**计算机网络实验报告**

**学号：71119103 姓名：许润**

**Lab-Unit2**

**Computer network labs**

1、Basic campus network structure

2、Basic network tools/commands (DOS)

3、Network protocol analysis (Network sniff software and protocol analysis)

4、Network design and configuration using network simulators/devices

Network application programming

**Targets**

Familiar with the basic network-related DOS commands, including

1、Ipconfig

2、Ping

3、Pathping

4、Net

5、Netstat

6、Nslookup

7、Tracert

**一、Ipconfig**

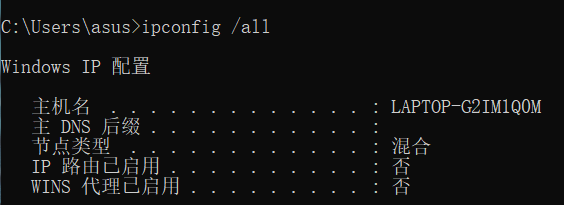
Ipconfig is a MS-DOS utility that can be used from MS-DOS and a MS-DOS shell to display the network settings currently assigned and given by a network. This command can be utilized to verify a network connection as well as to verify your network settings.

**Task:**

1、Using ipconfig to check your IP network configuration

2、Find out the meaning of all the items shown by ipconfig

3、Explain what you find



**主机名**：主机名就是计算机的名字（计算机名），网上邻居就是根据主机名来识别的，这个名字可以随时更改，从我的电脑属性的计算机名就可更改。 用户登陆时候用的是操作系统的个人用户帐号，这个也可以更改，从控制面板的用户界面里改就可以了。这个用户名和计算机名无关。因特网上的主机或 Web 站点由主机名识别。主机名有时称为域名。主机名映射到 IP 地址，但是主机名和 IP 地址之间没有一对一关系。主机名由称为 DNS 服务器或域名服务器的服务器映射到 IP 地址。DNS 代表域名服务。在大型网络中，许多 DNS 服务器可以相互协作，以提供主机名和 IP 地址之间的映射。

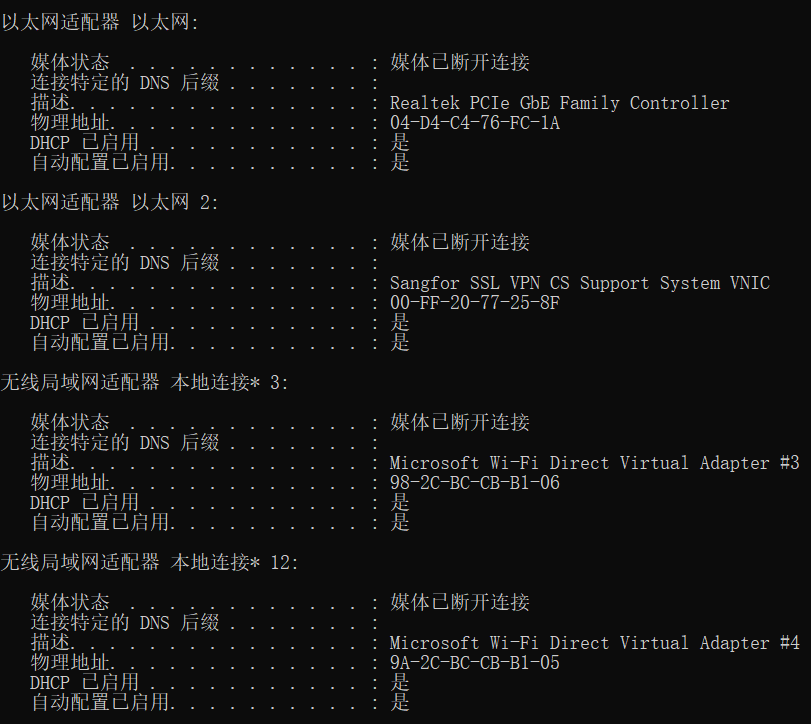
**主DNS后缀**：DNS，简单的说，就是 Domain Name System，翻成中文来讲，就是领域名称系统。在一个 TCP/IP架构的网路 (例如 Internet) 环境中，DNS是一个非常重要而且常用的系统。主要的功能就是将人易于记忆的 Domain Name 与人不容易记忆的 IP Address 做转换。事实上，将 IP Address 转换成 Domain Name 的功能也是相当常使用到的，当 login到一台Unix工作站时，工作站就会去做反查，找出是从哪个地方连线进来的。如需通过从已配置后缀列表中附加后缀的方式对不合格的域名进行解析。请选定Append these DNS suffixes（in order）并单击Add（添加）按钮以便向列表中添加后缀。如需配置针对特定连接的DNS后缀，在DNS suffix for this connection（此连接的DNS后缀）框中输入DNS后缀。

**节点类型**：在全球因特网中，每台主机和路由的每个接口都必须有一个全球唯一的 IP 地址。然而，这些地址不能以随意的方式自由选择，一个接口的 IP 地址的组成部分需要由其所连

接的子网来决定。网络设备上的接口可设为 access 接入，trunk 中继，hybrid 混合三种类型， 接口设为哪一种类型取决于相应接口的用途，access 用于直接接入网络终端，trunk 用于网络设备之间级联，hybrid 同时具备前面两种类型的特性。这与网速没有直接关系。

**IP路由已启动-否**：这条信息代表Windows的IP转发功能默认关闭，如果将其开启，则这个PC具备路由转发功能。如果这个PC有多块网卡连接不同的电脑，那这些电脑可以互相Ping通，互相访问。这个信息是随着Windows NT 3.51带来的，后来的版本已经放弃了，用组件中“路由和远程访问”来取代它了。但是仍旧可以使用，而且是在windows内核tcpip.sys直接生效，因此速度很快，性能很好，就是功能单一。配置其打开的方式是在注册表中，HKLM\system\CurrentControlSet\Service\TCPIP\Parameters中有个键值IpEnableRouter，默认为0，修改为1，重启即可。

**Wins代理已启动-否**：WINS全称Windows Internet Name Service，即Windows互联网名称服务。它和DNS一样，都是用来将主机名转换成IP地址的。但在互联网解析主机名的是DNS，事实上WINS主要的是用在局域网内缓解网络风暴。WINS 代理是一个 WINS 客户端计算机，该计算机配置为代表其他不能直接使用 WINS 的主机执行所需操作。WINS 代理帮助解析路由 TCP/IP 网络上的计算机的 NetBIOS 名称查询。 WINS 代理仅对于只包括 NetBIOS 广播（或 b 节点）客户端的网络有用或必要。对于大多数网络，一般都是启用 WINS 的客户端，因此不需要 WINS 代理。



**以太网适配器 以太网**：以太网控制器也称以太网适配器，就是我们通常称的“网卡”，其安装方法即是插在机器主板的PCI扩展槽里，一般为白色，然后安装所购买网卡中内附的驱动光盘即可。以太网控制器使用一个特定的物理层和数据链路层标准，例如以太网或令牌环来实现通讯所需要的电路系统。这为一个完整的网络协议栈提供了基础，使得在同一局域网中的小型计算机组以及通过路由协议连接的广域网，例如IP，都能够进行通讯。一块以太网控制器通常配有一个双绞线、光纤、BNC、AUI、HomePNA接口，其中后三者在现今已较少见，光纤则多用于服务器。

**无线局域网适配器 本地连接\*3/12**：本地连接（local connection）是指电脑中不同网络创建的链接，当创建家庭或小型[办公网络](https://baike.baidu.com/item/%E5%8A%9E%E5%85%AC%E7%BD%91%E7%BB%9C/522227" \t "https://baike.baidu.com/item/%E6%9C%AC%E5%9C%B0%E8%BF%9E%E6%8E%A5/_blank)时，运行 Windows 的计算机将连接到局域网 (LAN)。安装 Windows 时，将检测[网络适配器](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E9%80%82%E9%85%8D%E5%99%A8/214428" \t "https://baike.baidu.com/item/%E6%9C%AC%E5%9C%B0%E8%BF%9E%E6%8E%A5/_blank)，而且将创建本地连接。像所有其他连接类型一样，它将出现在“网络连接”文件夹中。默认情况下，本地连接始终是激活的。本地连接是唯一自动创建并激活的连接类型。“本地连接”在Windows 8及其以上版本中称为“以太网”。

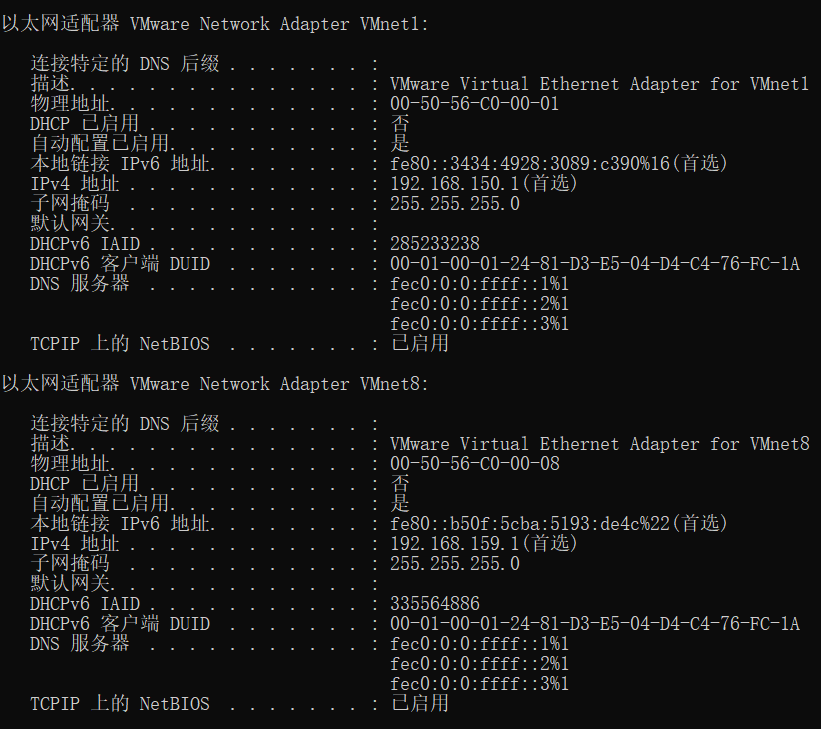
**媒体状态**：当前是否有媒体连接

**连接特定的DNS后缀**：参考主DNS后缀

**描述**：网卡的品牌型号

**物理地址**：网卡物理地址存储器中[存储单元](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%8D%95%E5%85%83/8727749" \t "https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%9C%B0%E5%9D%80/_blank)对应实际地址称物理地址，与逻辑地址相对应。网卡的物理地址通常是由[网卡](https://baike.baidu.com/item/%E7%BD%91%E5%8D%A1/155684" \t "https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%9C%B0%E5%9D%80/_blank)生产厂家写入网卡的[EPROM](https://baike.baidu.com/item/EPROM/1690813" \t "https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%9C%B0%E5%9D%80/_blank)（一种[闪存芯片](https://baike.baidu.com/item/%E9%97%AA%E5%AD%98%E8%8A%AF%E7%89%87/10365487" \t "https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%9C%B0%E5%9D%80/_blank)，通常可以通过程序擦写），它存储的是传输数据时真正赖以标识发出数据的电脑和接收数据的主机的地址。

