

# linux 网络相关入门介绍

贾明皓

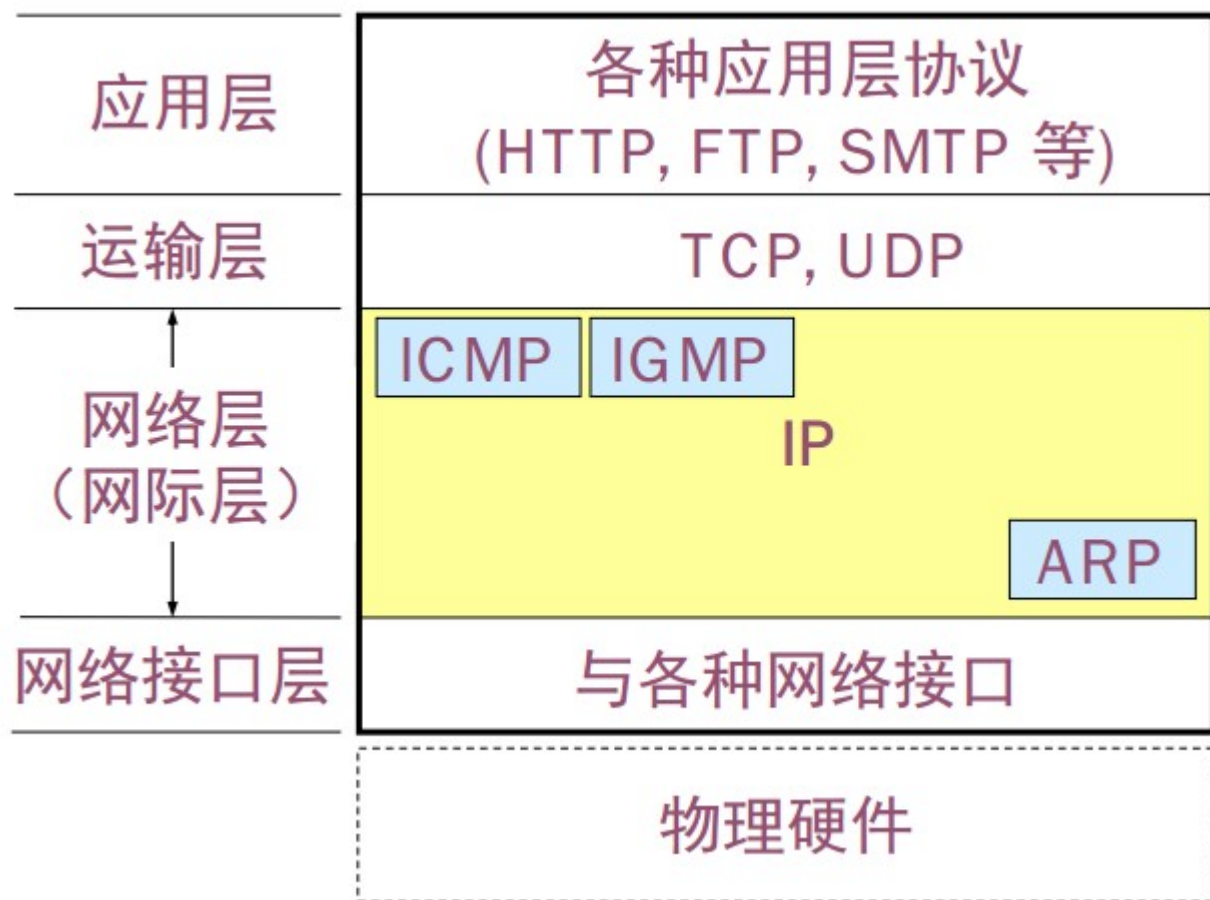
[jiajerry@mail.ustc.edu.cn](mailto:jiajerry@mail.ustc.edu.cn)

