随着距离的增加，信号会逐渐衰减。所以需要中继与信号放大器（带电设备）。

不同设备之间的连线使用直通线，而不是交叉线。两者区别在于制造网线时网线两端的排线规则。

数据块于上层是PDU（Protocol Data Unit），于下层是SDU（Service Data Unit），MAC层就是MSDU。从MSDU向MPDU转变的过程中可能有加密。

由于因特网发展很快而TCP/IP体系经常使用的局域网只剩下DIX Ethernet而不是IEEE802.3标准中的局域网，所以IEEE 802委员会制定的LLC子层的作用已经消失了。本来标准中规定，数据链路层有媒体访问控制子层MAC以及有逻辑链路控制子层LLC。LLC是MAC的上层:

LLC负责识别网络层协议，然后对它们进行封装。LLC 为上层提供了处理任何类型 MAC 层的方法。

MAC主要负责控制与连接物理层的物理介质。MAC子层的主要功能包括数据帧的封装/卸装，帧的寻址和识别，帧的接收与发送，链路的管理，帧的差错控制等。MAC子层的存在屏蔽了不同物理链路种类的差异性.

现在的标准就是DIX EthernetII，只有一层，并没有已经消失的标准中描述的这么简单。和Ethernet不同,802.11使用LLC层来封装上层协议，所以无线包就有考虑LLC和MAC这种分层。

集线器可以直接嗅探，因为所有数据的发送形式是广播。所以需要网络流量镜像只用安装一个集线器就好。默默坑人的就是无线集线器，支持发送wifi，你把原来的wifi AP设备替换成这个。

思科交换机有着自己的操作系统，名字叫做思科IOS，但是一个交换机里面的操作系统可以不止一个。启动的时候会进行自检（对硬件），汽车启动的时候也会对电路进行自检-》找到操作系统并启动-》找到配置文件并且加载交换机的配置包含：

协议编址和参数设置

管理和维护的选项

思科网络设备有三种口：网络口、控制口、AUX（辅助端口）远程拨号（通过Modem连接到这个AUX端口），对应不同的配置方式。交换机的配置存放有两个地方RAM（断电以后数据消失）以及NVRAM（直接写在上面永久存储，断电以后数据不会消失）。通过show running-config 看到RAM存的当前的配置，show startup-config 查看已经保存到NVRAM中的启动配置。FLASH存放IOS。通过copy run start(wr mem似乎也是)将running-config 保存到startup-config中。

交叉线Cross-Over（形状是黑色虚线）：用来连接2个相同设备。注意：路由器和电脑要用交叉线

直通线Straight-Through（形状是黑色直线）：用来连接2个不同设备。如交换机和路由器用直通线

控制方式：命令方式（CLI）以及网页的方式

双击以后的config标签中可以给每一个接口进行基本设置，包括IP以及子网掩码（交换机VLAN的设置还是需要使用命令来查看的）。本质上还是使用了命令，下面的命令窗口中都看的出来。

将鼠标悬停在设备上可以看到设备当前的简略配置信息

命令：

Would you like to enter the initial configuration dialog? [yes/no]: n 否则就回进入交互模式问你一大推问题

?就可以看到当前模式下支持的命令有哪些。按方向键的向下的箭头可以直接中断显示，或者回车键行行下来。

只有输入了一个词的部分以后，使用tab能对这个词补全。

前面加上一个no 就会去除这个命令产生的效果。

en进入特权模式-》conf t 进入配置模式-》int vlan 1 进入vlan 1的配置模式 exit回退，C-C直接退出到最上层

int f0/0 表示f类型接口的0号模块的0号接口

int f0/0.1和int f0/0.2 继续表示0,1两个子接口

S口是serial接口的意思，也叫高速异步串口

E口是Ethernet接口，叫以太网接口，也是主要连接以太网（局域网）用的，也是用普通的双绞线就可以连接，速率默认是10Mbps，现在新型的设备上已经把这个接口淘汰了

F口是FastEthernet接口，叫快速以太网口，主要连接以太网（局域网）用的，说白了就是连接交换机或电脑用的，用普通的双绞线就可以连接，速率默认是100Mbps，可以用命令限速，但是不可能超过100Mbps。

Console口，叫控制口，这个接口是用来调试路由器的

AUX接口，也是辅助控制接口

vty telnet、ssh远程 管理接口。

G口，是千兆以太网接口，是连接以太网用的

如果不知道简写是什么，就直接conf t进入配置信息的更改模式以后，通过interface GigabitEthernet0/1（《=》int g0/0 就是第一个首字母缩写来表示）进入接口的调整模式。

ip default-gazteway 192.168.1.1 配置默认网关。

show interface vlan 1 查看指定接口信息

输入no ip以后再打个问号就能看到之后的命令会是什么。

show run等命令，不是按方向键来继续查看，而是按回车键。

Switch(config)#no ip domain-lookup?

domain-lookup

Switch(config)#no ip domain-lookup ?

<cr> 这个表示这个命令已经结束了

no ip domain-lookup 这个命令是不对域名进行ip解析防止输错命令

no shut 从默认shutdown状态称为开启状态

show history 查看已经执行过的命令

reload 重启

service passwrod-encryption 开启密码加密服务

no service passwrod-encryption 关闭密码加密服务

enable password xxx 设置进入特权模式的密码（明文显示在配置文件中）

enable secret xxx 设置进入特权模式的密码

line console 0 （这条命令会进入对console端口的配置）->login->password xxxx 设置console口登录的密码。

line vty 0 4（这个4表示最多同时在线的连接数）->login->password xxxx 设置VTY终端的密码，使用ssh或者telnet登录。如果上面的步骤中将login变成no login就会没有密码。如果要设置超时的时间，也是要在line vty 0 4之后，设置的是连接的超时。

line aux 0->login->password xxxx 设置复制控制台登录的密码。

按住MODE键不放以后，开机直到Ssytem LED灯不再闪烁： 重置

flash\_init 初始化flash

dir flash 查看flash中有什么文件

rename/copy flash:config.txt flash:config.old 将txt 通过rename/copy进行备份

boot 重启并且回到出厂状态

双工设置：int f0/1 （很明显这是命令的缩写）进入配置模式以后 配置通过show int f0/1 查看

自动协商要求两边的双工设置一样

duplex full/half/auto 全双工/半双工/自动协商

speed 10/100/1000/auto 配置速率

冲突检测：

show interface 查看接口冲突总是与总报文的比值，如果多就表明噪声过大。

有的仪器能查出在大致多远的地方网线断了。

ACL的编写：

先是config模式下：

access-list 编号数字1-99 { permit|deny/|remark } IP ACL\_MASK

remark允许你给ACL添加一些描述性的注释信息

access-list access-list-number remark 这里是注释

如果是扩展：

access-list access-list-number {permit | deny} \

protocol source source-wildcard [operator port] \

destination destination-wildcard [operator port] \

[established] [log]

配置命名的ACL：

ip access-list {standard | extended} name 进入config {std- | ext-}nacl模式

在模式中通过[sequence-number] {permit | deny} {ip access list test conditions} {permit | deny} {ip access list test conditions} 制定规则，扩展ACL能够基于源IP地址、目标IP地址、协议号以及端口号进行过滤。

下面是可以指定顾虑协议的扩展规则：

10 deny tcp 192.168.10.0 0.0.0.255 any eq 25

11 deny tcp 192.168.10.0 0.0.0.255 any eq telnet

no sequence number 能够删除一条规则

示例：

RouterX(config)#ip access-list standard troublemaker

RouterX(config-std-nacl)#deny host 172.16.4.13

RouterX(config-std-nacl)#permit 172.16.4.0 0.0.0.255

RouterX(config-std-nacl)#interface e0

RouterX(config-std-nacl)#remark 这里是注释

RouterX(config-if)#ip access-group troublemaker out

no access-list 删除整个ACL

进入接口的config-if模式：

ip access-group access-list-number {in | out} 将创建的规则用于指定的接口的in或者out

no ip access-group access-list-number {in | out} 解除指定的规则。

ip access-group name {in | out} 应用命名指定的规则

RouterX(config)# access-list 1 permit 172.16.0.0 0.0.255.255

RouterX(config)# interface ethernet 0

RouterX(config-if)# ip access-group 1 out

查看：

show access-lists 直接查看所有的规则

show access-lists {access-list number|name} 查看指定规则

show ip interface 查看所有的接口

show ip interface s0 查看指定接口

NAT：

静态static手动配置；

ip nat inside source static local-ip global-ip 静态设置一条NAT表记录

no ip nat inside source static local-ip global-ip 删除静态添加的记录

进入接口的配置以后：

ip nat inside/outside 标志该接口连接到内部/外部网络

动态dynamic配置：

ip nat pool name start-ip end-ip {netmask netmask | prefix-length prefix-length} 地址池

通过ACL限制可以参与动态NAT转换的内部地址

ip nat inside source list access-list-number pool name 将动态转换配置上去

show ip nat translations（《=》debug ip nat） 查看NAT转化表(show命令最顶层才会有)

Inside global 是内部有地址出来以后

Inside local 是内部实际地址

Outside local以及Outside global是目的地址

PAT：也称为 overloading 的NAT

ip nat inside source list access-list-number interface interfaceName overload 指定可以复用ip的接口

