**黑龙江大学**

**实 验 报 告**

|  |  |  |  |
| --- | --- | --- | --- |
| **课程名称** | **计算机网络** | | |
| **实验项目名称** | **Socket通信编程** | | |
| **实验时间**  **（日期及节次）** | **2016/5/9** | | |
| **专业** | **计算机科学与技术** | **学生所在学院** | **计算机科学技术学院** |
| **年级** | **2013级** | **学号** | **20133104** |
| **姓名** | **郭耀文** | **指导教师** | **姜誉** |
| **实验室名称** | **4#513** | | |
| **实验成绩** |  | | |
| **教师签字** |  | | |

**黑龙江大学教务处**

|  |
| --- |
| 实验目的 掌握常用的网络指令，对网络进行管理。 |
| 1. **实验内容**   实验平台：Ubuntu15.10  内容：  arp、ipconfig、net、ping、tracert、netstat、nslookup、hostname、route |
| 1. **实验步骤及实验结果**   根据上通信的过程画出流程图  无标题 |
| 四，运行效果  arp用来管理系统的arp缓存，显示和修改地址解析协议使用的”ip到mac”地址转换表。  常用的命令参数：  arp:显示所有的表项    如果表项为空的话，需要先对某台主机建立一个连接，例如ping一下百度的  ping -c 4 [www.baidu.com](http://www.baidu.com)  但是再想添加新的目的arp表项时，ping其他网站的地址并不会有新的表项被添加的arp表中。所以只有ping一个局域网中的ip地址。例如：ping -c 4 192.168.1.105    arp -v  显示详细的ａｒｐ条目，跟省略参数一样  arp -a //通过询问当前数据，显示ip和mac的对应关系。如果指定ip地址，则只显示指定ip的arp表项。    arp -n //显示ip和mac的对应关系  arp -d ip\_addr //删除指定主机ip和Mac的对应关系  路由Ｒ4的路由表  arp -s hostname hw\_addr   //手动设置IP和MAC的对应关系    ifconfig 查看本机的网络配置，如ip地址，子网掩码，网关协议  路由Ｒ4的路由表  一般来说，直接输入 ifconfig 就会列出目前已经被启动的卡，不论这个卡是否有给予 IP，都会被显示出来。  UP:启动的模式，UNICAST是单播，BROADCAST广播，MULTICAST组播,  RUNNING:运行的模式，  如：UP BROADCAST RUNNING MULTICAST表示启动了广播和组播，正在使用组播，单播默认就支持，所以不会显示。  eth0网卡的在本系统中的名称代号  Link encap: 这个字段代表该网络设备位于OSI物理层的名称。  HWaddr：网卡的硬件地址，也就是MAC地址。  inet addr：显示网卡的IP地址  Bcast：该接口的网络广播地址 路由Ｒ4的路由表  Mask：该接口的网络子网掩码。  inet6 addr: IPv6地址  MTU：该接口的最大传输单元<字节>。以太网帧中的数据长度规定最小46字节，最大1500字节，ARP和RARP数据包的长度不够46字节，要在后面补填充位。最大值1500称为以太网的最大传输单元（MTU），不同的网络类型有不同的MTU，如果一个数据包从以太网路由到拨号链路上，数据包长度大于拨号链路的MTU了，则需要对数据包进行分 片（fragmentation）。注意，MTU这个概念指数据帧中有效载荷的最大长度，不包括帧首部的长度。  Metric：默认的路由成本。  RX ：那一行代表的是网络由启动到目前为止的封包接收情况， packets 代表封包数、errors 代表封包发生错误的数量、 dropped 代表封包由于有问题而遭丢弃的数量、overruns缓冲区溢出丢弃、frame: 表示 misaligned的frames。  TX：这一行代表的是网络由启动到目前为止的已传送或者传送中的封包统计 ，carrier?；  collisions：代表封包碰撞的情况，如果发生太多次， 表示你的网络状况不太好；  txqueuelen：代表用来传输数据的缓冲区的储存长度；  RX bytes：已接收的封包数据量，单位是字节。  