介绍日常学习生活下的所用的一些和网络相关的命令，并结合个人的使用做软件一些推荐

- 网络的基本概念
- 网络的查看配置
- 网络共享
- 一些网络应用

# 一些基本概念

tcp/ip 体系结构：



IP 地址：是 IP 协议提供的一种统一的地址格式，它为互联网上的每一个网络和每一台主机分配一个逻辑地址，以此来屏蔽物理地址的差异。

4 个 8 位二进制数组成，常用点分十进制表示：

218.104.71.168

IP 地址 = < 网络号 > . < 主机号 >

根据第一字节分 A,B,C,D,E 类网址

0,10,110,1110,1111。A,B,C 分别是前 8,16,24 位作网络号

前三个为基本类

私有地址（ Private address ）属于非注册地址，专门为组织机构内部使用。

以下列出留用的内部私有地址

A 类 10.0.0.0--10.255.255.255

B 类 172.16.0.0--172.31.255.255

C 类 192.168.0.0--192.168.255.255

子网掩码 subnet mask :

又叫网络掩码、地址掩码、子网络遮罩，它是一种用来指明一个 IP 地址的哪些位标识的是主机所在的子网以及哪些位标识的是主机的位掩码。

子网掩码不能单独存在，它必须结合 IP 地址一起使用。子网掩码只有一个作用，就是将某个 IP 地址划分成网络地址和主机地址两部分。

网关 (Gateway)：又称网间连接器、协议转换器。网关在网络层以上实现网络互连，是最复杂的网络互连设备，仅用于两个高层协议不同的网络互连。网关既可以用于广域网互连，也可以用于局域网互连。网关是一种充当转换重任的计算机系统或设备。在使用不同的通信协议、数据格式或语言，甚至体系结构完全不同的两种系统之间，网关是一个翻译器。

本机的流量要经由网关交付出去，网关有点类似宿管

由于历史的原因，许多有关 TCP/IP 的文献曾经把网络层使用的路由器称为网关，在今天很多局域网采用都是路由来接入网络，因此通常指的网关就是路由器的 IP ！

路由 route：网络层的数据包转发设备。路由器通过转发数据包来实现网络互连

因特网的核心部分是由许多网络和把它们互连起来的路由器组成，而主机处在因特网的边缘部分。

目标网址，网关，掩码

```
jerryjia@Archlinux ~ % route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.0.1    0.0.0.0         UG      203    0      0 enp5s0
192.168.12.0     0.0.0.0        255.255.255.0   U        0      0      0 wlp9s0
```



地址解析协议，即 ARP（Address Resolution Protocol），是根据 IP 地址获取物理地址的一个 TCP/IP 协议。其功能是：主机将 ARP 请求广播到网络上的所有主机，并接收返回消息，确定目标 IP 地址的物理地址，同时将 IP 地址和硬件地址存入本机 ARP 缓存中，下次请求时直接查询 ARP 缓存。

-----

ICMP 是（Internet Control Message Protocol）Internet 控制报文协议。它是 TCP/IP 协议族的一个子协议，用于在 IP 主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。

dhcp：动态主机配置协议（ Dynamic Host Configuration Protocol, DHCP ）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提供商自动分配 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