为了确定当前某条NAT转换条目是否被使用，可以使用命令 show ip t nat statistics检查命中计数器(hits counter)的值。

PPP配置：

RouterX(config-if)# encapsulation ppp

RouterX(config-if)# ppp authentication {chap | chap pap | pap chap | pap} 启用PAP或CHAP验证

RouterX(config)# hostname name 配置路由器的名称

RouterX(config)# username name password password 为远端路由器配置用户名(机器名）和密码、

show interface查看PPP的封装

debug p ppp negotiation观察PPP的协商过程。

CISCO Package Tracker：

直通线在cisco的软件中就是黑色实线。

双击电脑以后-》Desktop-》ip就可以配置ip，Command Prompt是模拟的终端

Cisco ASA( 安全应用设备，即防火墙) )

VPN可以在Cisco的多种设备上来部署: Cisco IOS路由器、ASA 5500系列安全应用设备和Cisco VPN客户端软件等。

数据格式：帧-》包（网络层）-》段（传输层）

Hub 不分冲突域。Hub 因为没有端口之间划分冲突域，所以hub的端口不会很多，所有口都用上去了容易导致碰撞失效。

Switch\Bridge 虽然隔离冲突域，但是不隔离广播（因为Switch\Bridge可以在一个网段内）

Router 隔离冲突域，同时也隔离广播（广播帧只会在指定的内网中广播，路由器就是连接不同的网段，广播不能跨网段）

交换机：

交换模式有

Cut-through 直通转发（直接转发，所以不会进行判断）

Store-and-forward 存储转发（存储并且进行判断）

Fragment-free 无碎片转发（直通转发的一种，只是对已经有了的大小进行判断，大于x字节才会转发）

二层交换机的arp表是端口与mac之间的映射，arp表是通过对自己转发包中mac和端口的对应关系不断记录“学习”到的。

一个vlan就是一个自己划分的网段。vlan数据包有802.1Q封装的

按照分层有Access Layer接入层，提供网络接入-》Distribution Layer接入层，提供基于策略的连接、流量汇聚-》Core Layer核心层，最高速的数据转发

传播时延(propagation delay)与发送时延(transmission delay) 。transmission delay涉及到包的大小与带宽，带宽是由端口的软硬件决定的。吞吐量说的是某时刻速率（或者一段时间之内的平均速率）两个端点之间能够传输的最大速率，带宽（带宽指单位时间能通过链路的数据量）是上限。带宽是连接设备以及连接用的线种类一起决定的。

傻瓜式交换机是指不具备生成树协议，不能自动切断级联交换机之间冗余端口的交换机。这种老式的东西不要忘记了，有时候能让自己方便很多。STP 生成树协议。广播风暴（broadcast storm）简单的讲是指当广播数据充斥网络无法处理，并占用大量网络带宽，导致正常业务不能运行，甚至彻底瘫痪，这就发生了“广播风暴”。所以不一定要成为一个循环。

核心交换机要使用带有端口隔离,端口限速,组播抑制以及dhcpsnoop功能的。当交换机开启了 DHCP-Snooping后，会对DHCP报文进行侦听，并可以从接收到的DHCP Request或DHCP Ack报文中提取并记录IP地址和MAC地址信息。另外，DHCP-Snooping允许将某个物理端口设置为信任端口或不信任端口。可以用于防止arp缓存攻击。

借助Cisco Package Tracker才知道原来进行HTTP访问的时候TCP连接建立时的第三个发包后面紧接着就发出了HTTP。

确定是否出错只需要一个位，确定并尝试纠正一个位的错误，那么要将其按照某种组织形式弄成二维表格形状，然后横纵分别进行异或运算，得到两组数据（CRC默认是xx位一组进行运算）。CRC只有检错，海明码可以纠错一位。

CRC:

CRC计算可以靠专用的硬件来实现,但是对于低成本的微控制器系统,在没有硬件支持下实现CRC检验,关键的问题就是如何通过软件来完成CRC计算

CRC计算机字节型算法的推导：

256^n表示256的n次方，Y[n]表示第n个字节的CRC，Z[n]是计算CRC所得到的商。

把按字节排列的数据流表示成数学多项式，设数据流为BYTE[n]BYTE[n－1]BYTE[n－2]、、、BYTE[1]BYTE[0]，表示成数学表达式为BYTE[n]×256^n+BYTE[n-1]×256^(n-1)

+...+BYTE[1]\*256+BYTE[0]，在这里+表示为异或运算。设生成多项式为G17（17bit），CRC码为CRC16。

则，CRC16＝(BYTE[n]×256^n+BYTE[n-1]×256^(n-1)+...+BYTE[1]×256+BYTE[0])×256^2/G17，即数据流左移16位，再除以生成多项式G17。

先变换BYTE[n-1]、BYTE[n-1]扩大后的形式，

总序列＝BYTE[n]×256^n×256^2/G17+BYTE[n-1]×256^(n-1)×256^2/G17+...+BYTE[1]×256×256^2/G17+BYTE[0]×256^2/G17（分配率得到的吧）

＝(Z[n]+Y[n]/G17)×256^n+BYTE[n-1]×256^(n-1)×256^2/G17+...+BYTE[1]×256×256^2/G17+BYTE[0]×256^2/G17

＝Z[n]×256^n+{Y[n]×256/G17+BYTE[n-1]×256^2/G17}×256^(n-1)+...+BYTE[1]×256×256^2/G17+BYTE[0]×256^2/G17

＝Z[n]×256^n+{(YH8[n]×256+YHL[n])×256/G17+BYTE[n-1]×256^2/G17}×256^(n-1)+...+BYTE[1]×256×256^2/G17+BYTE[0]×256^2/G17

＝Z[n]×256^n+{YHL[n]×256/G17+(YH8[n]+BYTE[n-1])×256^2/G17}×256^(n-1)+...+BYTE[1]×256×256^2/G17+BYTE[0]×256^2/G17

这样就推导出，BYTE[n-1]字节的CRC校验码为{YHL[n]×256/G17+(YH8[n]+BYTE[n-1])×256^2/G17}，即上一字节CRC校验码Y[n]的高8位（YH8[n]）与本字节BYTE[n-1]异或，所得的CRC校验码与上一字节CRC校验码Y[n]的低8位（YL8[n])乘以256（即左移8位）异或。

字节型算法如下：

1)CRC寄存器组初始化为全"0"(0x0000)。（注意：CRC寄存器组初始化全为1时，最后CRC应取反。）

2)CRC寄存器组向左移8位,并保存到CRC寄存器组。

3)原CRC寄存器组高8位（右移8位）与数据字节进行异或运算，得出一个指向值表的索引。

4)索引所指的表值与CRC寄存器组做异或运算。

5)数据指针加1，如果数据没有全部处理完，则重复步骤2)。

6)得出CRC。

因为IP层只负责把数据送到节点上，而不能区分上面的不同应用，所以TCP和UDP协议在其基础上加入了端口的信息。

Web Service技术， 能使得运行在不同机器上的不同应用无须借助附加的、专门的第三方软件或硬件， 就可相互交换数据或集成。

Web service是一个平台独立的，低耦合的，自包含的、基于可编程的web的应用程序

查表法:（这算法似乎算的不对，可能表格有错误）

static unsigned short ccitt\_table[256] = {

0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,

0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,

0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,

0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,

0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,

0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,

0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,

0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,

0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,

0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,

0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,

0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,

0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,

0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,

0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,

0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,

0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,

0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,

0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,

0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,

0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,

0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,

0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,

0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,

0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,

0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,

0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,

0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,

0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,

0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,

0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,

0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0

};

unsigned short crc\_ccitt(unsigned char \*q, int len)

{

unsigned short crc = 0;

while (len-- > 0)

crc = ccitt\_table[(crc >> 8 ^ \*q++) & 0xff] ^ (crc << 8);

return ~crc

}

我按照人工笔算的意思写的两种算法：（这两种算法与在线算的不一样，但是与我手算的一样）

1.

uint32\_t CRC(unsigned char \*data, unsigned short size) {

uint32\_t temp = ((uint32\_t)data[0] << 24) | ((uint32\_t)data[1] << 16) | ((uint32\_t)data[2] << 8) | ((uint32\_t)data[3]);

unsigned short pos = 4;

int count = 0;

while (true) {

temp ^= CRCDIV;

while (((uint32\_t)0x80 << 24 & temp) == 0) {

temp <<= 1;

count++;

if (count == 8 && pos != size) {

temp |= data[pos++];

count = 0;

}

if (count == 16 && pos == size) {

return (temp >> 16);

}

}

}

}

2.

uint32\_t CRC2(unsigned char \*data, unsigned short size) {

uint32\_t temp = ((uint32\_t)data[0] << 24) | ((uint32\_t)data[1] << 16) | ((uint32\_t)data[2] << 8) | ((uint32\_t)data[3]);

unsigned short pos = 4;

while (pos != size) {

for (int i1 = 0; i1 < 8; i1++) {

if(((uint32\_t)0x80 << 24 & temp) == 0)//不可以合并temp<<=1;哦

temp <<=1 ;

else {

temp <<= 1;

temp ^= CRCDIV2;