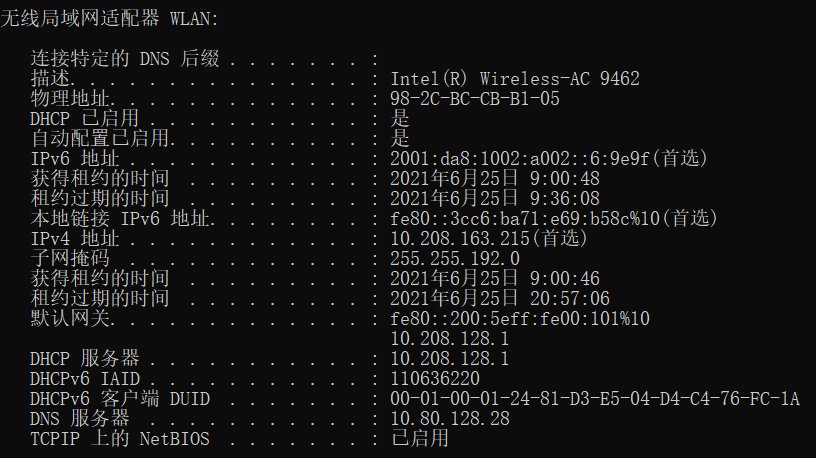
**DHCP已启用**：DHCP意思是动态主机配置协议，是一个局域网的网络协议，使用UDP协议工作。主要有两个用途：给内部网络或网络服务供应商自动分配IP地址，给用户或者内部网络管理员作为对所有计算机作中央管理的手段。DHCP已开启就是指客户机固定IP地址从DHCP服务器上取得一个暂时提供给这台机器使用的IP地址。在DHCP服务器上为这些IP地址指定了子网掩码、DNS、网关等信息。



**以太网适配器 VMware Network Adapter VMnet 1/8**:这是VMware用于虚拟桥接网络下的虚拟交换机；用于虚拟Host-Only网络下的虚拟交换机；用于虚拟NAT网络下的虚拟交换机；是Host用于与Host-Only虚拟网络进行通信的虚拟网卡；是Host用于与NAT虚拟网络进行通信的虚拟网卡；当你的计算机安装了VMware虚拟机软件时会在这自动生成的。

**DHCPv6 ：**DHCPv6是一个用来配置工作在IPv6网络上的IPv6主机所需的IP地址、IP前缀和/或其他配置的网络协议。IPv6主机可以使用无状态地址自动配置（SLAAC）或DHCPv6来获得IP地址。DHCP倾向于被用在需要集中管理主机的站点，而无状态自动配置不需要任何集中管理，因此后者更多地被用在典型家庭网络这样的场景下。使用无状态自动配置的IPv6主机可能会需要除了IP地址以外的其他信息。DHCPv6可被用来获取这样的信息，哪怕这些信息对于配置IP地址毫无用处。配置DNS服务器无需使用DHCPv6，它们可以使用无状态自动配置所需的邻居发现协议来进行配置。IPv6路由器，如家庭路由器，必须在无需人工干预的情况下被自动配置。这样的路由器不仅需要一个IPv6地址用来与上游路由器通信，还需要一个IPv6前缀用来配置下游的设备。DHCPv6 前缀代理提供了配置此类路由器的机制。

**TCPIP 上的 NetBIOS**：NetBIOS是一个网络协议，在上世纪80年代早期由IBM和Sytec联合开发，用于所谓的PC-Network。虽然公开发表的文档很少，协议的API却成为了事实上的标准。随着PC-Network被令牌环和以太网取代，NetBIOS也应该退出历史舞台。但是，由于很多软件使用了NetBIOS的API，所以NetBIOS被适配到了各种其他的协议上，比如IPX/SPX和TCP/IP。



**无线局域网适配器 WLAN**：WLAN(Wireless Local Area Network)是无线局域网的首字母缩写词，指应用无线通信技术将计算机设备互联起来，构成可以互相通信和实现资源共享的网络体系。无线局域网本质的特点是不再使用通信电缆将计算机与网络连接起来，而是通过无线的方式连接，从而使网络的构建和终端的移动更加灵活。它是相当便利的数据传输系统，它利用射频 RF(Radio Frequency)的技术，使用电磁波，取代旧式碍手碍脚的双绞铜线(Coaxial)所构成的局域网络，在空中进行通信连接，使得无线局域网络能利用简单的存取架构让用户透过它，达到“信息随身化、便利走天下”的理想境界。支持LAN的新兴无线网络标准是IEEE802.11a，其数据传输速率可达到54Mbps，另一标准IEEE802.11b的数据速率可达到11Mbps。802.11a能够同时支持更多无线用户和增强的移动多媒体应用，如数据流视频。此外，802.11a标准在无阻塞的5GHz频带上运行，从而减少了与无绳电话之间的干扰。

二、Ping

Helps in determining TCP/IP Networks IP address as well as determine issues with the network and assists in resolving them. See the ping definition for a full description.

**Tasks:**

1、Understand the principle of ping (by google)

2、Play with ping (-l, -n, -t, -r)

3、Ping –l (find out what’s the consequence of data size)

4、Ping localhost

5、Ping all the servers listed by ipconfig

6、Find out the distance (hops) between your PC and those hosts(TTL)

7、Try to find the route to other servers and their hostnames

[www.seu.edu.cn](http://www.seu.edu.cn)

[www.google.com](http://www.google.com)

**1、ping基本使用详解**

在网络中ping是一个十分强大的TCP/IP工具。它的作用主要为：

（1）、用来检测网络的连通情况和分析网络速度

（2）、根据域名得到服务器IP

（3）、根据ping返回的TTL值来判断对方所使用的操作系统及数据包经过路由器数量。

我们通常会用它来直接ping ip地址，来测试网络的连通情况。

**bytes值**：数据包大小，也就是字节。

**time值**：响应时间，这个时间越小，说明你连接这个地址速度越快。

**TTL值**：Time To Live,表示DNS记录在DNS服务器上存在的时间，它是IP协议包的一个值，告诉路由器该数据包何时需要被丢弃。可以通过Ping返回的TTL值大小，粗略地判断目标系统类型是Windows系列还是UNIX/Linux系列。

默认情况下，Linux系统的TTL值为64或255，WindowsNT/2000/XP系统的TTL值为128，Windows98系统的TTL值为32，UNIX主机的TTL值为255。因此一般TTL值：

① 100~130ms之间，Windows系统 ；

② 240~255ms之间，UNIX/Linux系统。

ping命令除了直接ping网络的ip地址，验证网络畅通和速度之外，它还有这些用法。

用法: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]

[-r count] [-s count] [[-j host-list] | [-k host-list]]

[-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]

[-4] [-6] target\_name

选项:

-t Ping 指定的主机，直到停止。

若要查看统计信息并继续操作，请键入 Ctrl+Break；

若要停止，请键入 Ctrl+C。

-a 将地址解析为主机名。

-n count 要发送的回显请求数。

-l size 发送缓冲区大小。

-f 在数据包中设置“不分段”标记(仅适用于 IPv4)。

-i TTL 生存时间。

-v TOS 服务类型(仅适用于 IPv4。该设置已被弃用，

对 IP 标头中的服务类型字段没有任何

影响)。

-r count 记录计数跃点的路由(仅适用于 IPv4)。

-s count 计数跃点的时间戳(仅适用于 IPv4)。

-j host-list 与主机列表一起使用的松散源路由(仅适用于 IPv4)。

-k host-list 与主机列表一起使用的严格源路由(仅适用于 IPv4)。

-w timeout 等待每次回复的超时时间(毫秒)。

-R 同样使用路由标头测试反向路由(仅适用于 IPv6)。

根据 RFC 5095，已弃用此路由标头。

如果使用此标头，某些系统可能丢弃

回显请求。

-S srcaddr 要使用的源地址。

-c compartment 路由隔离舱标识符。

-p Ping Hyper-V 网络虚拟化提供程序地址。

-4 强制使用 IPv4。

-6 强制使用 IPv6。

1. **ping - t的使用**

不间断地 Ping 指定计算机，直到管理员中断



这就说明电脑连接路由器是通的，网络效果很好。下面按按住键盘的Ctrl+c终止它继续ping下去，就会停止了，会总结出运行的数据包有多少，通断的有多少了。

1. **ping -a的使用**

ping-a解析计算机名与NetBios名。就是可以通过ping它的ip地址，可以解析出主机名。

1. **ping -n的使用**

在默认情况下，一般都只发送四个数据包，通过这个命令可以自己定义发送的个数，对衡量网络速度很有帮助，比如我想测试发送10个数据包的返回的平均时间为多少，最快时间为多少，最慢时间为多少就可以通过以下获知：



从以上我就可以知道在发送10个数据包的过程当中，返回了10个，没有丢失，这10个数据包当中返回速度最快为33ms，最慢为34ms，平均速度为34ms。说明我的网络良好。

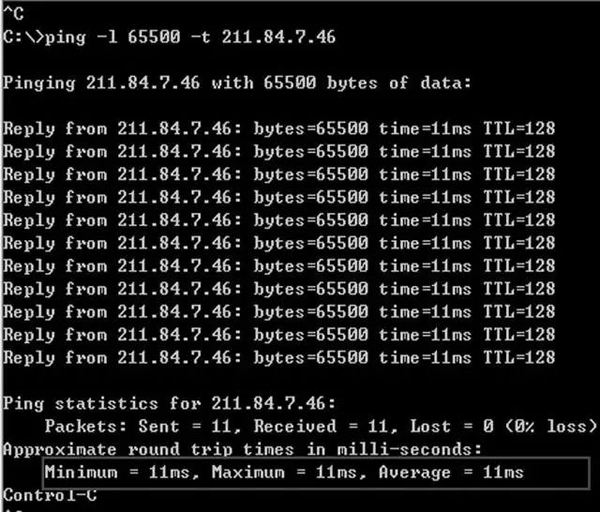
如果对于一些不好的网络，比如监控系统中非常卡顿，这样测试，返回的结果可能会显示出丢失出一部分，如果丢失的比较多的话，那么就说明网络不好，可以很直观的判断出网络的情况。

1. **ping -l的使用**

-l size：发送size指定大小的到目标主机的数据包。

在默认的情况下Windows的ping发送的数据包大小为32byt，最大能发送65500byt。当一次发送的数据包大于或等于65500byt时，将可能导致接收方计算机宕机。所以微软限制了这一数值;这个参数配合其它参数以后危害非常强大，比如攻击者可以结合-t参数实施DOS攻击。(所以它具有危险性，不要轻易向别人计算机使用)。

例如：ping -l 65500 -t 211.84.7.46会连续对IP地址执行ping命令，直到被用户以Ctrl+C中断。



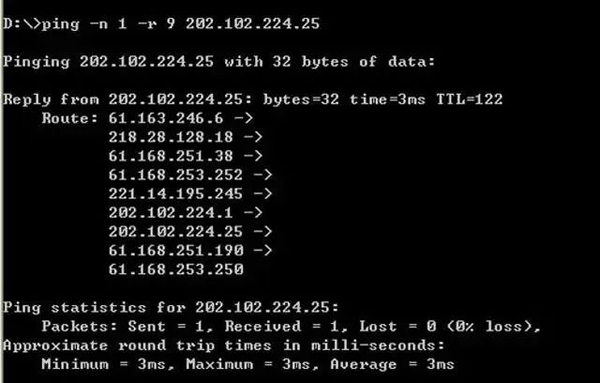
这样它就会不停的向211.84.7.46计算机发送大小为65500byt的数据包，如果你只有一台计算机也许没有什么效果，但如果有很多计算机那么就可以使对方完全瘫痪，网络严重堵塞，由此可见威力非同小可。

1. **ping -r的使用**

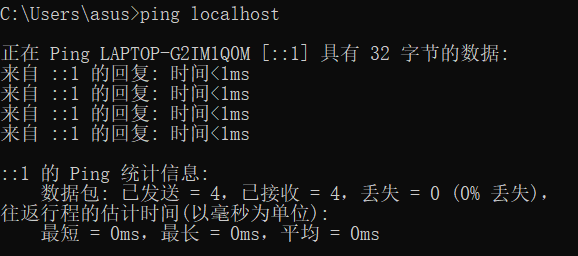
在“记录路由”字段中记录传出和返回数据包的路由，探测经过的

路由个数，但最多只能跟踪到9个路由。

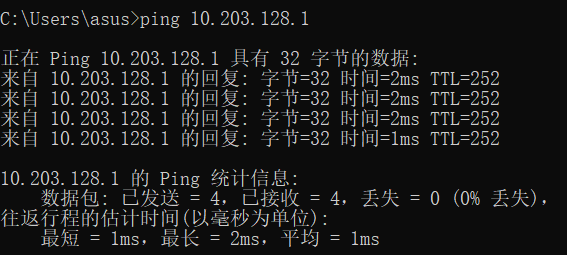
ping -n 1 -r 9 202.102.224.25 (发送一个数据包，最多记录9个路由)



1. **ping localhost的使用**



1. **ping all the servers listed by ipconfig**



三、Pathing,tracert

The tracert command is used to visually see a network packet being sent and received and the amount of hops required for that packet to get to its destination.