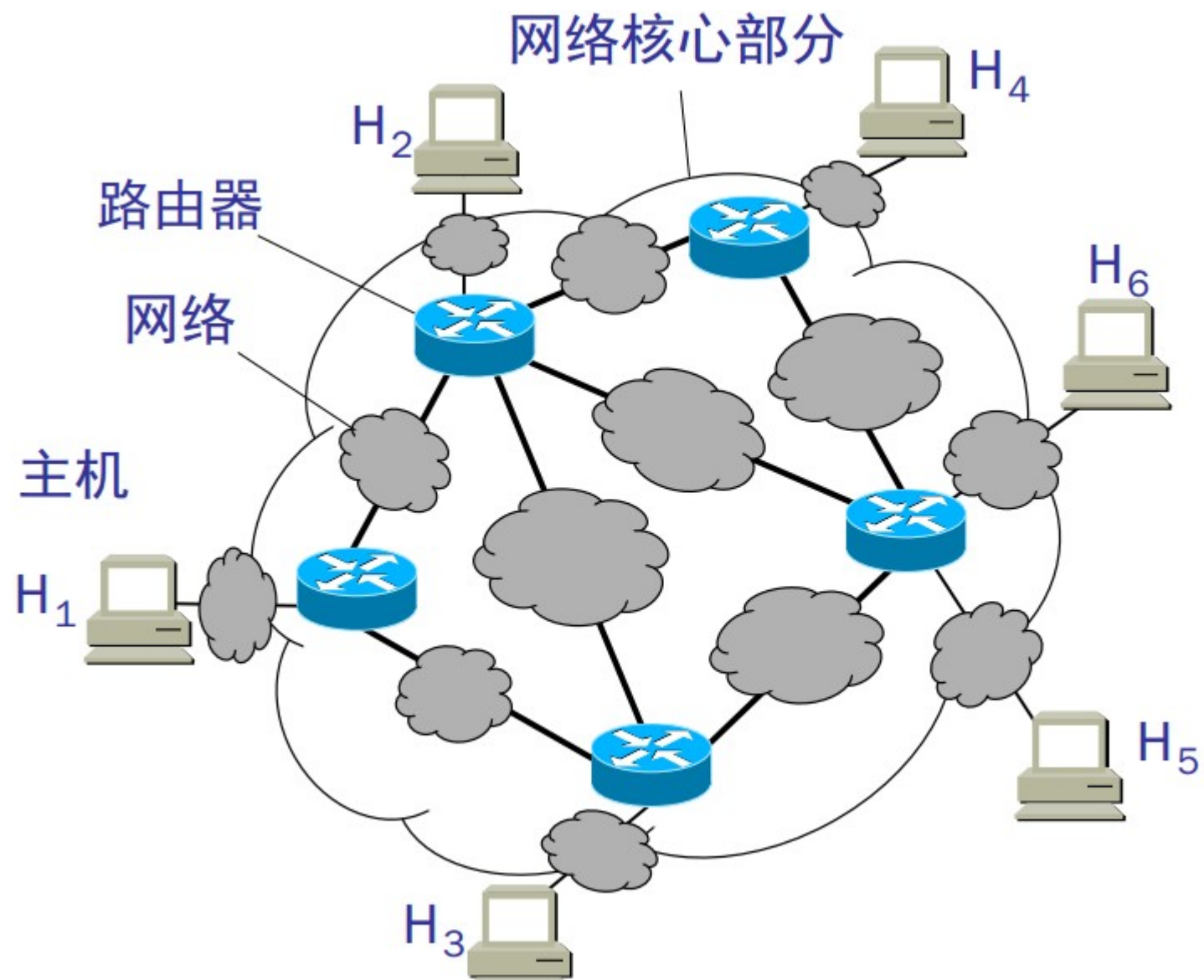
dns：域名系统，可以将域名和 IP 地址相互映射的一个分布式数据库，能够使人更方便的访问互联网，而不用去记住能够被机器直接读取的 IP

那么多概念，只要大家了解一个大致的网络过程。

一台电脑要上网，首先要在当前最小的网络环境获得一个可以用来网络通信的 ip 地址，指派给你 ip 的网关要有在 internet 上的 ip，同时要有域名解析服务。

数据流量从本机发出，交付给路由，路由查找路由表，将其转发出去。

而互联网上的设备，则经路由将数据发送到本机



日常的 linux 生活中，我们可能都安装了桌面，桌面组建往往提供了联网功能，如

networkmanager，wicd，e17 的 connman 等，但是掌握基本的网络相关命令是必要且有趣的，有时候会更加方便使用 linux

一些命令的研究我们不妨可以从这些集成的前端入手看看它到底调用了些什么

e.g. 在 Arch 下使用 wifi-menu 连接无线网络后，用 `ps -auxf | grep wl` 查看

ip

在 iproute2 包中

是一个很基本而且涉及面很广的命令

查看各网络信息 ip link , ip addr , ip route

打开关闭网络装置 ip link [-s] set <dev> up/down

修改网络装置名称 ip link set <dev> name <name>

修改网络装置 mac ip link set address :::