}

}

temp |= data[pos++];

}

for (int i1 = 0; i1 < 16; i1++) {

if (((uint32\_t)0x80 << 24 & temp) == 0)

temp <<= 1;

else {

temp <<= 1;

temp ^= CRCDIV2;

}

}

return temp >> 16;

}

CRC ModBus算法：

//在线验证过了

unsigned int crc\_chk(unsigned char\* data, unsigned char length)

{

int j;

unsigned int reg\_crc=0xFFFF;

while(length--)

{

reg\_crc ^= \*data++;

for(j=0;j<8;j++)

{

if(reg\_crc & 0x01) /\* LSB(b0)=1 \*/

reg\_crc=(reg\_crc>>1) ^ 0xA001;//那么02不就很大了么

else

reg\_crc=reg\_crc >>1;

}

}

return reg\_crc;

}

uint32\_t CRC3(unsigned char \*data, unsigned short size) {

}

4层协议使用1-255来标识，0是不可以的。但自然不是每一个数字都可能会有对应的协议。

tcp，udp包一般没有Options字段。

TCP还使用一个拥塞控制机制。该机制迫使每个新的TCP连接一开始以相对缓慢的速率传输数据，然而只要网络不拥塞，每个连接可以迅速上升到相对较高的速率。这个慢速传输的初始阶段称为缓启动(slow start)。

基于这些协议调节窗口所利用的反馈信息，我们可以大致地将其划分为3 类：

1) 基于丢包反馈的协议(loss-based congestion avoidance,简称LCA);

2) 基于路径延时反馈的协议(delay-based congestion avoidance,简称DCA);

3) 基于显示反馈的协议(ECN(explicit congestion notification)-like algorithmsz） 路由器返回拥塞信息通知发送端

TCP拥塞控制-慢启动、拥塞避免、快重传、快启动： http://blog.csdn.net/jtracydy/article/details/52366461 TCP Reno版本与不用的TCP Tahoe版本

tcp协议规范RFC要求每秒更换序列号25W次，但是实际上更换的频率远远小于，所以导致下一次更换的数字往往是可以预测的。不过现在基本上操作系统实现上已经让初始序列号的产生具有随机性。

除了RFC标准，还有一种叫做IETF（互联网工程任务部门Internet Engineering Task Force）标准

Circuit switching的最大特点是终端系统之间需要预约传输线路资源才可以进行持续的通讯，在通讯过程中传输速率保持在一个常数值。也就是说，circuit switching是一种电路资源预分配的方式，由于资源已经预先分配，因此在通讯结束之前，不管用户之间是否一直在传输信息，这条电路始终被这一对用户占用。

跟circuit switching不同，packet switch是不需要预约的，则每个connection是竞争关系，对于线路资源先到先得。就好像我们外出餐馆吃饭，进去餐馆看到有空位就坐下，把这个位置占了，没有空位就只能等了；而circuit switching则是要先预约座位，去到餐馆凭着预约号直接坐下。

电路交换 有两种模式：FDM 频分复用、TDM时分复用 老师pdf上FDM适合于模拟信号传输、TDM适合数字信号的传输。我觉得都适合。

分组交换包含两种基本途径： https://zhidao.baidu.com/question/503124110.html

虚拟电路（虚电路）分组交换（使得网络层变得面向连接） — 在两个终端节点会话期间,为传送所有分组信息,需在中间节点间建立路由,即我们所知的初始安装阶段.在每个中间节点中,需要在表中注册一个实体,表示连接中需要的路由已经建立。相关的词汇有VCs ：X.25 、ATM 、帧中继。注意虚电路属于分组交换。ATM在congestion control（拥塞控制）上，链路上可以返回拥塞情况，而不像TCP一样——仅仅根据终端的反馈信息决定。VC除了适合用在特殊的需要大量数据或者稳定快速的地方，还有即像电话一样，终端功能简单但是链路节点功能强悍。许多虚电路协议通过数据重传，包括检错纠错和自动重传请求（ARQ），提供可靠的通信服务。

数据报交换 — 该途径采用一种不同的、更动态的模式决定网络链路中的路由.将每个分组看作一个实体,每个分组头部包含关于分组目的地的全部信息.中间节点通过检查分组头部,决定能发送分组到达目的地的节点.

datagram networks（数据报网络）

virtual circuit networks（虚电路网络）

今天我又查阅了一些资料，发觉一个问题，TCP所谓的面向连接的特性以及建立虚电路，与网络层虚电路服务好像有着本质的区别，网络层的带有虚电路的服务的协议典型的有曾经应用在广域网的X.25和帧中继，这两个协议在建立连接时要建立所谓的虚电路号，在传输的数据单元中有已经填好的虚电路号，数据单元沿着既定的虚电路沿着既定的方向（一定的路由器序列）到接收端，这样能够保证数据单元到达接收端是有序地，而且也能很好的保障可靠传输。

多路访问协议Multiple Access protocols：

信道划分协议（时分复用、频分复用）：TDMA,CDMA是电路交换

随机接入协议：CSMA是分组交换

轮流协议：轮询协议、令牌传递协议

全双工传输不需要 CSMA/CD

arp是二层协议，所以不可路由，但是存在特殊情况被路由器转发。连接IP层与数据链路层。ping等命令很容易触发arp request。

当进入一个Open 无线的时候，可以看到有一种宣告自己的ARP，接收地址与发送地址都是自己的广播包，就是为了宣告自己已经使用了哪一个IP。

arp怪像：

小米mini会非常频繁地向电脑定向arp查询，电脑会返回

华为tel广播对网关的arp，但是小米mini没有理会。arp攻击是可以成功的。

按照7层ISU设计的协议很少很少

直接发出tcp ACK包，会被返回RST。

不存在对RST包的响应。

交换机是一个二层设备,它会根据包的目标地址将包只发送到对应地址的网卡,不会发送给其他网卡,混杂模式自然是抓不到的

无线局域网内抓包的几种方法：大致可以分为你是否与通路可以搭上关系

与必要通路搭上关系的有1.交换机端口镜像（公司里面用） 2.arp欺骗让自己成为通路的一部分

无关系的话就是设置网卡混杂模式抓包，但是这样子需要你的网卡是特殊的。比如说kali支持。

-

BT、ed2K都是P2P通信的一种协议看。文件存放在用户的电脑上（不过为了加快，也有专门存放文件的平台服务器来存储近期热门文件），不论是客户端还是服务器端都有“收”和“发”的功能，收发双方互为服务器。

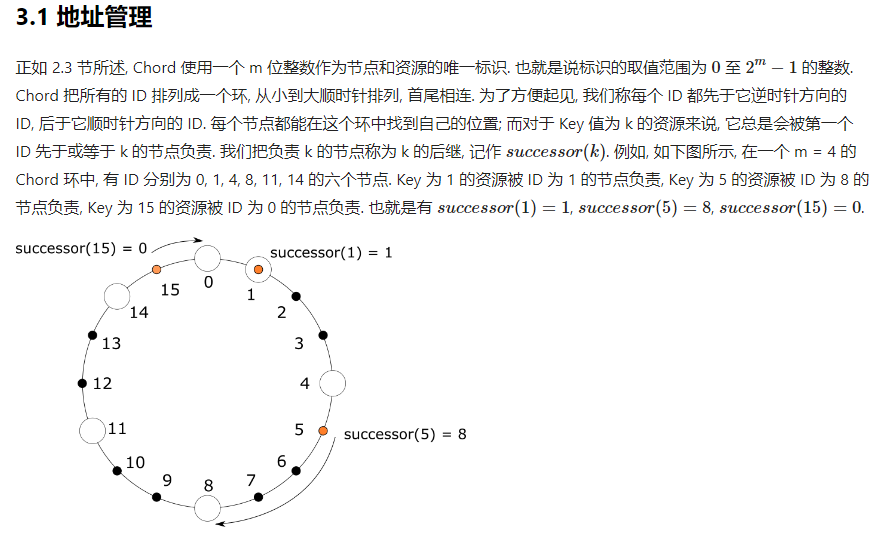
集中式，存在中心节点（存储了文件与已启动机器之间的映射关系）

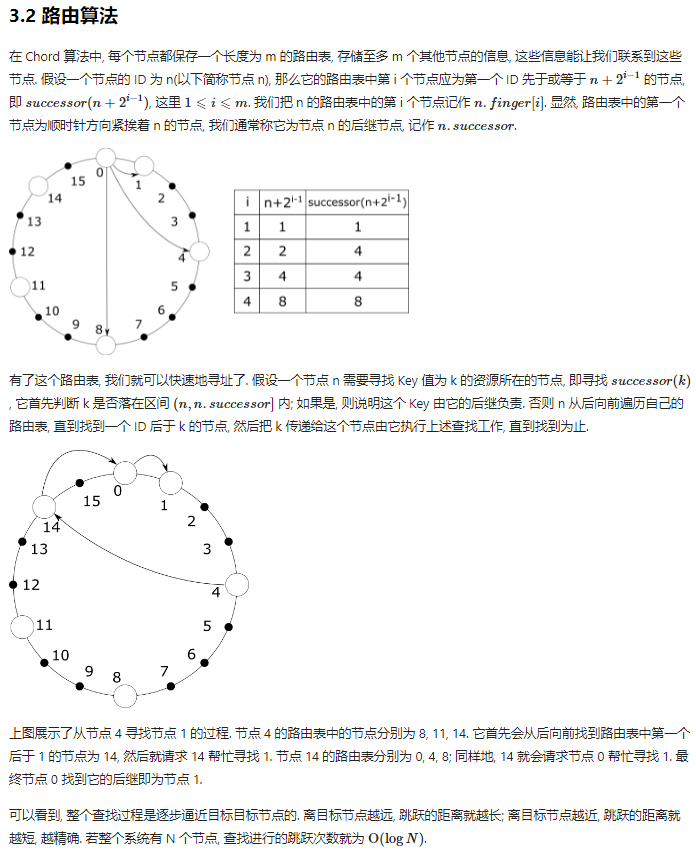
分布式，先找到一个已经在网络中的节点建连（比特币查询特定的DNS服务器），之后通过泛洪广播将新节点加入到网络中。因为ttl，可能看到的只是网络的一角。