**Tasks:**

1、Understand the principle of pathping / tracert (by google)

2、Play with pathping and tracert

3、Try them on

[www.seu.edu.cn](http://www.seu.edu.cn)

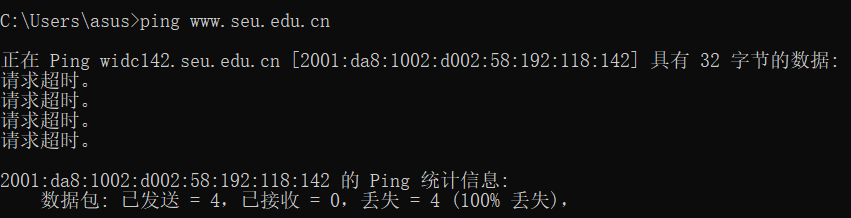
[www.google.com](http://www.google.com)

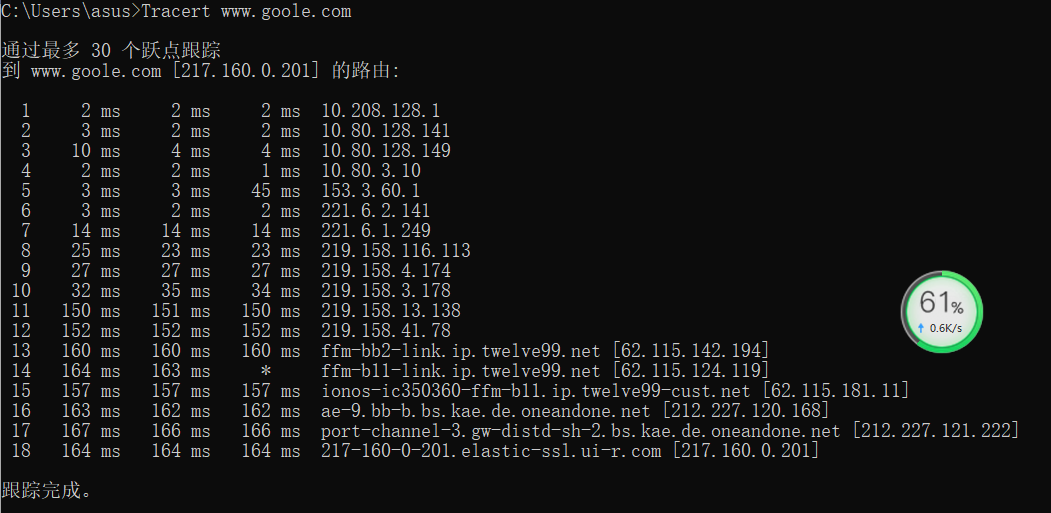
4、Compare your result with the routes obtained by ping –r

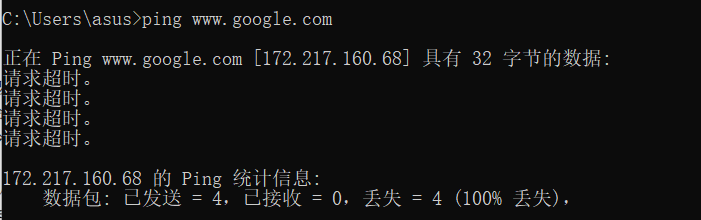
Find their relationship/difference

1. **tracert**







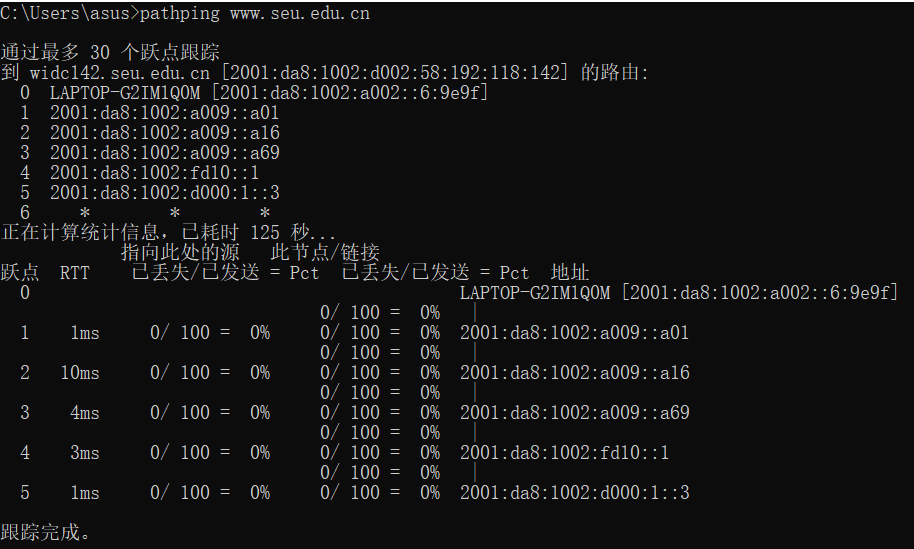


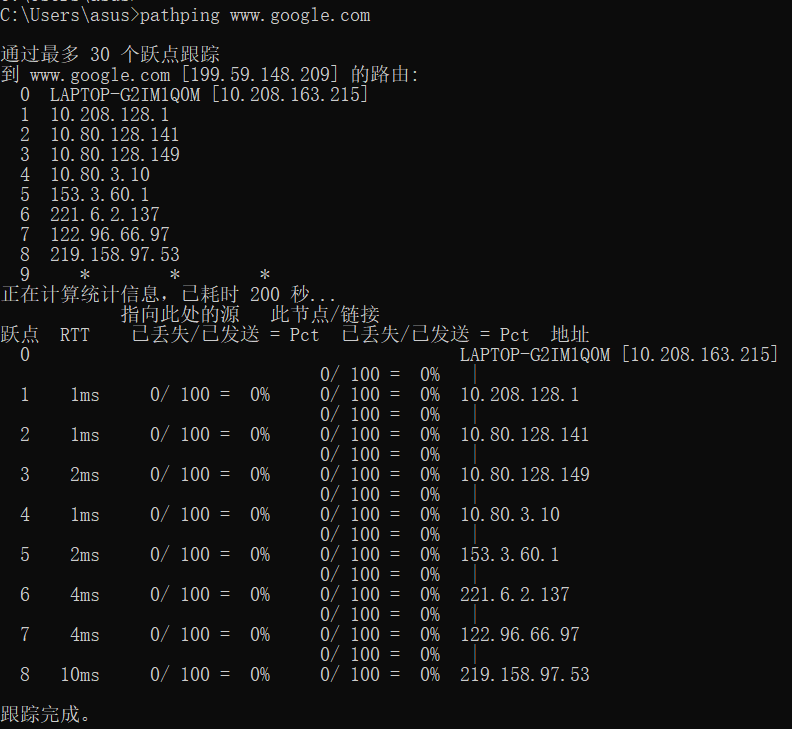
Tracert 命令用 IP 生存时间 (TTL) 字段和 ICMP 错误消息来确定从一个主机到网络上其他主机的路由。

首先，tracert送出一个TTL是1的IP 数据包到目的地，当路径上的第一个路由器收到这个数据包时，它将TTL减1。此时，TTL变为0，所以该路由器会将此数据包丢掉，并送回一个「ICMP time exceeded」消息（包括发IP包的源地址，IP包的所有内容及路由器的IP地址），tracert 收到这个消息后，便知道这个路由器存在于这个路径上，接着tracert 再送出另一个TTL是2 的数据包，发现第2 个路由器...... tracert 每次将送出的数据包的TTL 加1来发现另一个路由器，这个重复的动作一直持续到某个数据包 抵达目的地。当数据包到达目的地后，该主机则不会送回ICMP time exceeded消息，一旦到达目的地，由于tracert通过UDP数据包向不常见端口(30000以上)发送数据包，因此会收到「ICMP port unreachable」消息，故可判断到达目的地。

tracert 有一个固定的时间等待响应(ICMP TTL到期消息)。如果这个时间过了，它将打印出一系列的\*号表明：在这个路径上，这个设备不能在给定的时间内发出ICMP TTL到期消息的响应。然后，Tracert给TTL记数器加1，继续进行。

**2.pathping**





Pathping是一个基于TCP/IP的路由跟踪工具，该命令结合了 ping 和 tracert 命令的功能，返回两部分内容， 第一部分显示到达目的地经过了哪些路由，第二部分显示了路径中每个路由器上数据包丢失方面的信息。反映出数据包从源主机到目标主机所经过的路径、网络延时以及丢包率，能够有效地帮助我们解决网络问题。

        它使用ICMP回应信息来分析网络连通情况。Pathping发送回应信息到源地址与目标地址之间的所有路由器，它会计算出源主机与目标主机之间的每一个路由器的反应时间，通过这些数据，我们能够很方便地测出网络延时，有助于找出带宽瓶颈。

        提供有关在源和目标之间的中间跃点处网络滞后和网络丢失的信息。Pathping 在一段时间内将多个回响请求消息发送到源和目标之间的各个路由器，然后根据各个路由器返回的数据包计算结果。因为 pathping 可以表示在任何特定路由器或链接处的数据包的丢失程度，所以用户可据此确定可能存在网络问题的路由器或子网。Pathping 通过识别路径上的路由器来执行与 tracert 命令相同的功能。然后，该命令在一段指定的时间内定期将 ping 命令发送到所有的路由器，并根据每个路由器的返回非数值计算统计结果。

**Ping、Tracert、Pathping三个命令的简单对比：**

        Ping：用来检测目标主机的连通性，确认你想要去的地方是否可达？

        Tracert：用来显示数据包到达目标主机所经过的路径，并显示到达每个节点的时间，即了解你想去的地方如何走？

        Pathping：用来跟踪在源和目标之间的中间跃点处网络滞后和网络丢失的详细信息，即了解你所走的路径的路状如何？

        使用时不指定参数，pathping 则显示帮助。

1. **Netstat**

The netstat command is used to display the TCP/IP network protocol statistics and information.

**Tasks:**

1、Find out the principle of Netstat

2、Find out how many connections (TCP/UDP) established on your pc

3、Find out the processes who is communicating with others

4、Find out the result of

Netstat –r , -e , -s

5、Explain your understanding of the results

Netstat 是一款命令行工具，可用于列出系统上所有的网络套接字连接情况，包括 tcp, udp 以及 unix 套接字，另外它还能列出处于监听状态（即等待接入请求）的套接字。如果你想确认系统上的 Web 服务有没有起来，你可以查看80端口有没有打开。以上功能使 netstat 成为网管和系统管理员的必备利器。在这篇教程中，我会列出几个例子，教大家如何使用 netstat 去查找网络连接信息和系统开启的端口号。

-a或--all：显示所有连线中的Socket；

-A<网络类型>或--<网络类型>：列出该网络类型连线中的相关地址；

-c或--continuous：持续列出网络状态；

-C或--cache：显示路由器配置的快取信息；

-e或--extend：显示网络其他相关信息；

-F或--fib：显示FIB；

-g或--groups：显示多重广播功能群组组员名单；

-h或--help：在线帮助；

-i或--interfaces：显示网络界面信息表单；

-l或--listening：显示监控中的服务器的Socket；

-M或--masquerade：显示伪装的网络连线；

-n或--numeric：直接使用ip地址，而不通过域名服务器；

-N或--netlink或--symbolic：显示网络硬件外围设备的符号连接名称；

-o或--timers：显示计时器；

-p或--programs：显示正在使用Socket的程序识别码和程序名称；

-r或--route：显示Routing Table；

-s或--statistice：显示网络工作信息统计表；

-t或--tcp：显示TCP传输协议的连线状况；

-u或--udp：显示UDP传输协议的连线状况；

-v或--verbose：显示指令执行过程；

-V或--version：显示版本信息；

-w或--raw：显示RAW传输协议的连线状况；

-x或--unix：此参数的效果和指定"-A unix"参数相同；

--ip或--inet：此参数的效果和指定"-A inet"参数相同。

下面分析每一项的含义

**Proto**:协议名（tcp协议还是udp协议)；

**recv-Q**:网络接收队列,表示收到的数据已经在本地接收缓冲，但是还有多少没有被进程取走，recv()如果接收队列Recv-Q一直处于阻塞状态，可能是遭受了拒绝服务 denial-of-service 攻击；

**send-Q**:网路发送队列.对方没有收到的数据或者说没有Ack的,还是本地缓冲区.如果发送队列Send-Q不能很快的清零，可能是有应用向外发送数据包过快，或者是对方接收数据包不够快；这两个值通常应该为0，如果不为0可能是有问题的。packets在两个队列里都不应该有堆积状态。可接受短暂的非0情况。

**Local Address 解释**

1)Local Address 部分的0.0.0.0:873表示监听服务器上所有ip地址的所有(0.0.0.0表示本地所有ip)，比如你的服务器是有172.172.230.210和172.172.230.11两个ip地址，那么0.0.0.0:873此时表示监听172.172.230.210,172.172.230.211,127.0.0.1三个地址的873端口.