设定网络地址 ip addr add 192.168.12.39/24 dev <dev> broadcast + label <label>

网络路由设定 ip route add 192.168.12.0/24 dev <dev> via 192.168.1.1

<http://blog.csdn.net/radkitty/article/details/3022181>

route

路由设定

route add -net <ip> netmask <netmask> dev <devs> 添加一个本地路由

route add default gw 192.168.0.1

route -n 查看路由状态 ( 直接使用 ip 或 port num)

flags

U 路由是启动的

H 目标是一个主机

U 需要网关转送

R 启用动态路由时恢复路由资讯

D 已经由服务或转 port 功能设定为动态路由

M 路由已经被修改了

! 路由不被接受

ifconfig

net-tools

涉及网络设备管理，与 ip 的部分命令有重叠

ifconfig 仅能查看到当前活动的网卡

```
ifconfig <dev> (ip address) broadcast (broadcast)
        netmask (netmask)
```

```
ifconfig <dev> up/down
```

```
ifconfig eth0 hw ether :::
```



`iw iwconfig`

无线网络设备的设定

不过有关无线联网更推荐 `wpa_supplicant`

`iwconfig wl essid "ustcnet" mode managed key off`

`iwlist`

显示无线网络状态

`iwlist <dev> scan`

`iwlist scan | grep -i ssid` ( 找出当前可用无线 )

对于 wpa 加密的无线网络我们需要

wpa\_supplicant

wpa\_passphrase 这个可以帮助生成最基本的配置文件

/etc/wpa\_supplicant 下有例子

man 5 wpa\_supplicant.conf 有说明 + 样板

也可以用来连接 wep 加密的网络，但是那个时候用  
iwconfig 更简单 :-)

贴出我连接寝室分的无线网用的配置

```
ctrl_interface=DIR=/var/run/wpa_supplicant // 进程文件
```

```
GROUP=wheel // 指定用户组
```

```
network={
```

```
    ssid="Kancolle" // 名称
```

```
    scan_ssid=1 // 扫描
```

```
    key_mgmt=WPA-PSK // 加密类型
```

```
    psk="shimakaze" // 密钥
```

```
}
```

dhcpcd/dhclient

自动从 dhcp server 处获得可用的 ip 地址

一般来说用这个设定 ip 地址最方便 XD

而且对于一个固定网络环境下的固定设备，其实用  
dhcpcd 获得的地址往往是一样的 0.0

只分配 ipv4 : dhcpcd -4 <dev>

dhcp, 方便简单而且默认的往往就是合适的。

但是如果 dhcp server 性能不足，以及希望获得固定  
ip，或者提高 ip 获取速度，我们就需要 static ip

static ip

用开始提到的那些命令是可以的，不过那些命令一般是遇到网络问题或者有较多专业知识并且有其它配置目的使用的

```
# ip addr add 139.96.30.120/24 dev eth0
```

```
# ip link set up dev eth0
```

```
# ip route add default via 139.96.30.100 dev eth0
```

系统启动时指定。配置 /etc 下的网络相关配置文件  
不同发行版具体文件啊不一样

配置文件 /etc/resolv.conf

resolvconf resolveconf.conf

指定使用的 dns

nameserver 8.8.8.8

注意 resolv.conf 是会被 dhcp 或者 networkmanager 修改的，如果要固定它，要作一定设置

[https://wiki.archlinux.org/index.php/resolv.conf#Write-protect\\_.2Fetc.2Fresolv.conf](https://wiki.archlinux.org/index.php/resolv.conf#Write-protect_.2Fetc.2Fresolv.conf)

可以配置本地 dns 服务，一来适当解决 dns 污染问题，二来避免 dns 劫持，三可以加速访问

ping traceroute

和 icmp 报文协议相关的两个基本命令

可用来检查网络可用性，连通优良程度，检查数据包经过的路由节点

ping/ping6

-c count -t timeout -s packetsize

netstat

nmap

nslookup

tcpdump wireshark nc(netcat) . . .