混合式，在网络中设置层级，每一部分网络都选出“关键路由/中心节点”，新节点的加入只需要在关键路由之间传输即可。在一个有 n 个节点的分布式哈希表中, 每个节点仅需存储log n 个其他节点, 查找资源时仅需请求 log n 个节点, 并且无需中央服务器, 是一个完全自组织的系统。实际下载上，节点存储的索引信息一定是冗余的，并考虑文件的分块后索引，通过magnet磁力链接获取BT文件（infohash），记录了每一个文件分块的索引与MD5 。

DHT（分布式哈希表）用于存储具体的实现方案有 Chord、Pastry、CAN、Kademlia 等算法，其中 Kademlia 也是以太坊网络的实现算法，很多常用的 P2P 应用如 BitTorrent、电驴等也是使用 Kademlia。

Chord 算法：这种地址管理或许不需要再用DHT思想来路由，如果不大的话，各自存一份完整路由或者安排一个中心节点，直接hash映射到资源存在的节点O(1)。下图中黑色节点是资源，白色节点是路由，资源可以认为存储在节点上。





如果希望节点加入，会引起固定方向的遍历（因为每一个节点只能存储之后的logN），更新可能需要更新的节点。

Kademlia 算法 <https://shuwoom.com/?p=813> <https://luyuhuang.tech/2020/03/06/dht-and-p2p.html>

理解实现这么分布式存储的优缺点是什么？与Chord算法相比呢

wireshark查看协议关系上是先进行IP层的封装之后才会是P2P协议的封装，再到PPPoe，到Ethernet信息，最后到Frame信息。

overlay network 覆盖网络，定义为建立在另一网络上的网络，比如说P2P。我的理解就是按照某种规则建立起来的虚拟网络。

在MAC层中，有两种信号接入方式：信标模式、非信标模式。superframe”的格式，在超帧的开始发送信标帧，里面含有一些时序以及网络的信息，紧接着是竞争接入时期，在这段时间内各节点以竞争方式接入信道，再后面是非竞争接入时期，节点采用时分复用的方式接入信道，然后是非活跃时期，节点进入休眠状态，等待下一个超帧周期的开始又发送信标帧。非信标模式就是节点一直竞争。

0.0.0.0 这个ip如果作为监听目标的话表示接受所有的ip

因为自己不清楚对方的实际IP，所以就设置为0.0.0.0，用作

如果你在网络设置中设置了缺省网关,那么Windows系统会自动产生一个目的地址为0.0.0.0的缺省路由

组播地址:239.255.255.250是SSDP(简单服务发现协议),这是路由器的UPNP服务使用的协议。UPnP 的应用范围非常大，以致足够可以实现许多现成的、新的及令人兴奋的方案，包括家庭自动化、打印、图片处理、音频 / 视频娱乐、厨房设备、汽车网络和公共集会场所的类似网络。UPnP 是各种各样的智能设备、无线设备和个人电脑等实现遍布全球的对等网络连接（P2P）的结构。SSDP是基于UDP的。

NOTIFY \* HTTP/1.1

HOST: 239.255.255.250:1900

CACHE-CONTROL: max-age=60

LOCATION: http://192.168.2.1:5000/rootDesc.xml

SERVER: K2/OpenWrt/Barrier\_Breaker\_\_unknown\_ UPnP/1.1 MiniUPnPd/1.8

NT: urn:schemas-upnp-org:device:InternetGatewayDevice:1

USN: uuid:e5e33f8f-d28a-4ab2-b3f4-d34cbec73ae2::urn:schemas-upnp-org:device:InternetGatewayDevice:1

NTS: ssdp:alive

OPT: "http://schemas.upnp.org/upnp/1/0/"; ns=01

01-NLS: 1

BOOTID.UPNP.ORG: 1

CONFIGID.UPNP.ORG: 1337

包的行都是\r\n结尾

* 1XX：信息性状态码，表示接收的请求正在处理
* 2XX：成功状态码，表示请求正常处理完毕
* 3XX：重定向状态码，表示需要进行附加操作以完成请求
* 4XX：客户端错误状态码，表示服务器无法处理请求
* 5XX：服务端错误状态码，表示服务器处理请求出错

常见的状态码有：

* 200 OK，请求被正常处理
* 301 Move Permanently，永久性重定向
* 302 Found，临时性重定向
* 400 Bad Request，请求报文中存在语法错误
* 403 Forbidden，对请求资源的访问被服务器拒绝
* 404 Not Found，在服务器上不能找到请求的资源
* 500 Internal Server Error，服务器内部错误

Header Etag字段：

Etag是 Entity tag的缩写，可以理解为“被请求变量的实体值”，Etag是服务端的一个资源的标识，在 HTTP 响应头中将其传送到客户端。所谓的服务端资源可以是一个Web页面，也可以是JSON或XML等。服务器单独负责判断记号是什么及其含义，并在HTTP响应头中将其传送到客户端。比如，浏览器第一次请求一个资源的时候，服务端给予返回，并且返回了ETag: "50b1c1d4f775c61:df3" 这样的字样给浏览器，当浏览器再次请求这个资源的时候，浏览器会将If-None-Match: W/"50b1c1d4f775c61:df3" 传输给服务端，服务端拿到该ETAG，对比资源是否发生变化，如果资源未发生改变，则返回304HTTP状态码，不返回具体的资源。

收到消息要发ACK包进行确认，比如说建立tcp连接以后收到网页Http包。

RSTP：快速生成树协议（rapid spanning Tree Protocol ）：802.1w由802.1d发展而成，这种协议在网络结构发生变化时，能更快的收敛网络。它比802.1d多了两种端口类型：预备端口类型（alternate port）和备份端口类型。 STP（Spanning Tree Protocol ）是生成树协议的英文缩写。该协议可应用于环路网络，通过一定的算法实现路径冗余，同时将环路网络修剪成无环路的树型网络，从而避免报文在环路网络中的增生和无限循环。

QUIC协议：qucik udp internet connect

DNS规定，域名中的标号都由英文字母和数字组成，每一个标号不超过63个字符，也不区分大小写字母。标号中除了连字符（-）外不能使用其他标点符号。域名总体提不能超过255个字符。国家开发使用本名族语言的域名。RFC规定：多余512字节的数据使用TCP发送。

中国.cn 美国.us 日本.jp 韩国.kp

DDNS（Dynamic Domain Name Server）是动态域名服务。现在市场上很多网络摄像机需要注册一个动态域名（DDNS），然后启用UPnP或者自己做端口映射设置才能从外网访问到。不过有个广告说采用P2P技术的话就不需要任何配置。目前大部分家庭使用PPPOE拨号方式上网，每次上网获得的IP都是随机变换的，但是家里的网络监控、智能设备需要通过网络访问，每次使用都需要先知道IP非常麻烦。有了DDNS动态域名解析，我们只要到花生壳之类的域名提供商申请一个动态域名，以后只要记住域名就可以访问家里的网络，非常方便。在域名提供商处申请一个域名（会提供一个账号和密码），在路由器的动态DDNS里登陆就可以获得这个域名的解析，以后只要记住这个域名就能访问这个家庭的网络了。

浏览器、电脑本地都对DNS有缓存（浏览器的缓存时间很短很短），甚至还有DNS缓存服务器的存在，所以这时候如果将DNS进行更改会有一个生效期。

一般Local DNS Server会缓存TLD Server的ip地址，之后就可以不需要经过DNS Root Server 直接查询。

DNS包中的 authority 信息，真的是可能会启动作用（附加的可能会被使用的信息），如果能够起到作用，其实能够减少很多DNS查询。

最近 Cloudflare 宣布使用 1.1.1.1 作为 DNS，并且强调隐私保护。由于 Cloudflare DNS 支持 DNS-over-TLS 和 DNS-over-HTTPS，这使得加密 DNS 成为了热门话题。