2)127.0.0.1:25这个表示监听本机的loopback地址的25端口(如果某个服务只监听了回环地址，那么只能在本机进行访问，无法通过tcp/ip 协议进行远程访问)

3)192.168.1.81:2288这是因为我们在启动的时候指定了192.168.1.81:2288参数，如果不指定的话，会监听0.0.0.0：2288

**Foreign Address解释**

与本机端口通信的外部socket。显示规则与Local Address相同

**State解释**

链路状态，共有11种

state列共有12中可能的状态，前面11种是按照TCP连接建立的三次握手和TCP连接断开的四次挥手过程来描述的。

**LISTEN** ：首先服务端需要打开一个socket进行监听，状态为LISTEN./\*The socket is listening for incoming  connections. 侦听来自远方TCP端口的连接请求 \*/

**SYN\_SENT**：客户端通过应用程序调用connect进行activeopen.于是客户端tcp发送一个SYN以请求建立一个连接.之后状态SYN\_SENT。/\*The socket is actively attempting to establish aconnection. 在发送连接请求后等待匹配的连接请求 \*/

**SYN\_RECV**：服务端应发出ACK确认客户端的 SYN,同时自己向客户端发送一个SYN.之后状态置为SYN\_RECV/\* A connection request has been received from the network. 在收到和发送一个连接请求后等待对连接请求的确认 \*/

**ESTABLISHED**：代表一个打开的连接，双方可以进行或已经在数据交互了。/\* The socket has an established connection. 代表一个打开的连接，数据可以传送给用户 \*/

**FIN\_WAIT1**：主动关闭(activeclose)端应用程序调用close，于是其TCP发出FIN请求主动关闭连接，之后进入FIN\_WAIT1状态./\* The socket is closed, and the connection is shutting down. 等待远程TCP的连接中断请求，或先前的连接中断请求的确认 \*/

**CLOSE\_WAIT**：被动关闭(passiveclose)端TCP接到FIN后，就发出ACK以回应FIN请求(它的接收也作为文件结束符传递给上层应用程序),并进入CLOSE\_WAIT./\* The remote end has shut down, waiting for the socketto close. 等待从本地用户发来的连接中断请求 \*/

**FIN\_WAIT2**：主动关闭端接到ACK后，就进入了FIN-WAIT-2./\* Connection is closed, and the socket is waiting for a shutdownfrom the remote end. 从远程TCP等待连接中断请求 \*/

**LAST\_ACK**：被动关闭端一段时间后，接收到文件结束符的应用程 序将调用CLOSE关闭连接。这导致它的TCP也发送一个 FIN,等待对方的ACK.就进入了LAST-ACK./\* The remote end has shut down, and the socket is closed. Waiting foracknowledgement. 等待原来发向远程TCP的连接中断请求的确认 \*/

**TIME\_WAIT**：在主动关闭端接收到FIN后，TCP 就发送ACK包，并进入TIME-WAIT状态。/\* Thesocket is waiting after close to handle packets still in the network.等待足够的时间以确保远程TCP接收到连接中断请求的确认\*/

**CLOSING**：比较少见./\* Bothsockets are shut down but we still don’t have all our datasent. 等待远程TCP对连接中断的确认 \*/

**CLOSED**：被动关闭端在接受到ACK包后，就进入了closed的状态。连接结束./\*The socket is not being used. 没有任何连接状态 \*/

**UNKNOWN**：未知的Socket状态。/\* Thestate of the socket is unknown. \*/

**备注:**

**SYN:** (同步序列编号,SynchronizeSequence Numbers)该标志仅在三次握手建立TCP连接时有效。表示一个新的TCP连接请求。

**ACK:** (确认编号,AcknowledgementNumber)是对TCP请求的确认标志,同时提示对端系统已经成功接收所有数据。

**FIN:** (结束标志,FINish)用来结束一个TCP回话.但对应端口仍处于开放状态,准备接收后续数据。

**1.netstat -r**

C:\Users\asus>netstat -r

===========================================================================

接口列表

12...04 d4 c4 76 fc 1a ......Realtek PCIe GbE Family Controller

5...00 ff 20 77 25 8f ......Sangfor SSL VPN CS Support System VNIC

18...98 2c bc cb b1 06 ......Microsoft Wi-Fi Direct Virtual Adapter #3

19...9a 2c bc cb b1 05 ......Microsoft Wi-Fi Direct Virtual Adapter #4

16...00 50 56 c0 00 01 ......VMware Virtual Ethernet Adapter for VMnet1

22...00 50 56 c0 00 08 ......VMware Virtual Ethernet Adapter for VMnet8

10...98 2c bc cb b1 05 ......Intel(R) Wireless-AC 9462

1...........................Software Loopback Interface 1

===========================================================================

IPv4 路由表

===========================================================================

活动路由:

网络目标 网络掩码 网关 接口 跃点数

0.0.0.0 0.0.0.0 10.208.128.1 10.208.163.215 45

10.208.128.0 255.255.192.0 在链路上 10.208.163.215 301

10.208.163.215 255.255.255.255 在链路上 10.208.163.215 301

10.208.191.255 255.255.255.255 在链路上 10.208.163.215 301

127.0.0.0 255.0.0.0 在链路上 127.0.0.1 331

127.0.0.1 255.255.255.255 在链路上 127.0.0.1 331

127.255.255.255 255.255.255.255 在链路上 127.0.0.1 331

192.168.150.0 255.255.255.0 在链路上 192.168.150.1 291

192.168.150.1 255.255.255.255 在链路上 192.168.150.1 291

192.168.150.255 255.255.255.255 在链路上 192.168.150.1 291

192.168.159.0 255.255.255.0 在链路上 192.168.159.1 291

192.168.159.1 255.255.255.255 在链路上 192.168.159.1 291

192.168.159.255 255.255.255.255 在链路上 192.168.159.1 291

224.0.0.0 240.0.0.0 在链路上 127.0.0.1 331

224.0.0.0 240.0.0.0 在链路上 192.168.150.1 291

224.0.0.0 240.0.0.0 在链路上 192.168.159.1 291

224.0.0.0 240.0.0.0 在链路上 10.208.163.215 301

255.255.255.255 255.255.255.255 在链路上 127.0.0.1 331

255.255.255.255 255.255.255.255 在链路上 192.168.150.1 291

255.255.255.255 255.255.255.255 在链路上 192.168.159.1 291

255.255.255.255 255.255.255.255 在链路上 10.208.163.215 301

===========================================================================

永久路由:

无

IPv6 路由表

===========================================================================

活动路由:

接口跃点数网络目标 网关

10 301 ::/0 fe80::200:5eff:fe00:101

1 331 ::1/128 在链路上

10 301 2001:da8:1002:a002::6:9e9f/128

在链路上

16 291 fe80::/64 在链路上

22 291 fe80::/64 在链路上

10 301 fe80::/64 在链路上

16 291 fe80::3434:4928:3089:c390/128

在链路上

10 301 fe80::3cc6:ba71:e69:b58c/128

在链路上

22 291 fe80::b50f:5cba:5193:de4c/128

在链路上

1 331 ff00::/8 在链路上

16 291 ff00::/8 在链路上

22 291 ff00::/8 在链路上

10 301 ff00::/8 在链路上

===========================================================================

永久路由:

