

Name of Experiment: Mobile IP

Exp No: MC11

Background: Student should have basic knowledge of C#.

Summary: Mobile IP (or IP mobility) is a standard communications protocol that is designed to allow mobile device users to move from one network to another while maintaining a permanent IP address. It allows location independent routing of IP datagram on the Internet. So that each mobile node is identified by its home address disregarding its current location in the Internet.

Learning Objective: Student should be able to learn the concept of Mobile IP address, by performing this experiment. He/she will get the Mobile IP of their own device by running this experiment on the physical device.

Target Platforms: This experiment is tested on Windows Emulator and Nokia Lumia 800.

Procedure:

Step1. Repeat the steps [1-4] as in experiment no MC1 (i.e. Sample experiment on Windows Phone).

Step2. Right Click on the Project in the Solution Explorer and Add new class named as MyIPAddress.cs.

Step3. Create the methods like Find, GroupJoined and DoneReceiveFromGroup.[refer source code section]

Step4. Invoke the button1_click handler.

Step5. Now create the object for the class MyIPAddress inside the IpAddress.xaml.cs.

Step6. Call the method Find through object reference.

Step7. Invoke the dispatcher on the same thread for UI updating.

Step8. Assign the result to the text block to display on the screen.

Step9. Save all the changes made to the project by pressing ctrl+S or save icon of the palette.

Step10. Press F5, to debug and run the experiment on the Windows Emulator.

Step11. By this way, we can get the Mobile IP of the device.

*Note-Please test this experiment on the Mobile device under Wi-fi environment.


```

        {
            Dispatcher.BeginInvoke(() =>
            {
                textBlock2.Text = address == null ?
"Unknown" : address.ToString();
            });
        }
    }
}

```

MyIPAddress.cs

```

using System;
using System.Net;
using System.Diagnostics;
using System.Text;
using System.Net.Sockets;

namespace MobileIP
{
    public class MyIPAddress
    {
        Action<IPAddress> FoundCallback;
        UdpAnySourceMulticastClient MulticastSocket;
        const int PortNumber = 50000; // pick a number,
any number
        string MulticastMessage = "FIND-MY-IP-PLEASE" + new
Random().Next().ToString();

        public void Find(Action<IPAddress> callback)
        {
            FoundCallback = callback;

            MulticastSocket = new
UdpAnySourceMulticastClient(IPAddress.Parse("239.255.255.250"),
PortNumber);
            MulticastSocket.BeginJoinGroup((result) =>
            {
                try
                {
                    MulticastSocket.EndJoinGroup(result);
                    GroupJoined(result);
                }
                catch (Exception ex)
                {
                    Debug.WriteLine("EndjoinGroup exception
{0}", ex.Message);
                    // This can happen eg when wifi is off
                    FoundCallback(null);
                }
            },
            null);

            void callback_send(IAsyncResult result)
            {
            }

            byte[] MulticastData;

```

← Multicast IP Addressing

```

bool keepsearching;

void GroupJoined(IAsyncResult result)
{
    MulticastData =
Encoding.UTF8.GetBytes(MulticastMessage);
    keepsearching = true;
    MulticastSocket.BeginSendToGroup(MulticastData, 0,
MulticastData.Length, callback_send, null);

    while (keepsearching)
    {
        try
        {
            byte[] buffer = new
byte[MulticastData.Length];

MulticastSocket.BeginReceiveFromGroup(buffer, 0, buffer.Length,
DoneReceiveFromGroup, buffer);
        }
        catch (Exception ex)
        {
            Debug.WriteLine("Stopped Group read due to
" + ex.Message);
            keepsearching = false;
        }
    }
}

void DoneReceiveFromGroup(IAsyncResult result)
{
    IPEndPoint where;
    int responselength =
MulticastSocket.EndReceiveFromGroup(result, out where);
    byte[] buffer = result.AsyncState as byte[];
    if (responselength == MulticastData.Length )
    {
        Debug.WriteLine("FOUND myself at " +
where.Address.ToString());
        keepsearching = false;
        FoundCallback(where.Address);
    }
}
}

```

Screenshots



Fig. no.1 Main Screen

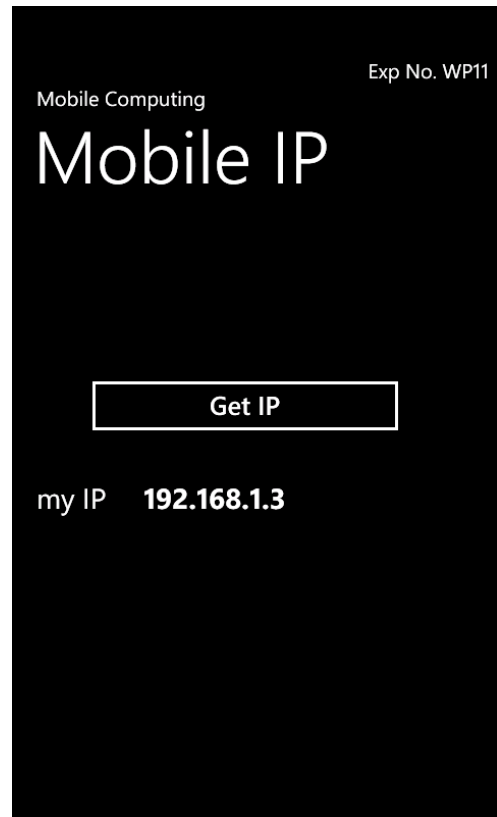


Fig. no.2 Mobile IP Address

Observations: After running this experiment, developer observed that Mobile IP can also contribute to the internet security as it has a static IP address.