**Project Report**

**Project Title: Advertisement Media Sharing Using TCP**

**Group Members:-**

1. Muhammad Jawwad Uddin Khan (59499)
2. Syed Hamza Ali (59035)
3. Usman Tariq ()
4. Arqum Usmani(59252)

**Introduction:-**

In our project, there is a server which stream advisement videos (send video in small chunks) to clients(Advertisement Display Screens) using TCP client server in C#. Clients will receive chunks of videos and play it on their screens.

**TCP Communication:-**

**TCP** Provides a reliable (i.e., No packet loss), connection ‐ oriented, byte ‐ Stream service to Internet Applications.

**TCP Connection** consists of a pair of byte streams, one in each direction.

**To communicate over TCP** - A client and a server each creates a TCP Socket.

**TCP Socket** -The client and the server each reads from and writes to the TCP socket.

**TCP Sockets:-**

A socket is a connection between two hosts. It can perform seven basic operations:

• Connect to a remote machine

• Send data

• Receive data

• Close a connection

• Bind to a port

• Listen for incoming data

• Accept connections from remote machines on the bound port

**The basic life cycle of a server:-**

1. A new ServerSocket is created on a particular port using a ServerSocket( ) constructor.

2. The ServerSocket listens for incoming connection attempts on that port using its accept( ) method.

3. accept( ) blocks until a client attempts to make a connection, at which point accept( ) returns a Socket object connecting the client and the server. 4. Socket's getInputStream( )method, getOutputStream( ) method, are called to get input and output streams that communicate with the client. 5. The server and the client interact according to an agreed-up on protocol until it is time to close the connection.

6. The server, the client, or both close the connection.

7. The server returns to step 2 and waits for the next connection

**Async Callbacks:-**

We use async callbacks on clients machines so that they could receive stream of video and also play that video on the screen at the same time.

**What is AsyncCallback?**

When the async method finish the processing, AsyncCallback method is automatically called, where post processing statements can be executed. With this technique there is no need to poll or wait for the async thread to complete.

**Callback Model:** The callback model requires that we specify a method to callback on and include any state that we need in the callback method to complete the call. The callback model can be seen in the following example:

static byte[] buffer = new byte[100];

static void TestCallbackAPM()

{

string filename = System.IO.Path.Combine (System.Environment.CurrentDirectory, "mfc71.pdb");

FileStream strm = new FileStream(filename,

FileMode.Open, FileAccess.Read, FileShare.Read, 1024,

FileOptions.Asynchronous);

// Make the asynchronous call

IAsyncResult result = strm.BeginRead(buffer, 0, buffer.Length,

new AsyncCallback(CompleteRead), strm);

}

In this model, we are creating a new AsyncCallback delegate, specifying a method to call (on another thread) when the operation is complete. Additionally, we are specifying some object that we might need as the state of the call. For this example, we are sending the stream object in because we will need to call EndRead and close the stream.