无

1. **netstat -s**

C:\Users\asus>netstat -s

IPv4 统计信息

接收的数据包 = 96626

接收的标头错误 = 0

接收的地址错误 = 0

转发的数据报 = 0

接收的未知协议 = 0

丢弃的接收数据包 = 362

传送的接收数据包 = 99349

输出请求 = 78928

路由丢弃 = 0

丢弃的输出数据包 = 0

输出数据包无路由 = 0

需要重新组合 = 26

重新组合成功 = 4

重新组合失败 = 0

数据报分段成功 = 10

数据报分段失败 = 0

分段已创建 = 450

IPv6 统计信息

接收的数据包 = 33986

接收的标头错误 = 0

接收的地址错误 = 2

转发的数据报 = 0

接收的未知协议 = 0

丢弃的接收数据包 = 20

传送的接收数据包 = 35632

输出请求 = 18187

路由丢弃 = 0

丢弃的输出数据包 = 0

输出数据包无路由 = 0

需要重新组合 = 0

重新组合成功 = 0

重新组合失败 = 0

数据报分段成功 = 0

数据报分段失败 = 0

分段已创建 = 0

ICMPv4 统计信息

已接收 已发送

消息 1010 1010

错误 0 0

目标不可达 109 90

超时 58 0

参数问题 0 0

源抑制 0 0

重定向 0 0

回显回复 842 0

回显 1 920

时间戳 0 0

时间戳回复 0 0

地址掩码 0 0

地址掩码回复 0 0

路由器请求 0 0

路由器播发 0 0

ICMPv6 统计信息

已接收 已发送

消息 597 639

错误 0 0

目标不可达 0 0

数据包太大 0 0

超时 20 0

参数问题 0 0

回显 8 546

回显回复 508 8

MLD 查询 0 0

MLD 报告 0 0

MLD 已完成 0 0

路由器请求 0 16

路由器播发 9 0

邻居请求 5 54

邻居播发 47 15

重定向 0 0

路由器重新编号 0 0

IPv4 的 TCP 统计信息

主动开放 = 4386

被动开放 = 194

失败的连接尝试 = 945

重置连接 = 316

当前连接 = 56

接收的分段 = 108256

发送的分段 = 33366

重新传输的分段 = 0

IPv6 的 TCP 统计信息

主动开放 = 669

被动开放 = 0

失败的连接尝试 = 298

重置连接 = 31

当前连接 = 0

接收的分段 = 35028

发送的分段 = 12614

重新传输的分段 = 0

IPv4 的 UDP 统计信息

接收的数据报 = 14203

无端口 = 127

接收错误 = 162

发送的数据报 = 11135

IPv6 的 UDP 统计信息

接收的数据报 = 2709

无端口 = 0

接收错误 = 18

发送的数据报 = 1004

1. **netstat -e**



1. **netstat -an**

C:\Users\asus>netstat -an

活动连接

协议 本地地址 外部地址 状态

TCP 0.0.0.0:135 0.0.0.0:0 LISTENING

TCP 0.0.0.0:445 0.0.0.0:0 LISTENING

TCP 0.0.0.0:902 0.0.0.0:0 LISTENING

TCP 0.0.0.0:912 0.0.0.0:0 LISTENING

TCP 0.0.0.0:2343 0.0.0.0:0 LISTENING

TCP 0.0.0.0:3306 0.0.0.0:0 LISTENING

TCP 0.0.0.0:3580 0.0.0.0:0 LISTENING

TCP 0.0.0.0:3582 0.0.0.0:0 LISTENING

TCP 0.0.0.0:5040 0.0.0.0:0 LISTENING

TCP 0.0.0.0:7680 0.0.0.0:0 LISTENING

TCP 0.0.0.0:8080 0.0.0.0:0 LISTENING

TCP 0.0.0.0:9012 0.0.0.0:0 LISTENING

TCP 0.0.0.0:9013 0.0.0.0:0 LISTENING

TCP 0.0.0.0:10001 0.0.0.0:0 LISTENING

TCP 0.0.0.0:33060 0.0.0.0:0 LISTENING

TCP 0.0.0.0:49664 0.0.0.0:0 LISTENING

TCP 0.0.0.0:49665 0.0.0.0:0 LISTENING

TCP 0.0.0.0:49666 0.0.0.0:0 LISTENING

TCP 0.0.0.0:49667 0.0.0.0:0 LISTENING

TCP 0.0.0.0:49670 0.0.0.0:0 LISTENING

TCP 0.0.0.0:49744 0.0.0.0:0 LISTENING

TCP 0.0.0.0:59110 0.0.0.0:0 LISTENING

TCP 0.0.0.0:59111 0.0.0.0:0 LISTENING

TCP 10.208.163.215:139 0.0.0.0:0 LISTENING

TCP 10.208.163.215:7680 10.208.119.223:60913 TIME\_WAIT

TCP 10.208.163.215:7680 10.208.119.223:60921 TIME\_WAIT

TCP 10.208.163.215:7680 10.208.119.223:60935 TIME\_WAIT

TCP 10.208.163.215:49764 20.198.162.76:443 ESTABLISHED

TCP 10.208.163.215:49874 36.155.229.147:14000 ESTABLISHED

TCP 10.208.163.215:49889 121.51.36.101:443 CLOSE\_WAIT

TCP 10.208.163.215:49934 140.206.78.21:80 ESTABLISHED

TCP 10.208.163.215:49969 121.51.36.101:443 CLOSE\_WAIT

TCP 10.208.163.215:49976 140.206.78.9:80 ESTABLISHED

TCP 10.208.163.215:50221 61.155.222.119:80 CLOSE\_WAIT

TCP 10.208.163.215:50262 112.25.105.37:443 CLOSE\_WAIT

TCP 10.208.163.215:50372 61.155.222.119:80 CLOSE\_WAIT

TCP 10.208.163.215:50664 180.163.235.98:443 ESTABLISHED

TCP 10.208.163.215:50764 110.43.89.210:443 CLOSE\_WAIT

TCP 10.208.163.215:50765 110.43.89.210:443 CLOSE\_WAIT

TCP 10.208.163.215:50861 153.35.88.41:443 CLOSE\_WAIT

TCP 10.208.163.215:50862 153.35.88.41:443 CLOSE\_WAIT

TCP 10.208.163.215:50863 61.155.222.119:80 CLOSE\_WAIT

TCP 10.208.163.215:51186 112.64.200.247:80 ESTABLISHED

TCP 10.208.163.215:52110 64.233.189.188:5228 ESTABLISHED

TCP 10.208.163.215:53596 183.3.226.50:443 CLOSE\_WAIT

TCP 10.208.163.215:54770 203.208.43.70:443 TIME\_WAIT

TCP 10.208.163.215:54771 114.250.68.38:443 TIME\_WAIT

TCP 10.208.163.215:54773 114.250.64.38:443 TIME\_WAIT

TCP 10.208.163.215:54774 114.250.68.38:443 TIME\_WAIT

TCP 10.208.163.215:54776 114.250.64.38:443 TIME\_WAIT

TCP 10.208.163.215:54833 180.163.150.161:443 TIME\_WAIT

TCP 10.208.163.215:54933 112.80.248.75:80 LAST\_ACK

TCP 10.208.163.215:54961 172.217.24.10:443 ESTABLISHED

TCP 10.208.163.215:54963 20.44.229.112:443 TIME\_WAIT

TCP 10.208.163.215:54976 20.44.229.112:443 ESTABLISHED

TCP 10.208.163.215:54980 120.52.183.181:80 TIME\_WAIT

TCP 10.208.163.215:54981 112.80.248.76:80 ESTABLISHED

TCP 10.208.163.215:54995 112.80.248.75:443 FIN\_WAIT\_1

TCP 10.208.163.215:54996 112.80.248.76:80 ESTABLISHED

TCP 10.208.163.215:54999 140.206.164.199:8080 ESTABLISHED

TCP 10.208.163.215:55000 112.80.248.75:443 FIN\_WAIT\_1

TCP 127.0.0.1:1042 0.0.0.0:0 LISTENING

TCP 127.0.0.1:1043 0.0.0.0:0 LISTENING

TCP 127.0.0.1:4301 0.0.0.0:0 LISTENING

TCP 127.0.0.1:4709 0.0.0.0:0 LISTENING

TCP 127.0.0.1:9012 127.0.0.1:49758 ESTABLISHED

TCP 127.0.0.1:9487 0.0.0.0:0 LISTENING

TCP 127.0.0.1:9487 127.0.0.1:49751 ESTABLISHED

TCP 127.0.0.1:10000 0.0.0.0:0 LISTENING

TCP 127.0.0.1:13010 0.0.0.0:0 LISTENING

TCP 127.0.0.1:13030 0.0.0.0:0 LISTENING

TCP 127.0.0.1:13031 0.0.0.0:0 LISTENING

TCP 127.0.0.1:13032 0.0.0.0:0 LISTENING

TCP 127.0.0.1:17945 0.0.0.0:0 LISTENING

TCP 127.0.0.1:21440 0.0.0.0:0 LISTENING

TCP 127.0.0.1:21441 0.0.0.0:0 LISTENING

TCP 127.0.0.1:27018 0.0.0.0:0 LISTENING

TCP 127.0.0.1:49675 127.0.0.1:49676 ESTABLISHED

TCP 127.0.0.1:49676 127.0.0.1:49675 ESTABLISHED

TCP 127.0.0.1:49678 127.0.0.1:49679 ESTABLISHED

TCP 127.0.0.1:49679 127.0.0.1:49678 ESTABLISHED

TCP 127.0.0.1:49680 127.0.0.1:49681 ESTABLISHED

TCP 127.0.0.1:49681 127.0.0.1:49680 ESTABLISHED

TCP 127.0.0.1:49683 127.0.0.1:49684 ESTABLISHED

TCP 127.0.0.1:49684 127.0.0.1:49683 ESTABLISHED

TCP 127.0.0.1:49695 127.0.0.1:49696 ESTABLISHED

TCP 127.0.0.1:49696 127.0.0.1:49695 ESTABLISHED

TCP 127.0.0.1:49697 127.0.0.1:49698 ESTABLISHED

TCP 127.0.0.1:49698 127.0.0.1:49697 ESTABLISHED

TCP 127.0.0.1:49751 127.0.0.1:9487 ESTABLISHED

TCP 127.0.0.1:49758 127.0.0.1:9012 ESTABLISHED

TCP 127.0.0.1:49799 127.0.0.1:65001 ESTABLISHED

TCP 127.0.0.1:49803 0.0.0.0:0 LISTENING

TCP 127.0.0.1:49803 127.0.0.1:49821 ESTABLISHED

TCP 127.0.0.1:49821 127.0.0.1:49803 ESTABLISHED

TCP 127.0.0.1:54360 0.0.0.0:0 LISTENING

TCP 127.0.0.1:54530 0.0.0.0:0 LISTENING

TCP 127.0.0.1:54530 127.0.0.1:54597 ESTABLISHED

TCP 127.0.0.1:54597 127.0.0.1:54530 ESTABLISHED

TCP 127.0.0.1:54598 127.0.0.1:54599 ESTABLISHED

TCP 127.0.0.1:54599 127.0.0.1:54598 ESTABLISHED

TCP 127.0.0.1:65001 0.0.0.0:0 LISTENING

TCP 127.0.0.1:65001 127.0.0.1:49799 ESTABLISHED

TCP 192.168.150.1:139 0.0.0.0:0 LISTENING

TCP 192.168.159.1:139 0.0.0.0:0 LISTENING

TCP [::]:135 [::]:0 LISTENING

TCP [::]:445 [::]:0 LISTENING

TCP [::]:3306 [::]:0 LISTENING

TCP [::]:7680 [::]:0 LISTENING

TCP [::]:9012 [::]:0 LISTENING

TCP [::]:9013 [::]:0 LISTENING

TCP [::]:33060 [::]:0 LISTENING

TCP [::]:49664 [::]:0 LISTENING

TCP [::]:49665 [::]:0 LISTENING

TCP [::]:49666 [::]:0 LISTENING

TCP [::]:49667 [::]:0 LISTENING

TCP [::]:49670 [::]:0 LISTENING

TCP [::]:49744 [::]:0 LISTENING

TCP [::1]:50329 [::]:0 LISTENING

UDP 0.0.0.0:2343 \*:\*

UDP 0.0.0.0:3600 \*:\*

UDP 0.0.0.0:3601 \*:\*

UDP 0.0.0.0:4008 \*:\*

UDP 0.0.0.0:5000 \*:\*

UDP 0.0.0.0:5001 \*:\*

UDP 0.0.0.0:5002 \*:\*

UDP 0.0.0.0:5050 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5353 \*:\*

UDP 0.0.0.0:5355 \*:\*

UDP 0.0.0.0:6000 \*:\*

UDP 0.0.0.0:6001 \*:\*

UDP 0.0.0.0:6002 \*:\*

UDP 0.0.0.0:10001 \*:\*

UDP 0.0.0.0:52335 \*:\*

UDP 0.0.0.0:59993 \*:\*

UDP 0.0.0.0:60983 \*:\*

UDP 0.0.0.0:62393 \*:\*

UDP 0.0.0.0:64027 \*:\*

UDP 0.0.0.0:65144 \*:\*

UDP 10.208.163.215:137 \*:\*

UDP 10.208.163.215:138 \*:\*

UDP 10.208.163.215:1900 \*:\*

UDP 10.208.163.215:5353 \*:\*

UDP 10.208.163.215:51898 \*:\*

UDP 127.0.0.1:1900 \*:\*

UDP 127.0.0.1:10010 \*:\*

UDP 127.0.0.1:40000 \*:\*

UDP 127.0.0.1:51647 \*:\*

UDP 127.0.0.1:51648 \*:\*

UDP 127.0.0.1:51649 \*:\*

UDP 127.0.0.1:51650 \*:\*

UDP 127.0.0.1:51651 \*:\*

UDP 127.0.0.1:51652 \*:\*

UDP 127.0.0.1:51897 \*:\*

UDP 127.0.0.1:52377 \*:\*

UDP 127.0.0.1:52378 \*:\*

UDP 127.0.0.1:52379 \*:\*

UDP 127.0.0.1:52380 \*:\*

UDP 127.0.0.1:56189 \*:\*

UDP 127.0.0.1:59902 \*:\*

UDP 192.168.150.1:137 \*:\*

UDP 192.168.150.1:138 \*:\*

UDP 192.168.150.1:1900 \*:\*

UDP 192.168.150.1:5353 \*:\*

UDP 192.168.150.1:51895 \*:\*

UDP 192.168.159.1:137 \*:\*

UDP 192.168.159.1:138 \*:\*

UDP 192.168.159.1:1900 \*:\*

UDP 192.168.159.1:5353 \*:\*

UDP 192.168.159.1:51896 \*:\*

UDP [::]:5353 \*:\*

UDP [::]:5353 \*:\*

UDP [::]:5353 \*:\*

UDP [::]:5353 \*:\*

UDP [::]:5353 \*:\*

UDP [::]:5353 \*:\*

UDP [::]:5353 \*:\*

UDP [::]:5355 \*:\*

UDP [::]:64028 \*:\*

UDP [::1]:1900 \*:\*

UDP [::1]:5353 \*:\*

UDP [::1]:51893 \*:\*

UDP [fe80::3434:4928:3089:c390%16]:1900 \*:\*

UDP [fe80::3434:4928:3089:c390%16]:51891 \*:\*

UDP [fe80::3cc6:ba71:e69:b58c%10]:1900 \*:\*

UDP [fe80::3cc6:ba71:e69:b58c%10]:51894 \*:\*

UDP [fe80::b50f:5cba:5193:de4c%22]:1900 \*:\*

UDP [fe80::b50f:5cba:5193:de4c%22]:51892 \*:\*

**四、Nslookup**

MS-DOS utility that enables a user to look up an IP address of a domain or host on a network.

