Objective:
The objective of this experiment is to investigate the performance of video streaming using different protocols in various network environments.

Submission:
This laboratory sheet together with the answers should be submitted to the lab tutor before the end of the lab session.

System Requirement:
Microsoft Windows 2000 Professional
Linux Red Hat 7.2
QuickTime Player 5.0

Required software: (stored in /eie552lab/resource)
“UDPRecv”
“RTSP Controller”

The Hong Kong Polytechnic University
Department of Electronic & Information Engineering

EIE552 Internet Technologies for Multimedia Application
Laboratory: Multimedia Streaming

Figure 1: The network environment to be set for this experiment
Part A

This part of the experiment is to create a simulated Internet environment as shown in Figure 1 to investigate how video packets are streamed from the video server to the client player.

Procedure:

Two students should work as a team with two computers allocated. One computer with two network cards will serve as the Gateway (GWx) running in Linux 7.2 and another computer (PCx) will serve as the Client running in Windows 2000 Professional. (The x denoted in GW and PC is your group number).

Step 1. Create a subnet

Configure the machines by entering the following information to GWx and PCx following the topology as shown in Figure 1.

<table>
<thead>
<tr>
<th>Machine</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCx</td>
<td>IP address: 192.168.x.2</td>
</tr>
<tr>
<td></td>
<td>Subnet mask: 255.255.255.0</td>
</tr>
<tr>
<td></td>
<td>Gateway: 192.168.x.1</td>
</tr>
<tr>
<td>GWx_1</td>
<td>IP address: 192.168.0.x</td>
</tr>
<tr>
<td></td>
<td>Subnet mask: 255.255.255.0</td>
</tr>
<tr>
<td>GWx_2</td>
<td>IP address: 192.168.x.1</td>
</tr>
<tr>
<td></td>
<td>Subnet mask: 255.255.255.0</td>
</tr>
</tbody>
</table>

For a Linux machine, to set the IP address and subnet mask to e.g. 111.111.111.111 and 255.255.255.0, respectively, to network device eth0, the following command can be used:

> ifconfig eth0 111.111.111.111 netmask 255.255.255.0

To check the result

> ifconfig

Drop down the result:

For a Linux machine, to set the default gateway of a machine to, e.g. 111.111.111.111, for network device eth0, the following command can be used:

> route add –net default eth0 gw 111.111.111.111
To check the result
   > route
Drop down the result:

Step 2.  IP forwarding
Configure the Gateway (GWx) as a software router that routes IP packets between subnets.
To allow this, a kernel module has to be loaded into the system,
   > insmod ip_tables
Then, turn on the IP forwarding mechanism,
   > echo "1">/proc/sys/net/ipv4/ip_forward
Finally, setup the forwarding rules in both directions,
   > iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
   > iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
Use a ping command to check the result,
   > ping 192.168.0.1

Step 3.  Connect to the video server
With a correct configuration, the client in PCx should be able to play a movie using QuickTime Player streamed by the video server VS1. Enter the URL of QuickTime Player as Figure 2. (You can set the QuickTime player to loop-forever by selecting an option from its menu bar).
rtsp://192.168.0.1:7070/sample.mov

**Step 4.  Determine the buffer size in the client player**

When the video is playing in the client player, disable the network connection of the client player.

*** Do not disconnect the network cable on the computer side ***

Disable the network by disconnecting the network cable between the hub and the computer on the hub side (check with the lab tutor the location of the hub). The video will cease to play after the connection is disabled, measure the approximated buffer size in terms of time.

---

**Step 5.  Determine the streaming path from VS1**

Enable the connection again by reversing the above stated procedure and playback the video again from VS1 (192.168.0.1). Remove the connection between the gateway GWx and the hub this time.

*** Do not disconnect the network cable on the computer side ***
Disconnect the cable on the hub side. Check if the video ceases to playback after the approximated buffer time as determined in step 4. Determine the streaming path from VS1.

---

**Step 6. Determine the streaming path from VS2**

Reconnect the network and set your QuickTime player to connect to the video server VS2 (192.168.0.2) for playback. Disable the connection between the gateway GWx and the hub again.

