```
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);
    }
}
public class SimplePing
{
    public static void Main()
    {
        int recv;
        byte[] data = new byte[1024];
        Socket host = new Socket(AddressFamily.InterNetwork, SocketType.Raw,
ProtocolType.Icmp);
        IPEndPoint iep = new IPEndPoint(IPAddress.Parse("8.8.8.8"), 0);
        EndPoint ep = (EndPoint)iep;
```

```
        ICMP packet = new ICMP();
        packet.Type = 0x08;
        packet.Code = 0x00;
        packet.Checksum = 0;
        Buffer.BlockCopy(BitConverter.GetBytes((short)1), 0, packet.Message, 0, 2);
        Buffer.BlockCopy(BitConverter.GetBytes((short)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);
        host.SendTo(packet.getBytes(), packetsize, SocketFlags.None, iep);
        try
        {
            data = new byte[1024];
            recv = host.ReceiveFrom(data, ref ep);
        }
        catch (SocketException)
        {
            Console.WriteLine("No response from remote host");
            return;
        }
        ICMP response = new ICMP(data, recv);
        Console.WriteLine("response from: {0}", ep.ToString());
        Console.WriteLine(" Type {0}", response.Type);
        Console.WriteLine(" Code: {0}", response.Code);
        int Identifier = BitConverter.ToInt16(response.Message, 0);
        int Sequence = BitConverter.ToInt16(response.Message, 2);
        Console.WriteLine(" Identifier: {0}", Identifier);
        Console.WriteLine(" Sequence: {0}", Sequence);
        string stringData = Encoding.ASCII.GetString(response.Message, 4,
        response.MessageSize - 4);
        Console.WriteLine(" data: {0}", stringData);
        host.Close();
        Console.ReadKey();
    }
}
```