**Tasks:**

1、Try nslookup on

2、The servers you get by ipconfig /all

[www.seu.edu.cn](http://www.seu.edu.cn)

[www.google.com](http://www.google.com)

3、Explain what you find out (how a domain name is correspondent to multiple IPs)?

先做个简单的介绍吧，nslookup (全称 name server lookup) ，是一个在命令行界面下的网络工具，它有两种模式:交互 & 非交互，进入交互模式在命令行界面直接输入nslookup按回车，非交互模式则是后面跟上查询的域名或者IP地址按回车。一般来说，非交互模式适用于简单的单次查询，若需要多次查询，则交互模式更加适合，例如计网第七版第二章的课后习题p19 (单纯举例子，没有书不影响接下来的阅读)，从根服务器进行迭代查询。

**DNS记录类型介绍：**

1、DNS：完成主机名（域名）到IP的映射

2、A记录：用来指定主机（或域名）对应的IP地址记录。

3、NS记录：域名服务器记录，用来指定该域名由那个DNS服务器来解析。

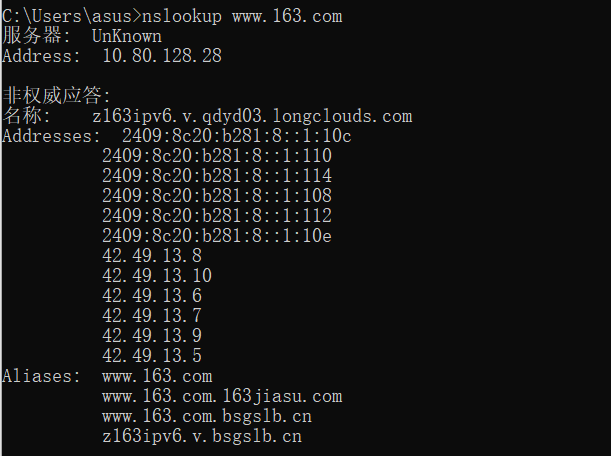
4、CNAME记录：别名记录，允许将多个名字映射到同一台计算机。

5、MX记录：邮件交换记录，用于电子邮件系统发邮件时根据收信人的地址后缀来定位邮件服务器。

6、PTR记录：反向记录。

7、TXT记录：一般指某个主机名或域名的说明。

8、TTL记录：告诉网络路由器包在网络中的时间是否太长而应被丢弃。



**发现：**

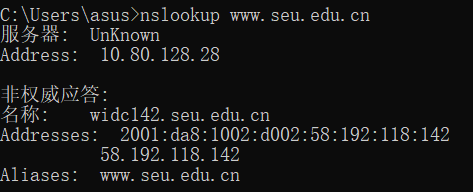
1）68.128.128.61.cq.cq.cta.net.cn中的68.128.128.61 与 61.128.128.68顺序是相反的。因为域名的顺序是自底向上（从子域到域）而IP地址的顺序是自顶向下（网络到主机）

2）非权威回答：假设DNS服务器中没有存储域名test.com的相关信息，则当客户端请求获得域名test.com的相关信息时DNS服务器就会通过迭代递归的方式从test公司的DNS服务器中获取test.com的域名信息返回给客户端，同时会把test.com的相关信息存储在自己缓冲中一段时间，当下次客户端再次请求test.com信息时就从自身缓冲中提取信息返回给客户端，这个回答叫“非权威回答”。

3）Addresses有很多记录，说明163.com域名实际对呀多个物理主机，当用户访问www.162.com时按照一定的规则由其中一台计算机进行应答。

4）Aliases:表示www.163.com www.163.com.lxdns.com只是DNS记录中的一个别名方便我们记忆





**一个域名如何对应多个IP？**

1. 一个域名设定多个dns服务或者服务器进行解析，同一个域名的每个解析都指向不同的ip地址，这样应答快的dns优先进行解析，这样就能保证最快定向到指定的网站空间去。如果空间也存在不同地点的相同内容镜像，那么这种方式最适合，例如sohu全国都有服务器，就是这么干的，除非你指定要访问某个地方的服务器，否则默认转向最快的空间，因为内容是镜像自动同步的，内容上不存在差别，用户是不知道的。一般域名管理机构只允许设定2到3个域名解析服务，所以再多就不支持了。  
     
   2、自己公司建立服务器，给自己进行dns解析服务，这样容易实现以上功能的操做。且数量不受限制，例如给客户提供移动、联通、网通、电信、东方宽带、长城宽带等等多个dns，那么使用对应宽带的用户都能自动定向到适合自己的网站去，企业可以分别在不同运营商那里建立网站，保证不同带宽的用户都能高速访问，很多游戏运营商都是这么干的。DNS负载均衡技术是在DNS服务器中为同一个主机名配置多个IP地址，在应答DNS查询时，DNS服务器对每个查询将以DNS文件中主机记录的IP地址按顺序返回不同的解析结果，将客户端的访问引导到不同的机器上去，使得不同的客户端访问不同的服务器，从而达到负载均衡的目的。

**Lab-Unit3**

**Targets**

1.Capture data packets using network analysing software

2.Protocol analysis of the following protocols

ARP

ICMP (Ping)

DHCP

DNS

UDP

TCP

HTTP

**Try to answer the following questions**

1.How to generate correspondent data packets?

2.How to capture those packets?

3.Packet analysis

The packet format-Fields and their meanings

4.Protocol analysis

Work flows of different protocols

**ARP**

How to generate ARP request?

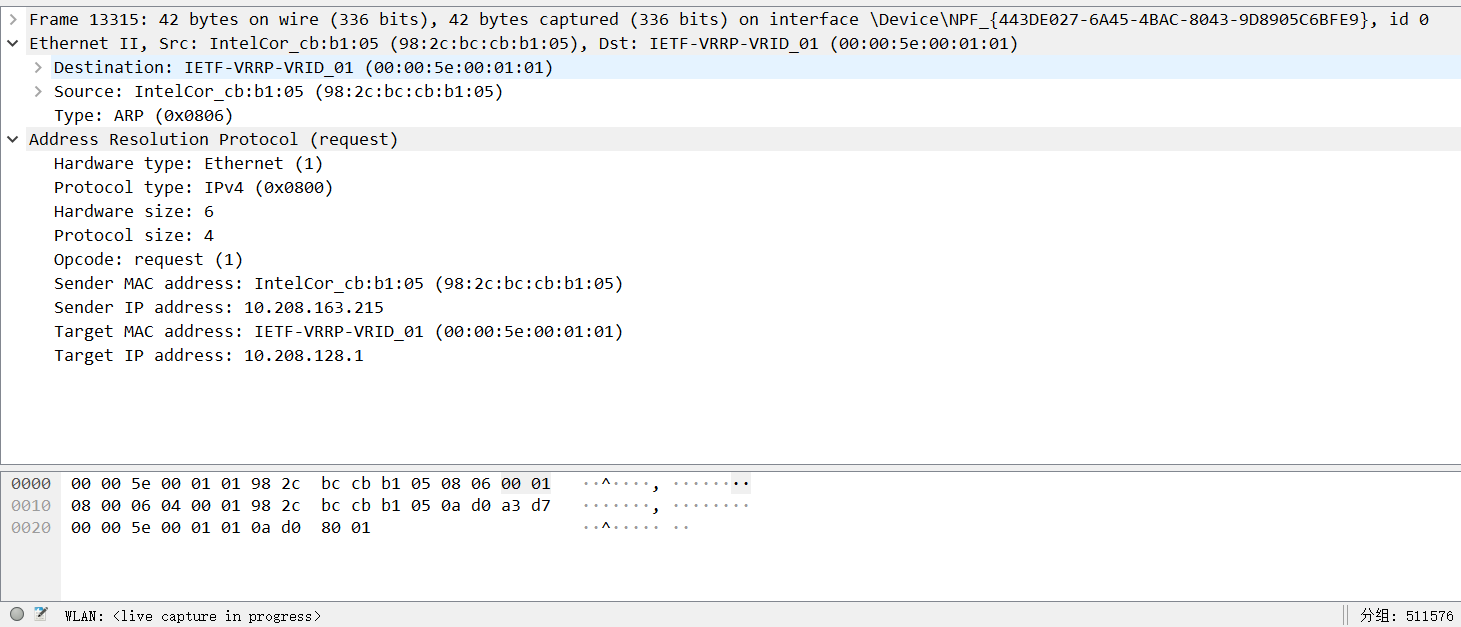
Arp –d

Ping IP\_address (LAN)

1. **ARP简介**

地址解析协议（Address Resolution Protocol），其基本功能为透过目标设备的IP地址，查询目标设备的MAC地址，以保证通信的顺利进行。它是IPv4中网络层必不可少的协议，不过在IPv6中已不再适用，并被邻居发现协议（NDP）所替代。主机发送信息时将包含⽬标IP地址的ARP请求⼴播到局域⽹络上的所有主机，并接收返回消息，以此确定⽬标的物理地址；收到返回消息后将该IP地址和物理地址存⼊本机ARP缓存中并保留⼀定时间，下次请求时直接查询ARP缓存以节约资源。

1. **分析包**



00 00 5e 00 01 01 Destination

98 2c bc cd b1 05 Source

08 06 Type:ARP

00 01 Hardware type

08 00 Protocol type

06 04 Size

98 2c bc cd b1 05 Sender MAC address

0a d0 a3 d7 Sender IP

00 00 5e 00 01 01 Target MAC address

0a d0 80 01 Target IP

1. **ARP工作流程**

假设主机A和B在同一个网段，主机A要向主机B发送信息，具体的地址解析过程如下：

(1) 主机A首先查看自己的ARP表，确定其中是否包含有主机B对应的ARP表项。如果找到了对应的MAC地址，则主机A直接利用ARP表中的MAC地址，对IP数据包进行帧封装，并将数据包发送给主机B。

(2) 如果主机A在ARP表中找不到对应的MAC地址，则将缓存该数据报文，然后以广播方式发送一个ARP请求报文。ARP请求报文中的发送端IP地址和发送端MAC地址为主机A的IP地址和MAC地址，目标IP地址和目标MAC地址为主机B的IP地址和全0的MAC地址。由于ARP请求报文以广播方式发送，该网段上的所有主机都可以接收到该请求，但只有被请求的主机（即主机B）会对该请求进行处理。

(3) 主机B比较自己的IP地址和ARP请求报文中的目标IP地址，当两者相同时进行如下处理：将ARP请求报文中的发送端（即主机A）的IP地址和MAC地址存入自己的ARP表中。之后以单播方式发送ARP响应报文给主机A，其中包含了自己的MAC地址。

(4) 主机A收到ARP响应报文后，将主机B的MAC地址加入到自己的ARP表中以用于后续报文的转发，同时将IP数据包进行封装后发送出去

**ICMP**

PING

1. **ICMP简介**

ICMP协议是一种面向无连接的协议，用于传输出错报告控制信息。它是一个非常重要的协议，它对于网络安全具有极其重要的意义。它属于网络层协议，主要用于在主机与路由器之间传递控制信息，包括报告错误、交换受限控制和状态信息等。当遇到IP数据无法访问目标、IP路由器无法按当前的传输速率转发数据包等情况时，会自动发送ICMP消息。

ICMP 是 TCP/IP 模型中网络层的重要成员，与 IP 协议、ARP 协议、RARP 协议及 IGMP 协议共同构成 TCP/IP 模型中的网络层。ping 和 tracert是两个常用网络管理命令，ping 用来测试网络可达性，tracert 用来显示到达目的主机的路径。ping和 tracert 都利用 ICMP 协议来实现网络功能，它们是把网络协议应用到日常网络管理的典型实例。

从技术角度来说，ICMP就是一个“错误侦测与回报机制”，其目的就是让我们能够检测网路的连线状况﹐也能确保连线的准确性。当路由器在处理一个数据包的过程中发生了意外，可以通过ICMP向数据包的源端报告有关事件。

其功能主要有：侦测远端主机是否存在，建立及维护路由资料，重导资料传送路径（ICMP重定向），资料流量控制。ICMP在沟通之中，主要是透过不同的类别(Type)与代码(Code) 让机器来识别不同的连线状况。

