```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;

public class ICMP
{
    public byte Type;
    public byte Code;
    public UInt16 Checksum;
    public int MessageSize;
    public byte[] Message = new byte[1024];
    public ICMP()
    {
    }
    public ICMP(byte[] data, int size)
    {
        Type = data[20];
        Code = data[21];
        Checksum = BitConverter.ToUInt16(data, 22);
        MessageSize = size - 24;
        Buffer.BlockCopy(data, 24, Message, 0, MessageSize);
    }
    public byte[] getBytes()
    {
        byte[] data = new byte[MessageSize + 9];
        Buffer.BlockCopy(BitConverter.GetBytes(Type), 0, data, 0, 1);
        Buffer.BlockCopy(BitConverter.GetBytes(Code), 0, data, 1, 1);
        Buffer.BlockCopy(BitConverter.GetBytes(Checksum), 0, data, 2, 2);
        Buffer.BlockCopy(Message, 0, data, 4, MessageSize);
        return data;
    }
    public UInt16 getChecksum()
    {
        UInt32 chcksm = 0;
        byte[] data = getBytes();
        int packetsize = MessageSize + 8;
        int index = 0;
        while (index < packetsize)
        {
            chcksm += Convert.ToUInt32(BitConverter.ToUInt16(data, index));
            index += 2;
        }
        chcksm = (chcksm >> 16) + (chcksm & 0xffff);
        chcksm += (chcksm >> 16);
        return (UInt16)(~chcksm);
    }
}

class TraceRoute
{
    public static void Main(string[] argv)
    {
        byte[] data = new byte[1024];
        int recv, timestart, timestop;
        Socket host = new Socket(AddressFamily.InterNetwork,SocketType.Raw,
ProtocolType.Icmp);
        IPHostEntry iphe = Dns.Resolve("www.vnexpress.net");
```

```
        IPEndPoint iep = new IPEndPoint(iphe.AddressList[0], 0);
        EndPoint ep = (EndPoint)iep;
        ICMP packet = new ICMP();
        packet.Type = 0x08;
        packet.Code = 0x00;
        packet.Checksum = 0;
        Buffer.BlockCopy(BitConverter.GetBytes(1), 0, packet.Message, 0, 2);
        Buffer.BlockCopy(BitConverter.GetBytes(1), 0, packet.Message, 2, 2);
        data = Encoding.ASCII.GetBytes("test packet");
        Buffer.BlockCopy(data, 0, packet.Message, 4, data.Length);
        packet.MessageSize = data.Length + 4;
        int packetsize = packet.MessageSize + 4;
        UInt16 chcksum = packet.getChecksum();
        packet.Checksum = chcksum;
        host.SetSocketOption(SocketOptionLevel.Socket,
                   SocketOptionName.ReceiveTimeout, 3000);
        int badcount = 0;
        for (int i = 1; i < 50; i++)
        {
            host.SetSocketOption(SocketOptionLevel.IP,
                  SocketOptionName.IpTimeToLive, i);
            timestart = Environment.TickCount;
            host.SendTo(packet.getBytes(), packetsize, SocketFlags.None, iep);
            try
            {
                data = new byte[1024];
                recv = host.ReceiveFrom(data, ref ep);
                timestop = Environment.TickCount;
                ICMP response = new ICMP(data, recv);
                if (response.Type == 11)
                    Console.WriteLine("hop {0}: response from {1}, {2}ms",
                      i, ep.ToString(), timestop - timestart);
                if (response.Type == 0)
                {
                    Console.WriteLine("{0} reached in {1} hops, {2}ms.",
                      ep.ToString(), i, timestop - timestart);
                    break;
                }
                badcount = 0;
            }
            catch (SocketException)
            {
                Console.WriteLine("hop {0}: No response from remote host", i);
                badcount++;
                if (badcount == 5)
                {
                    Console.WriteLine("Unable to contact remote host");
                    break;
                }
            }
        }

        host.Close();
        Console.ReadKey();
    }
}
```