*** Do not disconnect the network cable on the computer side ***

Disconnect the cable on the hub side. Check if the video ceased to playback after the approximated buffer time as determined in step 4. Determine the streaming path from VS2.

---

**Part B**

In this part, you will use the software “NISTNet” to emulate and account for the performance of video streaming in various network conditions.

**Procedure:**

NISTNet is a network tool capable of monitoring and emulating different routes and their properties such as datagram loss, delay variation, and the network bandwidth etc.

**Step 1. Turn on NISTNet**

Turn NISTNet on by issuing the command.

>`xinstnet &`

Make sure that it is turned on by checking the status in the bottom-left corner of its GUI.
Step 2. Determine the route to monitor

Enter two pairs of address into NISTNet on the left side of the GUI. For the first pair, the IP address of GWx_2 and PCx should be entered as the source and the destination, respectively. For the second pair, the IP address of PCx and GWx_2 should be entered as the source and the destination, respectively. Click the update button to issue the command at the bottom of the GUI.

Step 3. Monitor the network traffic under normal condition

Play back the video sample.mov from video server VS1 (192.168.0.1). During playback, select the following from the menu of the QuickTime player.

Movie → Get Movie Properties → Streaming Track → Bit rate

This will give you information on the current playback, drop down the following attributes

1. Average Bit Rate
2. Maximum Bit Rate
3. Average Packet Loss Rate
4. Maximum Packet Loss Rate

Also, record your objective opinion about the quality of the play back.
Step 4. **Monitor the network traffic under various conditions**

NISTNet provides you a way to alter the condition of your network traffic. On the right hand side of its GUI, there are several columns, of which you can enter your desire value to affect the underlying network. They are

- **Delay (ms)** Control the end-to-end delay of packets
- **Drop %** Control the percentage of packets to be dropped
- **Dup %** Control the percentage of packets to be duplicated
- **Bandwidth** Control the maximum bandwidth allowed

Repeat Step 3 to obtain the result from the following 12 different conditions. You have to update NISTNet every time you modify the condition.

**Delay (ms) – 100ms**

---

**Delay (ms) – 1000ms**

---

**Delay (ms) – 5000ms**
Drop Percentage - 5%

Duplicate Percentage – 5%

Drop Percentage - 10%

Duplicate Percentage – 10%

Drop Percentage - 15%
Duplicate Percentage – 15%

Bandwidth – 10000Kbps

Bandwidth – 8000Kbps

Bandwidth – 6000Kbps
Step 5. Evaluation

Determine which factor(s) will have a dominant effect to the performance of the real-time video streaming; and which other(s) will bring no significant effect to the qualities of the video playback.

Part C

In this part, the function of Real Time Streaming Protocol (RTSP) is investigated. RTSP is an application layer protocol that provides real-time network control for video streaming. A detailed description of RTSP can be found in Appendix A. To carry out the experiment, video stream will be routed through different paths to the client. Video Server VS2 is capable of selecting the route through GW28 and GWx to deliver its packets to client player. However, its selection of route is not static over time; it will swap in every 15 seconds such that it may send through GW28 during this 15s and through GWx during the next 15s alternatively. The video datagram sent from video server VS2 to client player will therefore go through different routes and this may introduce packet delay, duplicate, loss etc.

Step 1. Download the Video file by UDP

In PCx, execute the provided program “UDPRcv” with the following command to obtain a video file from video server VS2 (192.168.0.2) port 8001 and store to a file “demo.mov”

```bash
>UDPRcv  -p 8000 -h 192.168.0.2 -p 8001 -f demo.mov
```

Use the QuickTime Player to play back the downloaded movies.

Evaluate the quality of the video play back. Suggest the reason if unsatisfactory result is obtained.
Step 2. Experiencing the RTSP conversation

Start the RTSP controller by double clicking its icon on the desktop. The following GUI will be shown:

Try to make a connection to the video server by sending RTSP commands. See Appendix A for a typical procedure of a RTSP session. To start a RTSP session, press the following buttons on the GUI in sequence:

1. Describe;
2. Setup1;
3. Setup2;
4. Play;
5. Teardown.

Note that the commands need to be sent within certain duration of time; or otherwise, the server will be timeout and stop the transaction. While you are playing the movie, you can press the pause button to pause the playback. To resume from pause, you need to press play again. When playing the movie, you will notice that the numbers on the right hand side of the GUI are advancing. They are the RTP sequence numbers of the packets received in the video and audio channels. Drop down the command statements generated and the response statements received from the server when you press the pause and teardown button.
Step 3.  **Download the Video file by Real time Streaming Protocol (RTSP)**

In PCx, play back the video file “sample.mov” from the video streaming server VS2(192.168.0.2) port 7070 by using the QuickTime Player.