ICMP 是个非常有用的协议﹐尤其是当我们要对网路连接状况进行判断的时候。

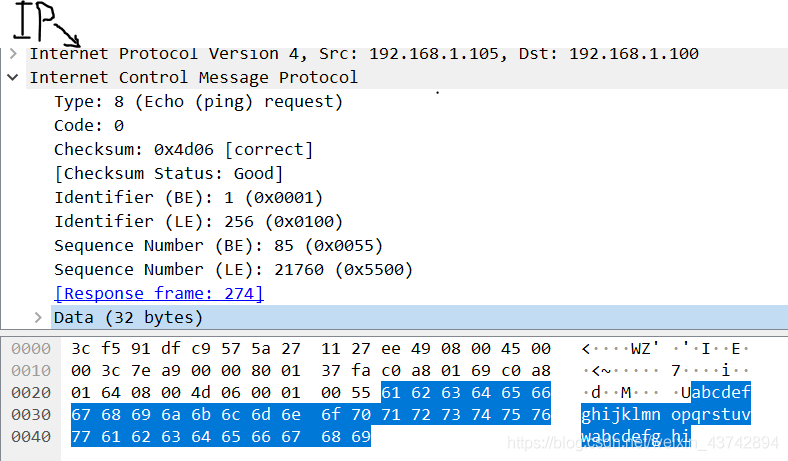
1. **ICMP工作原理**

ICMP提供一致易懂的出错报告信息。发送的出错报文返回到发送原数据的设备，因为只有发送设备才是出错报文的逻辑接受者。发送设备随后可根据ICMP报文确定发生错误的类型，并确定如何才能更好地重发失败的数据包。但是ICMP唯一的功能是报告问题而不是纠正错误，纠正错误的任务由发送方完成。

我们在网络中经常会使用到ICMP协议，比如我们经常使用的用于检查网络通不通的Ping命令（Linux和Windows中均有），这个“Ping”的过程实际上就是ICMP协议工作的过程。还有其他的网络命令如跟踪路由的Tracert命令也是基于ICMP协议的。

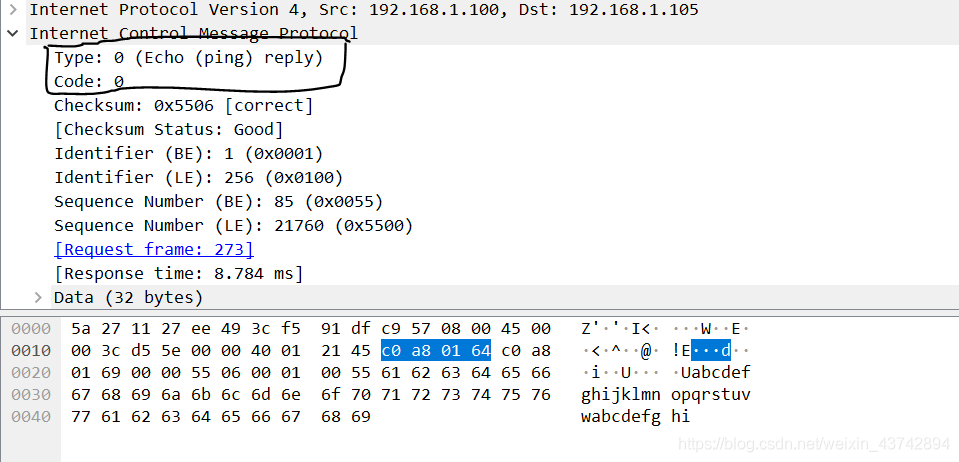
1. **分析包**

看一下第一个数据包，它显示了主机192.168.1.105在给192.168.1.100发送数据包。ICMP头部中的Type的值是8，Code值是0，说明这是一个echo请求数据包，Wireshark已经帮我们解析出来了。其实这是一个简单的ICMP数据包，使用IP发送，所包含的数据很少。除了指定的类型、代码以及校验和，这里还有序列号用于匹配请求和响应，并且在可变域中包含有一串随机字符串。



并且Response frame显示274，意思是request的reply包是274.

接下来我们再分析一下第二个数据包：



这个数据包是对我们请求的响应。在它的ICMP头部中，类型和代码的值都是0，表示这是一个echo响应。由于第二个数据包的序列号和第一个数据包可以相匹配，于是就可以确定它和第一个数据包是对应的。并且还可以发现，在Data的部分，这个数据包有着和第一个数据包相同的字符内容。当这个数据包被成功接收后，ping就成功了。

1. **应用**

ICMP 协议应用在许多网络管理命令中，下面以 ping 和 tracert 命令为例详细介绍 ICMP 协议的应用。

（1） ping 命令使用 ICMP 回送请求和应答报文

在网络可达性测试中使用的分组网间探测命令 ping 能产生 ICMP 回送请求和应答报文。目的主机收到 ICMP 回送请求报文后立刻回送应答报文，若源主机能收到 ICMP 回送应答报文，则说明到达该主机的网络正常。

（2）路由分析诊断程序 tracert 使用了 ICMP时间超过报文

tracert 命令主要用来显示数据包到达目的主机所经过的路径。通过执行一个 tracert 到对方主机的命令，返回数据包到达目的主机所经历的路径详细信息，并显示每个路径所消耗的时间。

**DHCP**

1.How to generate DHCP packets?

2.Packet structure

3.Find out what kind of information is transmitted by DHCP protocols

**1.DHCP简介**

DHCP(Dynamic Host Configuration Protocol),动态主机配置协议，是一个应用层协议。当我们将客户主机ip地址设置为动态获取方式时，DHCP服务器就会根据DHCP协议给客户端分配IP，使得客户机能够利用这个IP上网。

DHCP的前身是BOOTP协议（Bootstrap Protocol）,BOOTP被创建出来为连接到网络中的设备自动分配地址，后来被DHCP取代了，DHCP比BOOTP更加复杂，功能更强大。后面可以看到，在用Wireshark过滤显示DHCP包，需要输入过滤条件BOOTP，而不是DHCP,但或许是因为我使用的Wireshark版本是比较旧的1.12.9,没有在新版本中尝试过，也许可以输入DHCP让其只显示DHCP包。

1. **DHCP的实现**

DHCP的实现分为4步，分别是：

第一步：Client端在局域网内发起一个DHCP　Discover包，目的是想发现能够给它提供IP的DHCP Server。

第二步：可用的DHCP Server接收到Discover包之后，通过发送DHCP Offer包给予Client端应答，意在告诉Client端它可以提供IP地址。

第三步：Client端接收到Offer包之后，发送DHCP Request包请求分配IP。

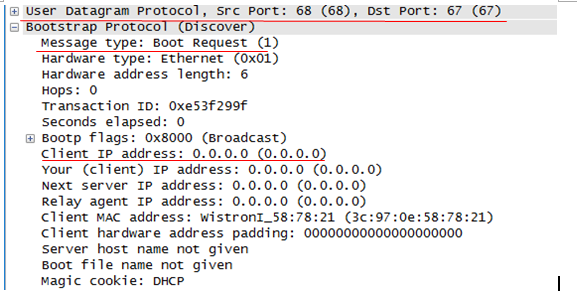
第四步：DHCP Server发送ACK数据包，确认信息。

**3.包分析**

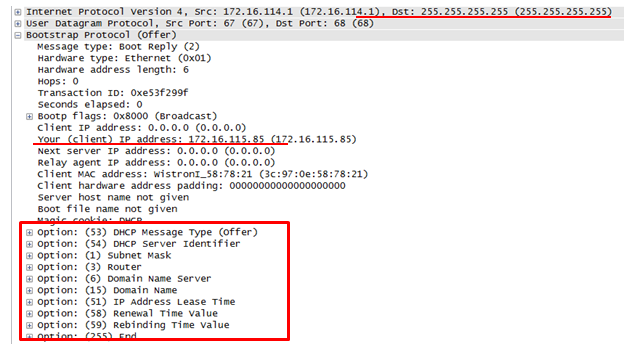
1，DHCP Discover数据包

(1)Client端使用IP地址0.0.0.0发送了一个广播包，可以看到此时的目的IP为255.255.255.255。Client想通过这个数据包发现可以给它提供服务的DHCP服务器。

1. 从下图可以看出，DHCP属于应用层协议，它在传输层使用UDP协议，目的端口是67。



2，DHCP Offer包  
当DHCP服务器收到一条DHCP Discover数据包时，用一个DHCP Offerr包给予客户端响应。



(1)DHCP服务器仍然使用广播地址作为目的地址，因为此时请求分配IP的Client并没有自己ip,而可能有多个Client在使用0.0.0.0这个IP作为源IP向DHCP服务器发出IP分配请求，DHCP也不能使用0.0.0.0这个IP作为目的IP地址，于是依然采用广播的方式，告诉正在请求的Client们，这是一台可以使用的DHCP服务器。

(2)DHCP服务器提供了一个可用的IP,在数据包的Your (client) IP Address字段可以看到DHCP服务器提供的可用IP。

(3)除此之外，如图中红色矩形框的内容所示，服务器还发送了子网掩码，路由器，DNS，域名，IP地址租用期等信息。

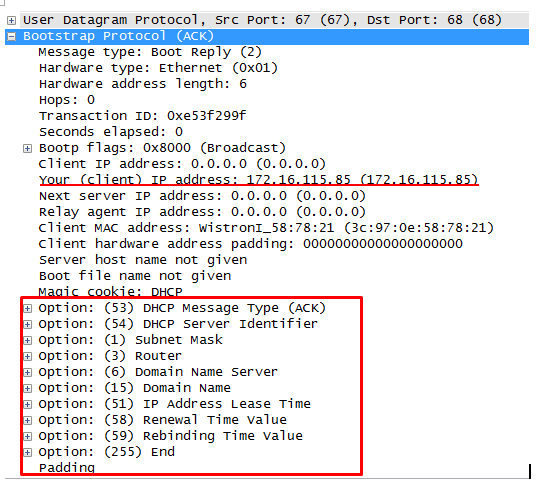
3，DHCP Request包

当Client收到了DHCP Offer包以后（如果有多个可用的DHCP服务器，那么可能会收到多个DHCP Offer包），确认有可以和它交互的DHCP服务器存在，于是Client发送Request数据包，请求分配IP。

此时的源IP和目的IP依然是0.0.0.0和255.255.255.255。

4，DHCP ACK包

服务器用DHCP ACK包对DHCP请求进行响应。



在数据包中包含以下信息，表示将这些资源信息分配给Client.

Your(client) IP address:分配给Client的可用IP。

后面有许多项option信息，前两项是DHCP服务器发送的消息类型（ACK）和服务器的身份标识，后面几项是：

Subnet Mask:Client端分配到的IP的子网掩码；

Router:路由器

Domain Name Server:DNS,域名服务器

Domain Name:域名

IP Address Lease Time:IP租用期。

**DNS**

1.When DNS is involved?