DNS/HTTP/.... Cache Server是一种Proxy Server

网络通信协议分层

应用层：

HTTP（Hypertext Transfer Protocol 超文本传输协议，显示网页）

DNS（Domain Name System）

FTP（File Transfer Protocol

SFTP（SSH File Transfer Protocol，和FTP不一样）

SCP（Secure copy，based on SSH）

SSH （Secure Shell）

通信层： 唯一标识一条连接需要包含协议的5元组

TCP（Transmission Control Protocol 三次握手传输协议）

UDP

网络层：

IP（Internet Protocol）

ICMP（Internet Control Message Protocol，主要用于路由发送错误报告）

SFTP和SCP都是基于SSH的，所以端口还是22。

NAT是地址映射，NAPT是端口映射

traceroute 有使用两种：使用ICMP的和使用UDP的。Microsoft使用ICMP，所以win95上发出的traceRT应使用的是ICMP，最后会经过几次对目的地的ICMP Request，返回是正常的ICMP Reply；其它包括unix和cisco router都使用UDP，目标端口为高端口号，如果到了目的地就会返回一个「ICMP port unreachable」消息。

114DNS

　　114.114.114.114

　　114.114.115.115

CNNIC SDNS

　　1.2.4.8

　　210.2.4.8

阿里AliDNS

　　223.5.5.5

　　223.6.6.6

192.168.\*\*.\*\* 和172.168.\*\*.\*\* 这都是内网！

因为RFC上规定的（建议的）内网地址应该是这个范围，实际范围要更大一些：10.0.0.0

- 10.255.255.255 (10/8 prefix)

172.16.0.0

- 172.31.255.255 (172.16/12 prefix)

192.168.0.0

- 192.168.255.255 (192.168/16 prefix)

本地DNS更改是会对发出去的包有进行目的地的设置。360断网DNS修复也就是改变本地网络接口的DN，这和改路由器DNS对于校园网崩坏并没有什么软用。因为那个是本地出口DNS，除非使用电信天翼飞扬这样倒是会换一个。

将路由器里面的“忽略解析文件”勾起来，那么DNS解析就需要本机设定，路由器不提供转发，于是没有设定的linux虚拟机就像是断网了一样。其特点就是分配到了ip，但是ping 和 浏览网站都是unknown host与404 server not find。

ipv4ipv6 地址字符串表示最大长度

1 for IPV4

#define INET\_ADDRSTRLEN 16

111.112.113.114

32位IPV4地址，使用10进制+句点表示时，所占用的char数组的长度为16，其中包括最后一个NULL结束符。

2 for IPV6

#define INET6\_ADDRSTRLEN 46

ipv6共128位，IPV4共32位，128-32 = 96位，6组16进制

ipv6以下几种表示方式(还有其他缩写此处仅为可能最长长度)：

1. 0001:0002:0003:0004:0005:ffff:111.112.113.114 　　(46 = 6\*4 + 6 + 15 + 1)，6组4位，6个:，ipv4，1个NULL

2 0001:0002:0003:0004:0005:0006:0007:0008　　　　　(40 = 8\*4 + 7 + 1)，8组4位，7个:，1个NULL

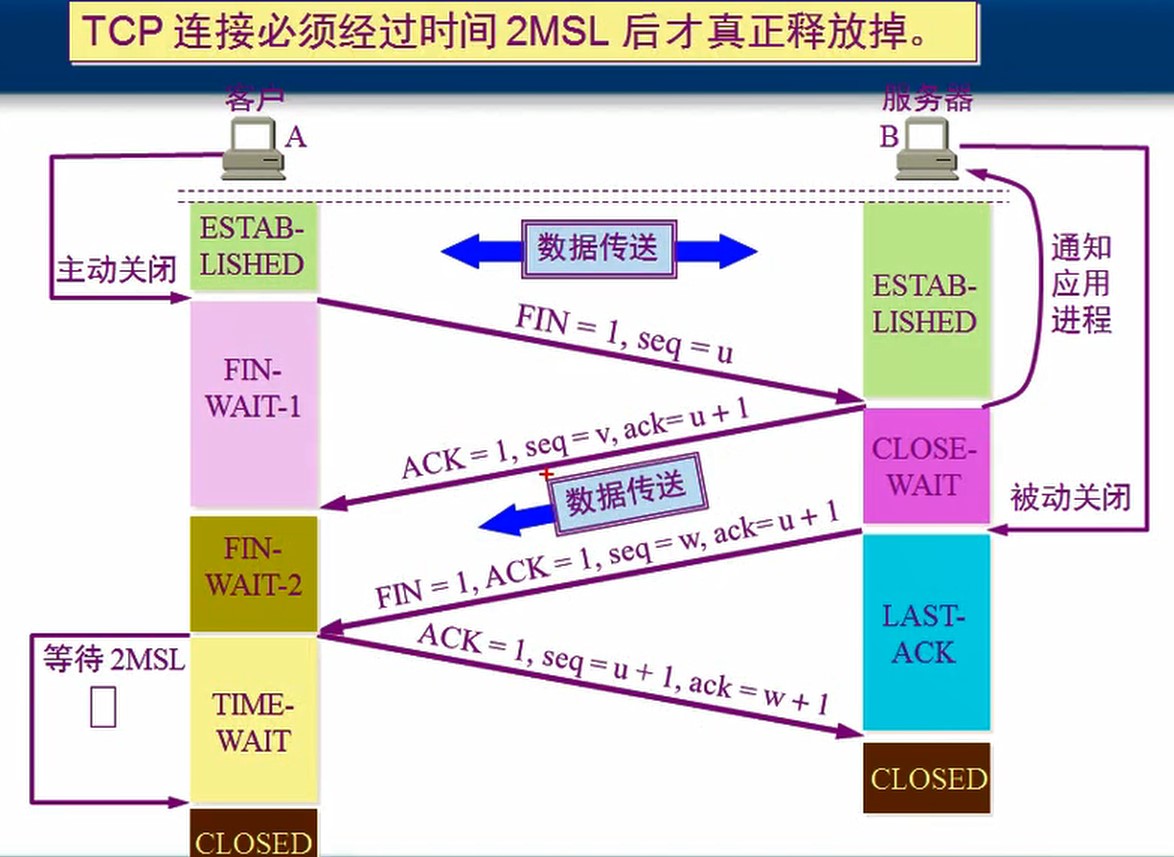
46 here is to support ipv4-mapped-on-ipv6

socket发送数据，不管对方接不接受，对方系统协议栈会临时接收，保存到缓冲区里，回复ACK。send()就是送进缓存区就函数返回。

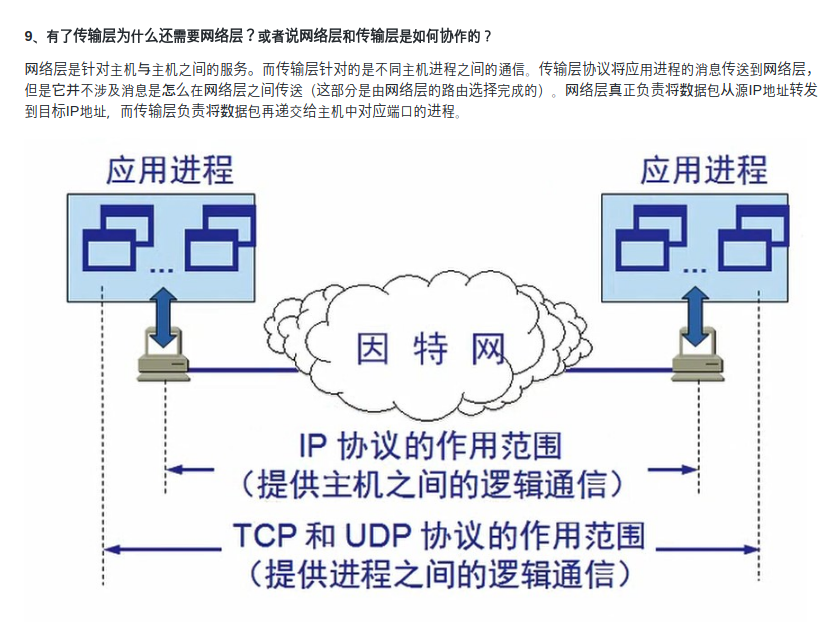
TCP三次握手建立连接：

第一次握手：主机A发送位码为syn＝1，随机产生seq number=1234567的数据包到服务器，主机B由SYN=1知道，A要求建立联机；  
 第二次握手：主机B收到请求后要确认联机信息，向A发送ack number=(主机A的seq+1)，syn=1，ack=1，随机产生seq=7654321的包；  
 第三次握手：主机A收到后检查ack number是否正确，即第一次发送的seq number+1，以及位码ack是否为1，若正确，主机A会再发送ack number=(主机B的seq+1)，ack=1，主机B收到后确认seq值与ack=1则连接建立成功。

四次挥手断开连接（time\_wait是即将关闭socket的一个状态，也就说该发送方会保持2MSL时间之后才会回到初始状态，MSL是数据包在网络中的最大生存时间。这个状态避免旧的数据包会传递给同一个五元组的新连接）。