## 网络共享

iptables, 启用 nat

```
# iptables -t nat -A POSTROUTING -o internet0 -j MASQUERADE
```

```
# iptables -A FORWARD -i net0 -o internet0 -j ACCEPT
```

```
# iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

```
sysctl net.ipv4.ip_forward=1 （允许转发
```

-----

dnsmasq, bridge, hostapd

dnsmasq 提供 dhcp 服务以及 dns 缓存服务

dns 缓存服务还有 pdns 等

bridge 提供桥接网络

Hostapd 实现 ap 模式无线分网的重点（还有个 ad-hoc 模式，不如 ap 好用）

Arch 的 netctl 包也提供了网络共享功能

注意各个发行版自己的网络配置

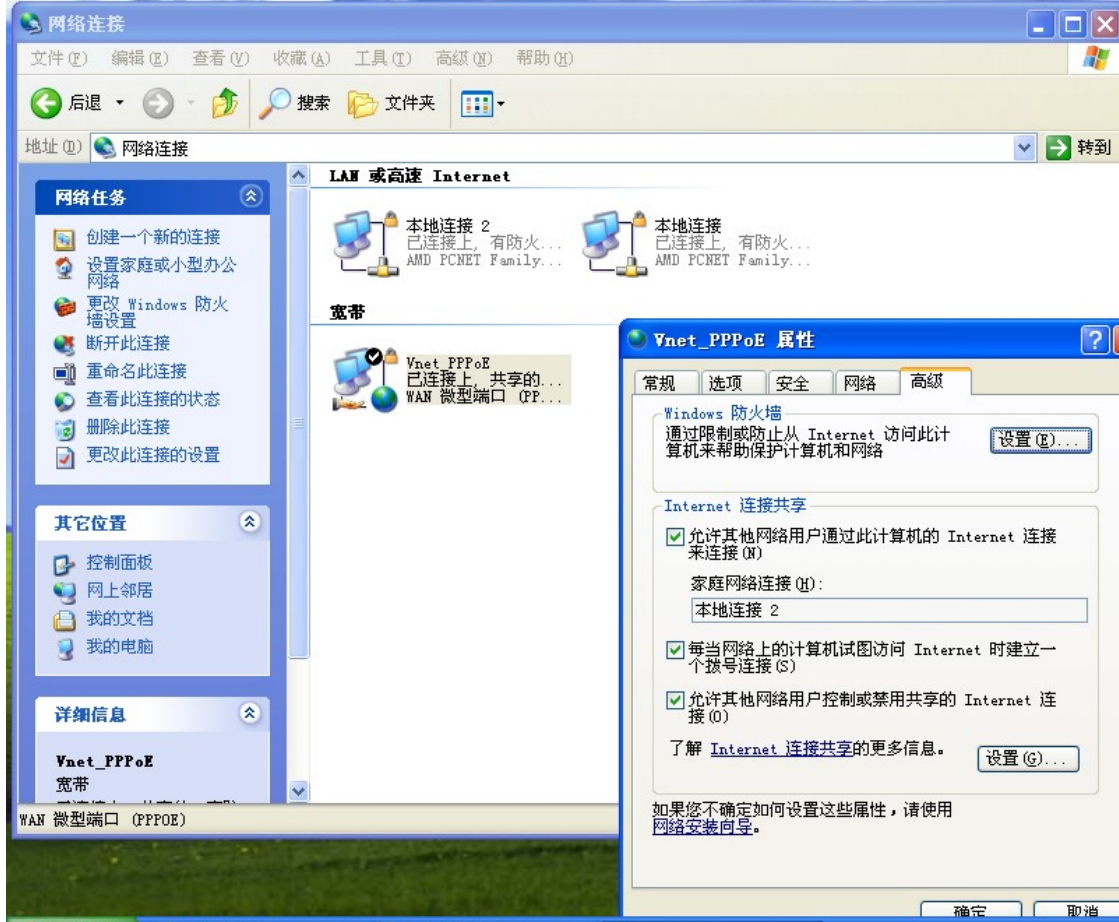
介绍自己常用的无线网络分享

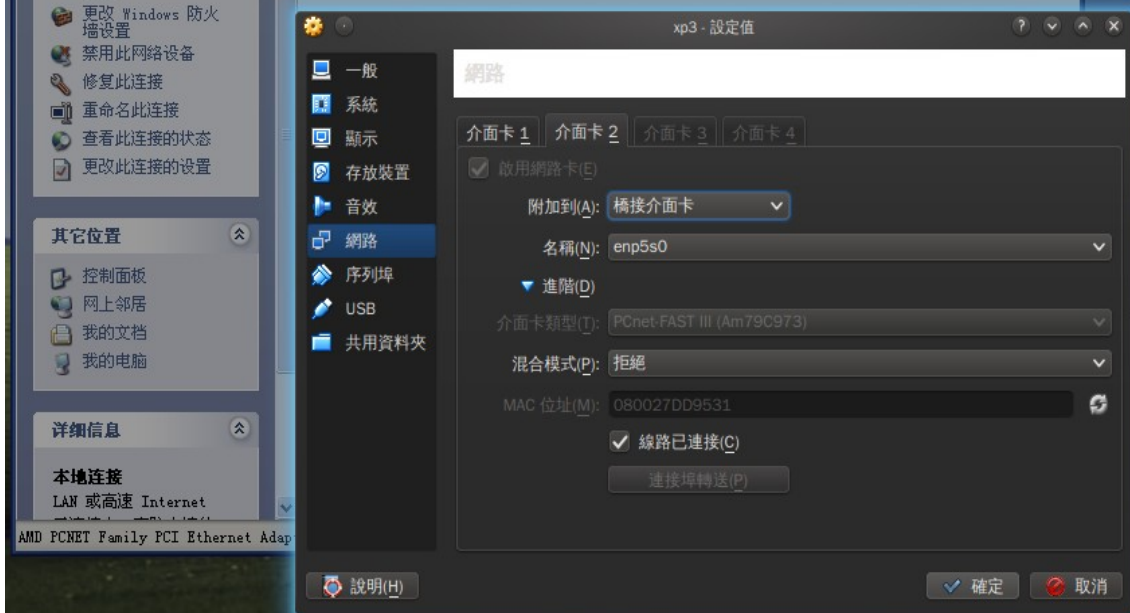
思路，一个网卡 A 获得可以访问互联网的能力

一个无线网卡 B 发出无线信号

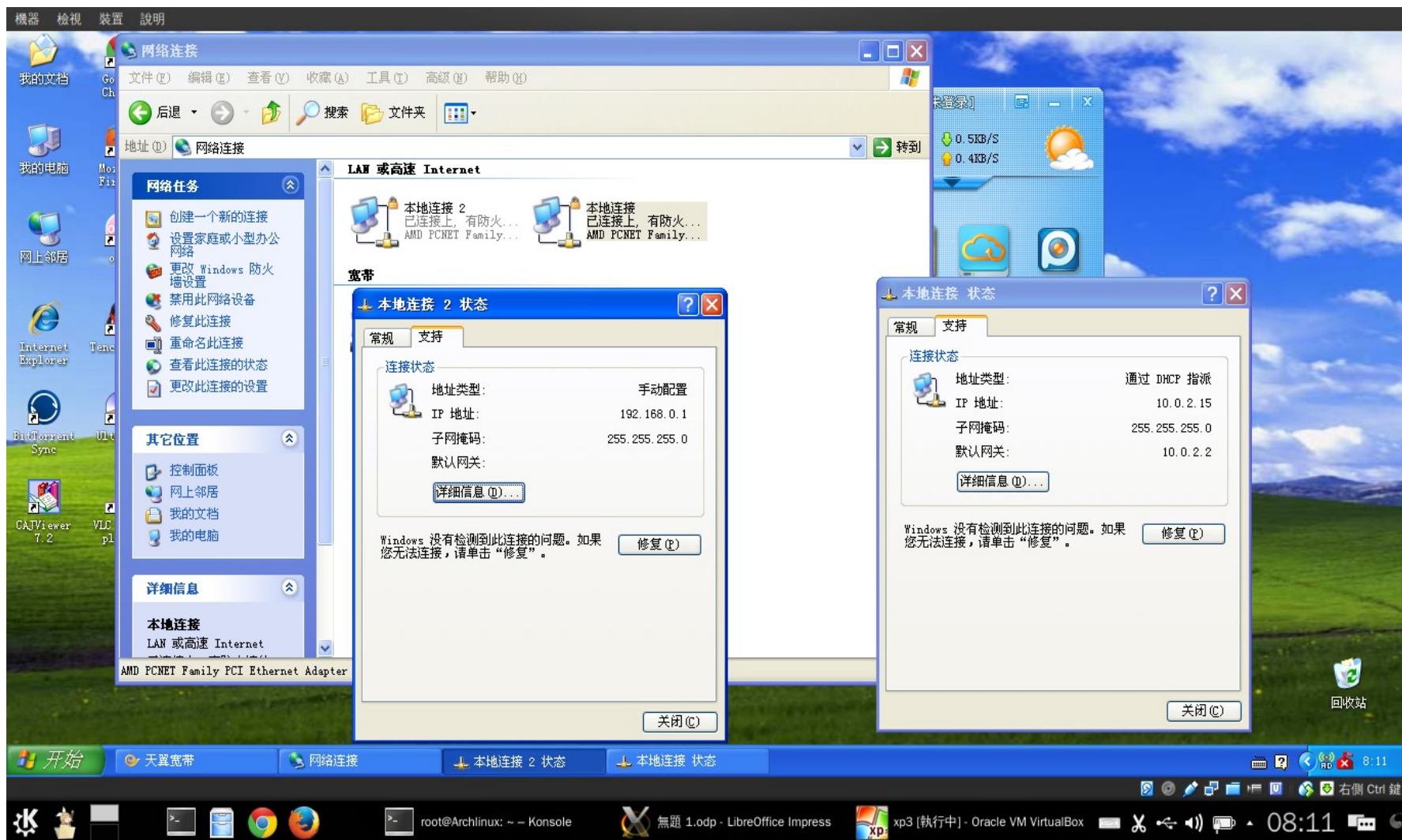
B 和 A 也是有共享 / 共用关系的

网路上也有配置好的脚本 `create_ap` 我们通过这个脚本来实现并网络共享的过程









虚拟机下的天翼客户端可以联网之后，在本机运行脚本 `create_ap <wifi interface> <interface with internet> <ssid> <password>`

该脚本自动配置了 `dnsmasq`，`hostapd` 等

脚本由 Arch 社区成员提供，可在 aur 获得

现在于 github 维护 <https://github.com/oblique>

脚本实际上由 `bash` 所写，可以通过阅读该 `bash` 进行学习

# hostapd

hostapd 可以实现软 ap, 网卡 master 模式

<http://hostap.epitest.fi/hostapd/>