2.Packet structure

3.Find out the transmitted information

**1.When DNS is involved?**

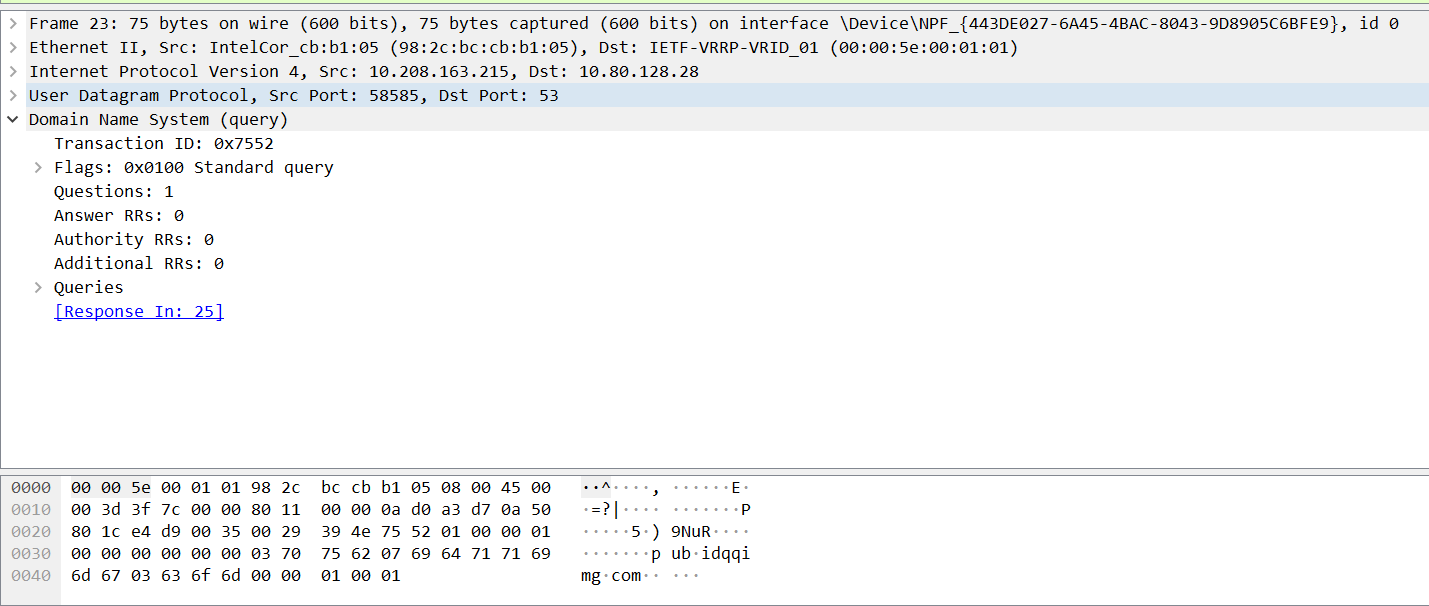
 DNS 的全称是 Domain Name System（域名系统，或者常称为域名解析系统）。

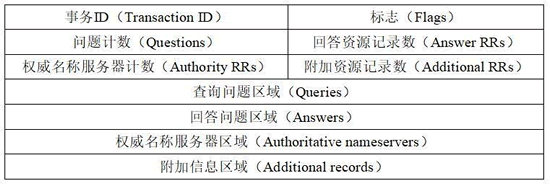
我们在上网的时候，经常要通过浏览器去访问一些网站，实际上就是访问网站所在的服务器IP地址，但是让用户记住设备的IP地址是非常困难的。我们常用的方式就是在浏览器中输入要访问的网址链接（例如 www.example.com)，然后浏览器就会通过解析这个域名地址，找到要访问的服务器，从而读取到需要访问的网站的数据。所以我们在因特网上，构建了域名和IP地址互相映射的分布式数据库，在IP地址和域名之间形成一种转换和查询机制。

        所以，DNS工作的过程可以简单描述为：域名地址经过DNS服务器解析后，得到对应的IP地址，通过该IP地址访问到服务器获取我们要访问的内容。

        DNS通信主要使用UDP，TCP为辅，使用端口号53。当然，端口号可以修改，但是默认是使用53端口；在某些情况下，也会采用TCP协议。DNS是网络分层里的应用层协议，事实上他是为其他应用层协议工作的。

**2.Packet structure**





上图中显示了 DNS 的报文格式。其中，事务 ID、标志、问题计数、回答资源记录数、权威名称服务器计数、附加资源记录数这 6 个字段是DNS的报文首部，共 12 个字节。  
整个 DNS 格式主要分为 3 部分内容，即基础结构部分、问题部分、资源记录部分。

**事务 ID**：DNS 报文的 ID 标识。对于请求报文和其对应的应答报文，该字段的值是相同的。通过它可以区分 DNS 应答报文是对哪个请求进行响应的。

**标志**：DNS 报文中的标志字段。

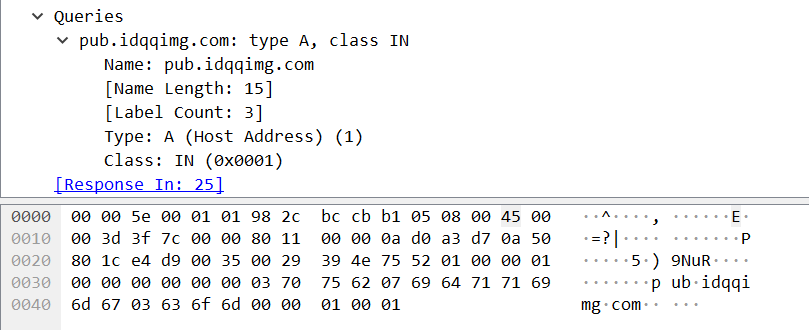
**问题计数**：DNS 查询请求的数目。

**回答资源记录数**：DNS 响应的数目。

**权威名称服务器计数**：权威名称服务器的数目。

**附加资源记录数**：额外的记录数目（权威名称服务器对应 IP 地址的数目）。

**3.Find out the transmitted information**



**UDP, TCP packets**

1. **How the TCP connection is established? (the first 3 packets)**

TCP是面向连接的协议。运输连接是用来传送tcp报文的，tcp运输连接的建立和释放时每一次面向连接的通信中必不可少的过程。因此，运输连接就有三个阶段，即：连接建立。数据传送，和连接释放。

tcp建立连接的过程的过程叫握手，握手需要在客户和服务器之间交换三个报文段

假定主机A运行的是TCP客户程序，而B运行TCP服务器程序，最初两端的tcp进程都处于close状态，A主动打开连接，B被动打开连接。

一开始，B的tcp服务器进程先创建传输控制块TCB（存储每一个连接中的重要信息，tcp连接表，指向发送和接收缓存的指针，指向重传队列的指针，当前的发送和接收序号等等），准备接受客户进程的连接请求，然后服务器进程就处于LISTEN（收听）状态，等待客户的连接请求，如有，即做出响应.

A的tcp客户进程也是首先创建传输控制模块TCB，在打算建立tcp连接时，客户端向服务端发出连接请求报文段，这时，首部中的同步位SYN=1，同时选择一个初始序号seq=x。tcp规定SYN报文段（SYN=1的报文段）不能携带数据，但是要消耗一个序号。完成后客户端进入SYN-SENT（同步发送）状态

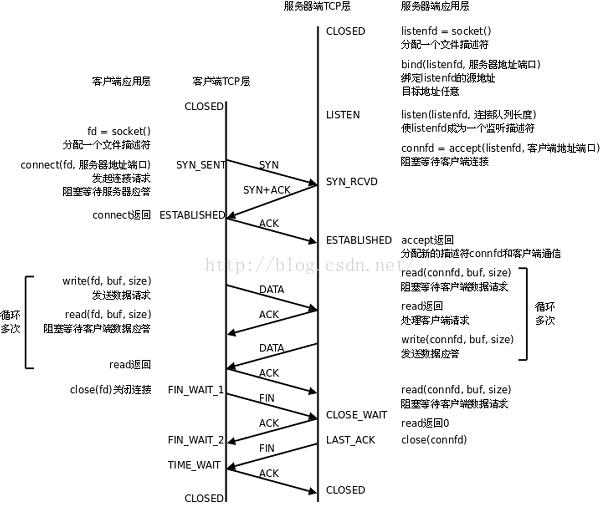
服务端收到连接请求报文段后，如果同意建立连接，则向客户端发送确认，在确认报文段中把SYN和ACK位都置1，确认号位ack=x+1，同时为自己选择一个初始序号seq=y。注意这个报文段也不能携带数据，但同样要消耗掉一个序号。这时服务端进程进入SYN-RCVD（同步收到）状态

客户端收到服务端确认后，还要向服务端给出确认，确认报文段ACK置1，确认号ack=y+1，而自己的序号seq=x+1,tcp标准规定ACK报文段可以携带数据。但如果不携带数据则不消耗序号，在这种情况下，下一个数据报文端的序号仍是seq=x+1

这时tcp连接已经建立，客户端进入established（已建立连接）状态

当服务端收到确认后，也进入established状态

以上连接建立过程叫做三报文握手.



**2.Packets generated by**

HTTP

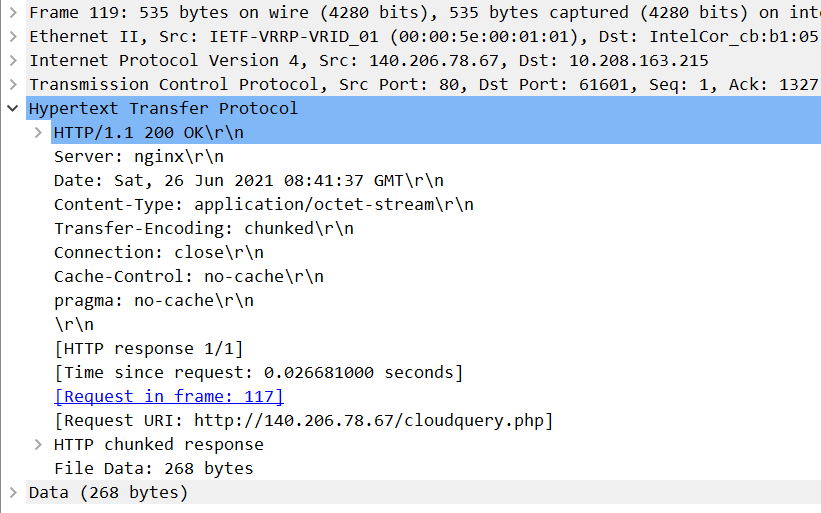
超文本传输协议（HTTP）是一个用于传输超媒体文档（例如 HTML）的应用层协议。它是为 Web 浏览器与 Web 服务器之间的通信而设计的，但也可以用于其他目的。HTTP 遵循经典的客户-服务模型，客户端打开一个连接以发出请求，然后等待它收到服务器端响应。HTTP 无状态协议，这意味着服务器不会在两个请求之间保留任何数据（状态）。该协议虽然通常基于 TCP/IP 层，但可以在任何可靠的传输层上使用；也就是说，不像 UDP，它是一个不会静默丢失消息的协议。RUDP——作为 UDP 的可靠化升级版本——是一种合适的替代选择。

1. **How to obtain a webpage?**

采⽤了请求/响应模型。客户端向服务器发送⼀个请求报⽂，请求报⽂包含请求的⽅法、URL、协议版 本、请求头部和请求数据。服务器以⼀个状态⾏作为响应，响应的内容包括协议的版本、成功或者错误 .

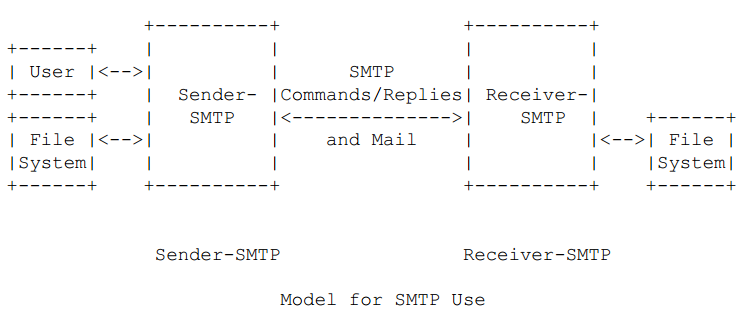
1. **The content of a webpage is obtained in a batch or piece-by-piece?**

是批量获取的。



SMTP

SMTP（Simple Mail Transfer Protocol）即简单邮件传输协议,它是一组用于由源地址到目的地址传送邮件的规则，由它来控制信件的中转方式。SMTP协议属于TCP/IP协议簇，它帮助每台计算机在发送或中转信件时找到下一个目的地。通过SMTP协议所指定的服务器,就可以把E-mail寄到收信人的服务器上了，整个过程只要几分钟。SMTP服务器则是遵循SMTP协议的发送邮件服务器，用来发送或中转发出的电子邮件。SMTP是一种TCP协议支持的提供可靠且有效电子邮件传输的应用层协议。



**What’s the difference between an user-agent based service and a web-mail service?**

user-agent service多为用户通过浏览器向web服务器请求资源，将资源下拉到本地，多为文件接收方来发起请求﹔而web-mail service由发送方来发起请求，从而将邮件从发送方邮件服务器推到接收邮件服务器中。