最后一个ACK还要等待2MSL的时间，如果在2MSL内客户端都没有收到服务端的任何消息，便认为服务端收到了确认。此时可以结束TCP连接。



出现类似“该应用程序调用内存xxxxxxxx,该内存不能为只读”，的错误？

请将网卡驱动程序重装一次，一般都可解决。

IP地址换换成数字地址的方法如下

例子：

219.239.110.138

具体计算过程如下：

219\*2563+ 239\*2562+110\*2561+138\*2560=3689901706

219.239.110.138-->3689901706

转换后的3689901706即为ip 219.239.110.138的数字地址

浏览器支持，不过我没成功，chrome会因为访问google尝试去搜索而GG。

windows的发包真是可怕，直接把握操作系统给出了：

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.71 Safari/537.36

不知道哪里，肯定有，给出了屏幕分辨率。

NTP服务器【Network Time Protocol（NTP）】是用来使计算机时间同步化的一种协议，它可以使计算机对其服务器或时钟源（如石英钟，GPS等等)做同步化，它可以提供高精准度的时间校正（LAN上与标准间差小于1毫秒，WAN上几十毫秒）

ntpdc -n -c command target

Listpeers指令，列出目标NTP服务器的peers（NTP Servers）

Monlist指令，可以获取与目标NTP Server进行过同步的最后600个客户机IP。这意味着，一个很小的请求包，就能获取到大量的活动IP地址组成的连续UDP包正是由于此指令的不等价交换（1：562的报酬），UDP协议的通信模糊性（无三次握手验证），以及NTP服务器的无认证机制，使得反射型DDos攻击成为可能。包最后会有一堆00，要是没有00，那么效果是不定的，但是很可能失败。

Linux的ntpdc指令超时时间很短，因此不容易返回完整的列表。

NAT：每当来自本地局域网的设备向公网发起请求时，路由器便会详细的记录该请求是由本地局域网中，哪个设备向哪个服务发起的请求。然后，当它成功从目标服务器获得响应时，它则会将其转发回本地局域网中IP所对应的设备。这也意味着，在大多数设置中，指向路由器的未经请求的数据包将被丢弃。

NAT也相当于一个防火墙，其作用效果与云服务器的“云安全组”一样。只有内部从外部连接这一个过程session上的内容才会被允许发送，如果外部向内部连接，就会起到防火墙的作用，如果没有手动配置，就会直接被拒绝。

P2P与NAT：因为NAT如果不在session中，就会起到防火墙的作用，要想建立完整的P2P网络，每一个节点都应该起到客户端与服务器的作用，这时候就需要一个中介服务器，如果使用UDP协议，直接通过中介服务器交换目标地址与端口信息就能进行交流，直接使用这个端口向目标发送一个包，这时候NAT设备就产生了一个新的会话；如果使用TCP协议，需要开启端口重用功能。

如果一个架构符合REST原则，就称它为RESTful(Representational State Transfer)架构。我们把"资源"具体呈现出来的形式，叫做它的"表现层"（Representation），可以理解位文件格式。如果客户端想要操作服务器，必须通过某种手段，让服务器端发生"状态转化"（State Transfer）。而这种转化是建立在表现层之上的，所以就是"表现层状态转化"。

简单来说，RESTful API 是基于HTTP协议产生的一种相对简单的API设计方案，属于无状态传输。RESTful 的核心是 everything is a “resource”，所有的HTTP action，都应该是相应resource上可以被操作和处理的，而API 就是对资源的管理操作，而这个具体操作是由 HTTP action 指定的。

URL中应尽量使用名词，尽量避免使用动词；

应该尽量将API部署在专用域名之下；

https://api.example.com

如果确定 API 很简单，不会有进一步扩展，可以考虑放在主域名下；

https://example.org/api/

应该将API 的版本号放入URL，或者将版本号放在HTTP头信息中；

https://api.example.com/v1/

举例来说，有一个API提供动物园（zoo）的信息，还包括各种动物和雇员的信息，则它的路径应该设计成下面这样。

https://api.example.com/v1/zoos

https://api.example.com/v1/animals

https://api.example.com/v1/employees

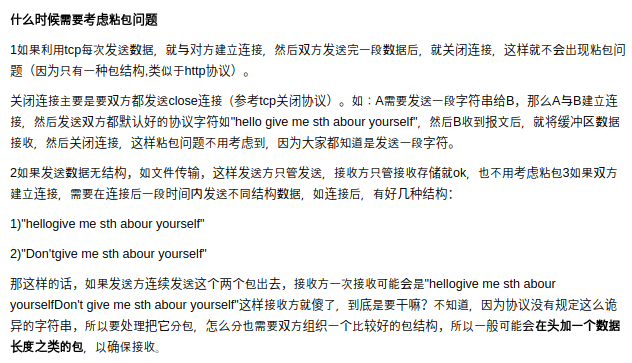
公认端口：0-1023 不过0一般不使用

注册端口1024-49151 用于松散地绑定于一些网络服务，可以把许多服务绑定到同一端口作为触发条件（一个服务把其他几个都开起来了），注册端口可以自定义。

动态、私有端口：49152-65535

Nagle算法是以他的发明人John Nagle的名字命名的，它用于自动连接许多的小缓冲器消息；这一过程（称为nagling）通过减少必须发送包的个数来增加网络软件系统的效率。Nagle的算法通常会在TCP程序里添加两行代码，在未确认数据发送的时候让发送器把数据送到缓存里。任何数据随后继续直到得到明显的数据确认或者直到攒到了一定数量的数据了再发包。

这种算法会带来“TCP粘包”的问题，



1.发送端需要等缓冲区满才发送出去，造成粘包

2.接收方不及时接收缓冲区的包，造成多个包接收

解决：

socket设置为TCP\_NODELAY即关闭Nagle的使用

TCP提供了强制数据立即传送的操作指令push，每一次数据都直接发送（相当于不使用缓冲区）

给数据格式设定良好的协议，接受方专门创建一个预处理线程分包（粘包、分包、封包、拆包区分）

在流传输中出现，UDP不会出现粘包，因为它有消息边界

tcp TIME\_WAIT出现在主动关闭方，会等待2MSL（MSL 在RFC 1122上建议是2分钟，而源自berkeley的TCP实现传统上使用30秒）

超时机制避免最后一次ACK丢失

Socket四元组，假设没有time\_wait，断开连接重连，虽然有ACK序号，还是可能存在“新连接收到旧连接报文”

出现大量TIME\_WAIT：

大量短连接，如http keep-alive: close

避免/减少TIME\_WAIT

首先服务器可以设置SO\_REUSEADDR套接字选项来通知内核，如果端口忙，但TCP连接位于TIME\_WAIT状态时可以重用端口。在一个非常有用的场景就是，如果你的服务器程序停止后想立即重启，而新的套接字依旧希望使用同一端口，此时SO\_REUSEADDR选项就可以避免TIME\_WAIT状态。

缩短MSL

ICMP是分装在IP之上的第三层。著名的有TIMESTAMP确定目标系统时间，ADDRESS MASK确定本地子网掩码

DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）通常被应用在大型的局域网络环境中，主要作用是集中的管理、分配IP地址，使网络环境中的主机动态的获得IP地址、Gateway地址（address of first-hop router for client）、DNS服务器地址等信息（name and ip）、子网掩码，并能够提升地址的使用率。DHCP分配一个IP需要完整的四步：客户端DHCP Discover、服务器DHCP Offer（提供了IP）、客户端DHCP Request（网络中没准有多台DHCP服务器）、服务器DHCP ACK（被选择的DHCP才会发出）四种都是广播包，因为只有在整个流程结束以后才会真正分配到ip。其中DHCP ACK包含了分配的ip、Gateway地址、DNS服务器地址、子网掩码等信息。

不管IP地址的租限有没有到期，但当每一次启动DHCP客户端时，都会自动利用广播的方式，给网络中所有的DHCP服务器发送一个DHCP请求信息，以便请求该DHCP客户端继续使用原来的IP地址及其配置。租期到一半、87.5%的时候都会尝试DHCP request，如果87.5%的时候没有被ACK就会停止使用这个IP。

当续租失败时，DHCP服务器将会给该DHCP客户端发送一个DHCPNACK信息，DHCP客户端在收到该信息后，说明该IP地址已经无效或被其他的DHCP客户端使用。

总体来说，DHCP Snooping具有以下两方面功能：

保证DHCP客户端从合法的DHCP服务器处获取IP地址。

记录DHCP客户端IP地址与MAC地址的对应关系。

DNS在进行区域传输的时候使用TCP协议,其它时候则使用UDP协议

OpenSSL 是一个安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议

TLS协议，强制https连接

SSL可能还需要客户端证书（通常金融机构）

https使用ssl+HTTP，但是在证书颁布的对象并不一定是目标域名，ip.cn的证书上写着颁布给“sni.cloudflaressl.com”，颁发者来自“CloudFlare Inc ECC CA-2”。 一般而言，由 CA 业界发出的数字证书，有别于国内浏览器业者比对域名信息等方式，采取更为严格的企业及所有权验证，为电商环境树立更为可信的运作环境。

ICMP直接封装在IP包内，所使用IP协议号为1。Ping用到的是ICMP协议。不是端口。ICMP没有端口号可言。

简单网络管理协议（SNMP），是基于TCP/IP协议族的网络管理标准，是一种在IP网络中管理网络节点（如服务器、工作站、路由器、交换机等）的标准协议。有v1，v2，v3这几个版本（对于实现SNMPvl和SNMPv2的设备，由于缺少鉴别和加密机制）。几乎所有的网络设备和网络操作系统都支持SNMP。