<http://wireless.kernel.org/en/users/Documentation/hostapd>

**参数 -B 后台进程方式运行**

**-P 指定 pid 文件路径**

**-d show debug info**

**hostapd <path to config file>**

/etc/hostapd/

这里面有详细的配置说明与例子

主要需要指定设备，驱动， ssid ， password ，

hw\_mode,channel, 加密方式

还可以启用 .accept,.deny, 根据 mac 过滤连接

注意 hostapd 和 networkmanager 是可能有冲突的

<http://hueidou.github.io/2012/08/16/use-hostapd-set-up-wireless-ap.html>



ssid=Vocaloid

interface=wlp9s0

driver=nl80211

hw\_mode=g

channel=1

ctrl\_interface=/tmp/create\_ap.wlp9s0.conf.k8rnls8B/hostapd\_ctrl

ctrl\_interface\_group=0

ignore\_broadcast\_ssid=0

wpa=3

wpa\_passphrase=kagamine

wpa\_key\_mgmt=WPA-PSK

wpa\_pairwise=TKIP

rsn\_pairwise=CCMP

dnsmasq

dns 缓存和 dhcp 服务功能

/etc/dnsmasq

dnsmasq -C <path config file> -x <pid file>

作 dhcp 服务 :

interface=wlp9s0

bind-interfaces

dhcp-range=192.168.12.1,192.168.12.254,255.255.255.0,24h

dhcp-option=option:router,192.168.12.1

no-hosts

作 dns 缓存：

dnsmasq 添加 listen-address=<ip>

再针对 dhcpcd 或 dhclient 配置