Evaluate the quality of the video play back. How is it compared with that in Step 1? Can you suggest the reason why?
Appendix A
Introduction to Real Time Streaming Protocol

RTSP stands for Real Time Streaming Protocol. It is an application-level protocol enables the control of on-demand delivery of real-time data, e.g. audio, video. The first draft of protocol specification, RTSP 1.0 was submitted to IETF (Internet Engineering Task Force) on Oct 1996 by RealNetworks and Netscape. It became “Proposed Standard” on June 1998.

The major functions of RTSP include providing synchronization between media streams and allowing virtual presentation. It enables load balancing using redirection at connect and during streaming. It supports any session description and facilitates device control, including camera pan, zoom, tilt, etc.

RTSP is similar to HTTP in that it has the same format as HTTP: text based, MIME header, etc. It also has the same request / response mechanism and similar set of status codes and reason phases. However, to achieve multimedia functions, some new methods and status codes are introduced.

As an application layer protocol, RTSP does not restrict the underlying transport protocol. The media data packets can be transported using multicast UDP, unicast UDP or inline TCP (data interleaved with the RTSP control stream). A typical connection between a RTSP server and a RTSP client is as follows:

A typical RTSP session is as follows:

\[
\text{c->s: OPTIONS rtsp://foo.com:554 RTSP/1.0} \\
\text{CSeq: 1} \\
\text{s->c: RTSP/1.0 200 OK} \\
\text{CSeq: 1}
\]

Client asks server what methods and options it can support. Besides, client clarifies the protocol version. The sequence number indicates it is the first transaction. Server replies with the protocol version and the status code 200 (means OK). Server will further list the available methods supported and other options.

\[
\text{c->s: DESCRIBE rtsp://foo.com:554/single.rm RTSP/1.0} \\
\text{CSeq: 2} \\
\text{s->c: RTSP/1.0 200 OK}
\]

Client asks server what methods and options it can support. Besides, client clarifies the protocol version. The sequence number indicates it is the first transaction. Server replies with the protocol version and the status code 200 (means OK). Server will further list the available methods supported and other options.
Client asks the server to describe the details of the media file single.rm. Server indicates that the description requires 197 bytes. By default, the description is made using Session Description Protocol (SDP).

```plaintext
c->s:  SETUP rtsp://foo.com:554/single.rm RTSP/1.0  
      CSeq: 3  
      Transport: x-real-rdt/udp;client_port=6970,  
                  rtp/avp;unicast;client_port=6970-6971
s->c:  RTSP/1.0 200 OK  
       CSeq: 3  
       Session: 968367008  
       Transport: x-real-rdt/udp;client_port=6970;  
                  server_port=7970
```

Client asks the server to setup a communication channel using either Real Networks RDT or RTP transport protocol. If RDT is used, the port number of client will be 6970. If RTP is used, the port number of client will be 6970 for RTP and 6971 for RTCP. Server replies it chooses RDT. The port number of server is 7970. The session number which identifies this communication channel is 968367008.

```plaintext
c->s:  PLAY rtsp://foo.com:554/single.rm RTSP/1.0  
      CSeq: 4  
      Session: 968367008
s->c:  RTSP/1.0 200 OK  
       CSeq: 4
```

Client asks the server to start the play back using the communication channel as specified by session number 968367008.

```plaintext
c->s:  TEARDOWN rtsp://foo.com:554/single.rm RTSP/1.0  
       CSeq: 5  
       Session: 968367008
s->c:  RTSP/1.0 200 OK  
       CSeq: 5
```

Client asks the server to stop the play back that is being done with the communication channel as specified by session number 968367008.