端口：UDP161（服务端）、162（客户端） #DHCP UDP 67（服务端） 68（客户端）

一套完整的SNMP系统主要包括管理信息库（MIB）、管理信息结构（SMI）及SNMP报文协议。

MIB：任何一个被管理的资源都表示成一个对象，称为被管理的对象。MIB是被管理对象的集合。它定义了被管理对象的一系列属性：对象的名称、对象的访问权限和对象的数据类型等。每个SNMP设备（Agent）都有自己的MIB。MIB也可以看作是NMS（网管系统）和Agent之间的沟通桥梁

使用库里的命令编号进行查询（树形分支查询），最后加上实例编号。

.1.3.6.1.2.1.25.4.2.1.2 列出系统进程

.1.3.6.1.4.1.77.1.2.25.1.1 列系统用户列表

.1.3.6.1.4.1.77.1.4.1.0 列出域名

.1.3.6.1.2.1.25.6.3.1.2 列出安装的软件

.1.3.6.1.2.1.1 列出系统信息

由于SNMP主要采用UDP传输，很容易进行IP源地址假冒。大多数SNMP设备接收来自网络广播地址的SNMP消息，攻击者甚至可以不必知道目标设备的lP地址，通过发送广播SNMP数据包达到目的。UDP是面向无连接的，那么消息流可能被攻击者进行恶意的重组、延迟和重放。

windows安装snmp，运行appwiz.cpl-》添加组件-》管理和监视工具 里面的全部安装，然后services.msc里面就会有snmp。在snmp service的配置里面有一个“陷井”，当snmp作为客户端使用的时候就会使用它，配置好团体名字（comunity string）、陷井目标，就会自动把一些信息主动（隔一段时间发或者触发容量不够等）发给监控的服务器（和mqtt客户端有点像）。默认情况下，允许接受所有的主机snmp数据包，而且默认团体是只读的public。

http://blog.csdn.net/shanzhizi/article/details/11606767

git clone ssh://git@example.com/path/to/git-repo.git 意外发现确认了确实可以用ssh来传送文件。

VNC（Virtual Network Computing），为一种使用RFB协议的屏幕画面分享及远程操作软件。此软件借由网络，可发送键盘与鼠标的动作及即时的屏幕画面。

RFC描述：如果一个关闭的端口收到数据包没有SYN、ACK、RST，那么就会发送RST作为响应。开放端口则不会理睬。

Xmas tree TCP端口扫描原理：

发送了FIN+PSH+URG 这种非常规数据包

NULLTCP端口扫描原理：

发送了什么标志位都没有的非常规数据包。

集线器会把流量广播给所有端口。交换机则是根据arp表来发送。

RTSP发起/终结流媒体（TCP）、RTP传输流媒体数据（UDP） 、RTCP对RTP进行控制，同步。

vsftpd服务，一个使用广泛的FTP服务

ingreslock 是个很古老的危险服务，可以很简单地实现入侵，telnet 192.168.1.152 1524

无类别域际路由选择（Classes inter domain routing，CXIDR）格式来定义IP地址段。

windows系统改注册表的话也能变成TTL值

RFC 1918私有网络地址分配 (Address Allocation for Private Internets)

林根域： 林就是在多域情况下形成的森林,根表示基础,其他在此根部衍生

域已经搭建完成，主域控会生成一个krbtgt账号 他是Windows活动目录中使用的客户/服务器认证协议，为通信双方提供双向身份认证

根域有一个很显著的特征，在根域的DC上会有两个特殊的组：Enterprise admins、Schema admins

鉴权协议又被成为挑战---认证模式：信道可以被侦听，但是不能被篡改。

SPDY（读作“SPeeDY”）是Google开发的基于TCP的传输层协议，用以最小化网络延迟，提升网络速度，优化用户的网络使用体验。SPDY并不是一种用于替代HTTP的协议，而是对HTTP协议的增强。新协议的功能包括数据流的多路复用、请求优先级以及HTTP报头压缩。传闻能将页面加载时间减少50%。互联网工程任务组（IETF）对谷歌提出的SPDY协议进行了标准化，于2015年5推出了类似于SPDY协议的 HTTP 2.0 协议标准（简称HTTP/2）。谷歌因此宣布放弃对SPDY协议的支持，转而支持HTTP/2。此外，著名的开源HTTP服务器软件 Nginx 也于2015年9月移除了对SPDY的支持，转而支持HTTP/2。

移动解析（HttpDNS）基于Http协议向腾讯云的DNS服务器发送域名解析请求，替代了基于DNS协议向运营商Local DNS发起解析请求的传统方式，可以避免Local DNS造成的域名劫持和跨网访问问题，解决移动互联网服务中域名解析异常带来的困扰。

telnet 192.168.1.235

要求用与登录的账户必须设置密码，否则就会登陆错误。

远程桌面是全新的桌面，如果是远程协助那么就会是共享桌面。

常见的邮件协议包括：

（1）SMTP：简单邮件传输协议，用于发送电子邮件的传输协议；

（2）POP3：用于接收电子邮件的标准协议；下载的时候,同时删除保存在邮件服务器上的邮件。无状态

（3）IMAP：互联网消息协议，是POP3的替代协议。有状态

这三种协议都有对应SSL加密传输的协议，分别是SMTPS，POP3S和IMAPS。

如果连接URL使用RTSP，RTSP会自动协商内容的最佳传送机制。 然后它指示RTP协议使用UDP传送流式传输内容，或者在不支持UDP的网络上使用基于TCP的协议。

OPTIONS rtsp://192.168.1.30:8554/test RTSP/1.0\r\n

RTSP/1.0 200 OK\r\n

DESCRIBE rtsp://192.168.1.30:8554/test RTSP/1.0\r\n

RTSP/1.0 503 Service Unavailable\r\n

使用TCP连接，但是使用UDP/RTP来发送数据

网线制作：橙白，橙，绿白，蓝，蓝白，绿，棕白，棕。

M-JPEG over HTTP：将每一张图片根据Random String分离到一个个HTTP包当中。如果你的封装是成功的话，那么 QuickTime or VLC是可以查看的。

ICANN接管了IANA和其他组织根据与美国政府签定的合同而承担的一部分互联网管理职能

ICANN的下属机构很多，其中三个：

Address Supporting Organization ASO www.aso.icann.org 管理IP分配：ASO->RIR区域Internet注册机构->ISP->LIR 本地注册机构

Generic Name Support Organization GNSO 通用名称支持组织 管理通用顶级域名

Country Code Domain Name Supporting Organization CNNSO 管理国家顶级域名

http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml ipv4分配方案

https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml 已注册的端口号

https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml 已分配的协议号

特殊用途的ip：http://www.rfc-editor.org/rfc/rfc3330.txt

Address Block Present Use Reference

---------------------------------------------------------------------

0.0.0.0/8 "This" Network [RFC1700, page 4]

10.0.0.0/8 Private-Use Networks [RFC1918]

14.0.0.0/8 Public-Data Networks [RFC1700, page 181]

24.0.0.0/8 Cable Television Networks --

39.0.0.0/8 Reserved but subject

to allocation [RFC1797]

127.0.0.0/8 Loopback [RFC1700, page 5]

128.0.0.0/16 Reserved but subject

to allocation --

169.254.0.0/16 Link Local --

172.16.0.0/12 Private-Use Networks [RFC1918]

191.255.0.0/16 Reserved but subject

to allocation --

192.0.0.0/24 Reserved but subject

to allocation --

192.0.2.0/24 Test-Net

192.88.99.0/24 6to4 Relay Anycast [RFC3068]

192.168.0.0/16 Private-Use Networks [RFC1918]

198.18.0.0/15 Network Interconnect

Device Benchmark Testing [RFC2544]

223.255.255.0/24 Reserved but subject

to allocation --

224.0.0.0/4 Multicast [RFC3171]

240.0.0.0/4 Reserved for Future Use [RFC1700, page 4]

可能每一个路由接口分别使用了不同的ACL（访问控制列表，主要用于过滤，可以对流量进行分类ftp、vpn）。inbound interface package先经路由表判断-》选择端口以后-》ACL判断，检查顺序是从上往下逐个进行匹配，这就有性能与规则顺序的关系。通过判断ACL的编号来判断是标准的还是功能扩展的，扩展的ACL能进行更加精确地指定。或者根据cisco IOS的版本。ACL的隐含规则是拒绝所有,所以一定会有一条permit。ACL中的通配符，恰好与子网掩码的书写方式相反，如果是0.0.0.0那么可以指定不写，有any这种缩写表示0.0.0.0 255.255.255.255，前面是IP，后面是允许全部的通配符。

Traceroute在送出UDP datagrams到目的地时，它所选择送达的port number 是一个一般应用程序都不会用的号码（30000 以上），所以当此UDP datagram 到达目的地后该主机会送回一个「ICMP port unreachable」的消息，因为端口开着便不返回消息，而当traceroute 收到这个消息时，便知道目的地已经到达了。所以traceroute 在Server端也是没有所谓的Daemon 程式。这是默认原理，但是linux可以开更多姿势ICMP ECHO、TCP SYN。

