Simply put, SMTP is used when receiving and sending emails (like your very own mailman who picks up and delivers your mail to different locations), while POP is the protocol used for the storing of emails (like your very own Post Office Box for mail storage). SMTP is the protocol in general use at the moment.

PACKET FLOW ANALYSIS

The image displays a Wireshark packet capture analysis. The top window shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 1649) is expanded to show its details: Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and QUIC. The bottom window shows the packet's raw bytes in hexadecimal and ASCII format.

No.	Time	Source	Destination	Protocol	Length	Info
1630	383.355313	192.168.0.1	239.255.255.250	SSDP	315	NOTIFY * HTTP/1.1
1631	383.357010	157.240.7.20	192.168.0.102	TCP	54	443 → 50319 [ACK] Seq=9217 Ack=2187 Win=204 Len=0
1632	383.357012	157.240.7.20	192.168.0.102	TLSv1.2	96	Application Data
1633	383.398603	192.168.0.102	157.240.7.20	TCP	54	50319 → 443 [ACK] Seq=2187 Ack=9259 Win=253 Len=0
1634	383.457757	192.168.0.1	239.255.255.250	SSDP	370	NOTIFY * HTTP/1.1
1635	383.560144	192.168.0.1	239.255.255.250	SSDP	370	NOTIFY * HTTP/1.1
1636	383.662529	192.168.0.1	239.255.255.250	SSDP	315	NOTIFY * HTTP/1.1
1637	383.764933	192.168.0.1	239.255.255.250	SSDP	354	NOTIFY * HTTP/1.1
1638	383.867372	192.168.0.1	239.255.255.250	SSDP	306	NOTIFY * HTTP/1.1
1639	383.969604	192.168.0.1	239.255.255.250	SSDP	315	NOTIFY * HTTP/1.1
1640	384.072159	192.168.0.1	239.255.255.250	SSDP	374	NOTIFY * HTTP/1.1
1641	384.174594	192.168.0.1	239.255.255.250	SSDP	368	NOTIFY * HTTP/1.1
1642	384.276877	192.168.0.1	239.255.255.250	SSDP	315	NOTIFY * HTTP/1.1
1643	384.379351	192.168.0.1	239.255.255.250	SSDP	370	NOTIFY * HTTP/1.1
1644	384.481774	192.168.0.1	239.255.255.250	SSDP	308	NOTIFY * HTTP/1.1
1645	387.246311	172.23.195.178	224.0.0.251	NDIS	83	Standard query 0x0000 PTR _sleep-proxy_udp.local, "QM" question
1646	388.948548	74.125.130.189	192.168.0.102	QUIC	81	Payload (Encrypted), PKN: 73
1647	388.973201	192.168.0.102	74.125.130.189	QUIC	80	Payload (Encrypted), PKN: 68, CID: 6739953991083793660
1648	392.734087	192.168.0.102	74.125.130.188	TCP	55	[TCP Keep-Alive] 50223 → 443 [ACK] Seq=34 Ack=34 Win=256 Len=1
1649	392.838384	74.125.130.188	192.168.0.102	TCP	66	[TCP Keep-Alive ACK] 443 → 50223 [ACK] Seq=34 Ack=35 Win=180 Len=0 SLE=34 SRE=35
1650	393.182409	192.168.0.102	74.125.130.189	QUIC	65	Payload (Encrypted), PKN: 125, CID: 3674399295196270268
1651	393.312588	74.125.130.189	192.168.0.102	QUIC	72	Payload (Encrypted), PKN: 130

Frame 1: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface 0
> Ethernet II, Src: LiteonTe_68:18:96 (30:10:b3:68:18:96), Dst: Tp-LinkT_4d:a1:08 (c4:6e:1f:4d:a1:08)
> Internet Protocol Version 4, Src: 192.168.0.102, Dst: 74.125.130.189
> User Datagram Protocol, Src Port: 62651, Dst Port: 443
> QUIC (Quick UDP Internet Connections)

0000 c4 6e 1f 4d a1 08 30 10 b3 68 18 96 08 00 45 00 .n.M..0..h....E.
0010 00 33 2f 2f 40 00 00 11 3d 42 c0 a8 00 66 4a 7d .3//@...=B...fj}
0020 82 bd f4 bb 01 bb 00 1f da 64 0c fc 9c 06 f1 71d.....q
0030 20 89 5d 22 b9 05 3a 7a 74 14 e1 5b 7d ed f7 8c .]...:z t..{}...
0040 ca