TX bytes：已传送的封包数据量，单位是字节  interrupt:这张网卡接口的IRQ中断值。  Scope:Host/Link?    ifconfig eth0 192.168.100.100 netmask 255.255.255.128 mtu 8000  # 设置网络接口，同时设置 MTU 的数值    sudo ifconfig eth0 mtu 1500  # 仅修改该接口的 MTU 数值，其他的保持不动    # ifconfig eth0:0 192.168.50.50  # 仔细看那个接口， eth0:0 。那就是在该网络接口上，再仿真一个网络接口，  # 亦即是在一个网卡上面设置多个 IP 的意思啦  Linux下的net命令，不是一个网络命令，而是samba服务器管理员的命令，所以在这里不再详细叙述。  ping 主要透过 [ICMP 封包](http://vbird.dic.ksu.edu.tw/linux_server/0140networkcommand/0110network_basic.php" \l "protocol_icmp) 来进行整个网络的状况报告，当然啦，最重要的就是那个 ICMP type 0, 8 这两个类型， 分别是要求回报与主动回报网络状态是否存在的特性。要特别注意的是， ping 还是需要透过 [IP 封包](http://vbird.dic.ksu.edu.tw/linux_server/0140networkcommand/0110network_basic.php" \l "ipandmac_head)来传送 ICMP 封包的， 而 IP 封包里面有个相当重要的 TTL (Time To Live) 属性，这是很重要的一个路由特性， 详细的 IP与ICMP表头数据请参考[网络基础](http://vbird.dic.ksu.edu.tw/linux_server/0140networkcommand/0110network_basic.php)的详细介绍。  -d：使用Socket的SO\_DEBUG功能；  -c<完成次数>：设置完成要求回应的次数；  -f：极限检测；  -i<间隔秒数>：指定收发信息的间隔时间；  -I<网络界面>：使用指定的网络界面送出数据包；  -l<前置载入>：设置在送出要求信息之前，先行发出的数据包；  -n：只输出数值；  -p<范本样式>：设置填满数据包的范本样式；  -q：不显示指令执行过程，开头和结尾的相关信息除外；  -r：忽略普通的Routing Table，直接将数据包送到远端主机上；  -R：记录路由过程；  -s<数据包大小>：设置数据包的大小；  -t<存活数值>：设置存活数值TTL的大小；  -v：详细显示指令的执行过程。     * 64 bytes：表示这次传送的 ICMP 封包大小为 64 bytes 这么大，这是默认值， 在某些特殊场合中，例如要搜索整个网络内最大的 MTU 时，可以使用 -s 2000 之类的数值来取代； * icmp\_seq=1：ICMP 所侦测进行的次数，第一次编号为 1 ； * ttl=125：TTL 与 IP 封包内的 TTL 是相同的，每经过一个带有 MAC 的节点 (node) 时，例如 router, bridge 时， TTL 就会减少一，默认的 TTL 为 255 ， 你可以透过 -t 150 之类的方法来重新配置默认 TTL 数值； * time=149ms：响应时间，单位有 ms(0.001秒)及 us(0.000001秒)， 一般来说，越小的响应时间，表示两部主机之间的网络联机越良好！   Tracert在linux下的命令是tracerroute  Traceroute (选项) (参数) -n ：可以不必进行主机的名称解析，单纯用 IP ，速度较快！  -w ：若对方主机在几秒钟内没有回声就宣告不治...默认是 5 秒  -i ：用在比较复杂的环境，如果你的网络接口很多很复杂时，才会用到这个参数；  举例来说，你有两条 ADSL 可以连接到外部，那你的主机会有路由Ｒ4的路由表  两个 ppp，  你可以使用 -i 来选择是 ppp0 还是 ppp1 啦！  -g ：与 -i 的参数相仿，只是 -g 后面接的是 gateway 的 IP 就是了。  好吧，我是败给这个神奇的指令了  理想中时这样的：  现实中却是这样的：    完全没有显示路径，有待进一步研究，我只有如实的记录下实验结果。  