/etc/resolv.conf.head

prepend domain-name-servers 127.0.0.1

还有个 pdns 也提供 dns 缓存服务

网速哪里去了？！

iftop 查看，结合 arp 等定位 mac，tc+iptables 解决 ( 这个我也在学习 )

iftop -i wlp9s0 查看网速

arp -n -i 查看对应 mac

arp -a

nmap 端口扫描或者对网段扫描

nmap -sn 192.168.12.1/24

man nmap 会看到中文翻译 \_(:3 」 ∠ )\_ 也是有例子说明的

## 一些有用的网络工具

wget

-o 指定 output file

-c 继续下载, 可以添加 alias wget='wget -c' 到 zshrc 等中

--local-encoding --remote-encoding 乱码时候试试

-r 递归下载

-m 镜像

-p 获得图像

-k 转换为本地地址

-np 不要进入父目录

--accept= 接受的格式

-L 不进入其它主机

支持 cookie

# cURL

个人用的不多，但是这是个很强大的网络应用层相关的命令，支持多种协议

`curl -I http://` 只查看返回的头部

`-H header`

`-X` 指定命令

欢迎常用此命令的大神多多介绍

axel , aria2c

也是命令行下的下载工具，后者支持 bt  
多线程支持，调节简单，可以限速

ftp

本机架设 ftp , vsftp (方便局域网下文件共享  
客户端 比起 filezilla 什么的, 个人推荐 lftp

lftp 可以对目录进行操作

lftp `username@host`

get 下载到本地

put 上传到服务器

-R 反向上传    mirror -R



openssh

telnet ? ssh !

都提供远程登录的功能，但是 telnet 安全性不如 ssh

ssh Secure Shell Protocol

本地安装 openssh 同时提供了客户端和服务端

意味着你既可以登陆到别的主机，也可以让别人来  
登陆

ssh(client)      sshd(server)

ssh 默认端口 22