netstat  netstat -[rn] <==与路由有关的参数  netstat -[antulpc] <==与网络接口有关的参数参数：  与路由 (route) 有关的参数说明：  -r ：列出路由表(route table)，功能如同 route 这个命令；  -n ：不使用主机名与服务名称，使用 IP 与 port number ，如同 route -n  与网络接口有关的参数：  -a ：列出所有的联机状态，包括 tcp/udp/unix socket 等；  -t ：仅列出 TCP 封包的联机；  -u ：仅列出 UDP 封包的联机；  -l ：仅列出有在 Listen (监听) 的服务之网络状态；  -p ：列出 PID 与 Program 的檔名；  -c ：可以配置几秒钟后自动升级一次，例如 -c 5 每五秒升级一次网络状态的显示；  列出目前的路由表状态，且以 IP 及 port number 显示    秀出目前已经启动的网络服务    观察本机上头所有的网络联机状态    nslookup  nslookup命令是常用域名查询工具，就是查DNS信息用的命令。  nslookup4有两种工作模式，即“交互模式”和“非交互模式”。在“交互模式”下，用户可以向域名服务器查询各类主机、域名的信息，或者输出域名中 的主机列表。而在“非交互模式”下，用户可以针对一个主机或域名仅仅获取特定的名称或所需信息。 进入交互模式，直接输入nslookup命令，不加任何参数，则直接进入交互模式，此时nslookup会连接到默认的域名服务器（即/etc /resolv.conf的第一个dns地址）。或者输入nslookup -nameserver/ip。进入非交互模式，就直接输入nslookup 域名就可以了。    hostname  [Linux](http://www.chinabyte.com/keyword/Linux/" \t "/home/bool/Documents\\x/_blank)[操作系统](http://soft.chinabyte.com/os/" \t "/home/bool/Documents\\x/_blank)的hostname是一个kernel变量，可以通过hostname命令来查看本机的hostname。也可以直接cat /proc/sys/kernel/hostname查看。  不带参数时显示主机名。  格式：hostname  带上-i参数时显示ip地址。  格式：hostname -i  带上-f参数时显示全域名（FQDN，Fully Qualified Domain Name）。  格式：hostname -f  修改主机名时，后面跟上新的主机名即可。（注意此方法立即生效，但不会永久保存，详见后面的示例）。  格式：hostname newhostname    route  两部主机之间一定要有路由才能够互通 TCP/IP 的协议，否则就无法进行联机。 一般来说，只要有网络接口，该接口就会产生一个路由。  route [-nee]  route add [-net|-host] [网域或主机] netmask [mask] [gw|dev]  route del [-net|-host] [网域或主机] netmask [mask] [gw|dev]  -n ：不要使用通讯协议或主机名，直接使用 IP 或 port number；  -ee ：使用更详细的信息来显示  添加 (add) 与删除 (del) 路由的相关参数：  -net ：表示后面接的路由为一个网域；  -host ：表示后面接的为连接到单部主机的路由；  netmask ：与网域有关，可以配置 netmask 决定网域的大小；  gw ：gateway 的简写，后续接的是 IP 的数值喔，与 dev 不同；  dev ：如果只是要指定由那一块网络卡联机出去，则使用这个配置，后面接 eth0 等  单纯的观察路由状态    路由的添加与删除  Sudo route del -net 169.254.0.0 netmask 255.255.0.0 dev wlan0    添加路由器  sudo route add -net 169.254.0.0 netmask 255.255.0.0 dev wlan0  设置默认路由，注意只设置一个就够了  route add default gw 10.214.48.0 |
| **实验总结与心得**  每一个指令都是一个强大软件，包含着许多参数，要想一一详细的叙述所有的参数在短短的几页内是不可能的。我只有挑选最重要最常用的几条验证一下。 |