最简单的使用: `ssh user@hostname` (然后提示密码输入)

-4 只用 ipv4

-v 详细输出调试信息 (可以帮助我们了解 ssh 过程)

-p port 指定端口 (当有其它端口开放时)

-D [bind\_address] <local port> 绑定本地端口动态转发 (socks v5 协议代理)

-N 不执行远端指令 (端口转发时候用)

-L <local port>:<target-host>:<target-host-port> ssh-server-host 本地端口转发, 访问本地 local port 后流量通过 ssh-server-host 再转发到 target-host:target port 上

-R <remote port>:<target-host>:<target-host-port> ssh-server-host

其实这里 ssh-server 也可以是本地 (localhost), 但是得先开启 sshd 服务

利用 freeshell 在寝室也能获得校园网

```
ssh -D 23333 -p 30764
```

[jerryjia@ssh.freeshell.ustc.edu.cn](mailto:jerryjia@ssh.freeshell.ustc.edu.cn)

还可以加 -T 如果不想被分配一个 pty 的话

再附上 -v 看输出可以帮助自己理解过程

然后本地浏览器进行设置，例如 chrome 就  
proxyswitchy sharp 下进行设置选 socks5 代理

还可以用 privoxy 之类的工具将其转换为 http 代理  
( 这样 ftp 什么的就不行咯 )

scp 基于 ssh 的网络传输

其实就是 ssh+cp

在 cp 的基础上添加 ssh 的 user@host:/path/to 就行了

/etc/ssh/{ssh\_config,sshd\_config}

# 文本网络浏览工具

links

lynx

w3m

这里介绍的都是比较基本简单的命令，像 ip,ifconfig,curl 等有更多强大的又灵活的功能，不过这个涉及进一步学习网络的原理，本人使用的也不多

iptables 是 linux 内核支持的防火墙工具，功能非常强大，是我们值得我们进阶学习的