SFTP=SSH加密的FTP；FTPS使用了较强密码或者基于SSL证书的FTP

telnet和tftp都是不安全的，telnet是明文，tftp（快餐式文件传输协议）不仅仅是明文只要知道了路径就可以进行下载，甚至遗留下错误的文件系统访问控制列表。MTFTP是多点的TFTP服务，应用在windows无盘工作站的服务端。

MSRPC : tcp 135 linux的111或者Solaris 32771端口上rpc也没有身份验证

在MSRPC实现中，可以采用如下的网络传输协议：①ncacn\_ip\_tcp：TCP/IP传输协议，使用TCP135端口②ncadg\_ip\_udp:UDP/IP传输协议，使用UDP135端口③ncacn\_np:SMB命名管道传输协议，使用SMB的TCP139或445端口④ncalrpc:本地网络RPC传输⑤ncacn\_http:HTTP传输协议，基于IIS服务，缺省配置在TCP593端口。支持远程过程调用的服务进程启动时，需要注册自己的服务访问点，接口标识符以及服务版本号，一个特殊的RPC服务

这个端口开放的一个目的，使远程用户能够OutLook连接到使用Exchange Server的服务器上，使用微软的一些程序需要开着这个135端口。

SSH: (安全外壳协议)

SSH是由客户端和服务端的软件组成的，有两个不兼容的版本分别是：1.x和2.x。 用SSH 2.x的客户程序是不能连接到SSH 1.x的服务程序上去的。OpenSSH 2.x同时支持SSH 1.x和2.x。

连接过程类似于HTTPs，多一步对ssh版本的验证协商，https://blog.csdn.net/qwertyupoiuytr/article/details/71213463

NetBios名字服务：udp 137 广播

本来这只是一个IBM创建的用于Local Area Network间计算机进程交流的协议（NetBIOS 的开放，意味着你的硬盘可能会在网络中成为共享。），后来发展为可以over TCP/IP，这时候也叫NetBT或者NBT。在Windows操作系统中，默认情况下在安装TCP/IP协议后会自动安装NetBIOS。windows工作组一般都叫WORKGROUP。网络->选择 TCP/IP 协议，点击“属性” 这里可以关闭。绝大部分情况下，ip与这个协议中的名字是等价的，ping命令。禁用后，局域网内计算机不可以用双斜杠访问你，保护了计算机的安全。但是如果DHCP服务器没有开启NetBIOS，那么主机就要自己开启NetBIOS，windows里面的默认配置在这情况下就会开启NetBIOS，否则你是会断网的。

域(Domain)是Windows网络中独立运行的单位，域之间相互访问则需要建立信任关系，每一个域都有一个域控制器DC（备份域控制器就是BDC。DC有着自己专门的职责，比如说参与域用户登录的身份验证）,域控制器是windows网路身份证书的保管者。域的出现方便了网管，可以对各个成员批量管理。域的管理靠DNS。域用户，可以登陆域内任何主机，包括DC。然后就需要登录DC，尝试获得域的管理员权限。DC会有一个群组策略控制台。

域中有一个叫做LDAP（Lightweight Directory Access Protocol）的成分，负责将系统中的账号信息数据库库存储以及对域账号的验证（使用域账号登入极速你以后不需要再登录系统后再一次输入密码，进入系统；有多个域账号或者密码，使员工任何一个登录就能获取自己的权限）。Microsoft称之为活动目录active directory AD。只要能通过ldap建立一个会话就能直接获取所有的用户名。dn:CN=John Doe,OU=Texas,DC=example,DC=com，右边的DC是根节点，在配置文件/etc/openldap/ldap.conf。389非加密端口 636加密端口。

禁用SMB和NBNS对老版本widows来说是一种噩梦。

活动目录（Active Directory）是面向Windows Standard Server、Windows Enterprise Server以及 Windows Datacenter Server的目录服务。（Active Directory不能运行在Windows Web Server上，但是可以通过它对运行Windows Web Server的计算机进行管理。）Active Directory存储了有关网络对象的信息，并且让管理员和用户能够轻松地查找和使用这些信息。Active Directory使用了一种结构化的数据存储方式，并以此作为基础对目录信息进行合乎逻辑的分层组织。”活动目录（AD）是在内网中运行域控、DNS等等的基础，域控只不过是一台控制机器，它和DNS等等一切服务都是运行在AD之中的。

NetBios数据报服务：udp 138 主要用于浏览器服务通知。这些消息用于在用户桌面系统中构建网上邻居。

NetBios会话服务: tcp 139 文件和打印机服务构成了NetBIOS会话服务产生的通信的主要部分。在“网络”属性-》更改高级共享设置-》启用文件和打印机共享 从而变成一台提供该服务的的服务器。统称SMB服务器。

WINS解析：

WINS是Windows Internet Name Server（Windows网际名字服务）的简称，是微软开发的域名服务系统。NetBIOS是IBM开发的内网域名服务。WINS为NetBIOS名字提供名字注册、更新、释放和转换服务，这些服务允许WINS服务器维护一个将NetBIOS名链接到IP地址的动态数据库，大大减轻了对网络交通的负担。之所以NetBIOS对大网络会有负担是因为，每一台计算机都是采取广播的方式，所以报文会很多，而这中服务只需要维护一个服务器上的表就好。

lmhosts文件：

一个纯文本文件，将NETBIOS名和IP地址对应起来，功能类似于DNS。也是转化名字的一种途径。

WINS以及lmhosts文件在适配器的属性-》高级里面有设置。

SMB： 用于文件打印共享的服务Server Message Block。本质上空连接就是一个SMB的匿名会话（将Everyone包含在匿名访问令牌中）

SMB由微软开发，后来linux引入。据说这是微软历史上出现安全问题最多的协议。

SMB1 2000/xp/2003

SMB2 Visfa SP1/2008

SMB2.1 7/2008 R2

SMB3 8/2012

在windows NT中，SMB基于NBNS（NetBIOS）实现。在Windows2000/xp/2003中，SMB除了基于NBT的实现，还可以直接通过445端口实现。默认开启的只有135，445。

当windows2000以上的操作系统作为SMB文件服务器时，就面临着选择139或者445端口了。下面的情况确定会话使用的端口：

1、如果SMB文件服务器启用了NBT，那么客户端连接SMB文件服务器的时候将同时访问139和445端口，如果从445端口得到回应，那么客户端将发送RST到139端口，终止这个端口的连接，接着就从445端口进行SMB的会话了；如果没有从445端口而是从139得到回应，那么就从139端口进行会话；如果没有得到任何回应，那么SMB会话失败。

2、如果SMB文件服务器禁用了NBT，那么客户端连接SMB文件服务器的时候就将只从445端口进行连接。

Samba：

linux下除了使用SMB可以进行文件分享，还有Samba（/etc/smb.conf）

BGP： tcp 179 Border Gateway Protocol

边界网关协议（路由协议），这是一个必须开启的协议。BGP是自治系统AS间的路由协议，充当BGP发言人。BGP交换包含全部AS path的网络可达性信息。 BGP发言人不一定是边界网关路由器。只有具备超过一个上行链路的网络才使用BGP，也就是比较大的组织。

BGP只使用IP和ASN号码（16位整数，向ip发布机构购买，作为该组织在互联网上的标语，可以认为就是ip）。

ssh和rdp，Disconnect != Log Off。对于rdp，在关机按钮那里有选择。只有log off才是对会话的释放，于是与会话相关的资源就会被释放。

端口敲门：knockd服务。公有云通常可以从互联网的任何地方都访问到，这使得他们容易受到各种各样的SSH攻击。这当然可以通过禁用SSH服务来防止，但是这样也无法使用SSH来管理服务器了。端口敲门是一项和防火墙一起使用的技术，在收到一个特殊的端口敲击顺序后再开启指定的端口。它使用数据头同服务器交换隐秘信息，打开一个指定的端口。隐秘敲门暗号被封装成一组有序的端口，并且需要发送SYN包来验证有效性。当对任意端口的SYN包的正确顺序被接收到时，客户端才会被允许访问该端口。这种方法很有局限性，因为这是一种隐晦式安全的实现，并且攻击者仍然可以暴力攻击SYN包需要发送的端口顺序。

ipsec与L2TP，它们是vpn的一种实现标准，分别取代了点对点隧道协议PPTP与第二层转发L2F（Layer 2 Forwarding）。

ipsec：工作在IP层，处理了可靠性和分片的问题。不像SSL不能支持UDP，IPSec能支持所有的协议，但是这也就意味着更多的开销，不过想来应该不怎么样。IPsec是一个安全协议的框架，它结合多种安全的协议，能够提供

数据机密性、完整性和验证等功能。

L2TP：Layer 2 Tunneling Protocol 是一个数据链路层协议。其报文分为数据消息和控制消息两类。不过也是点对点连接。L2TP不保证数据消息的可靠投递，若数据报文丢失，不予重传，不支持对数据消息的流量控制和拥塞控制。PPTP和L2TP都使用PPP协议对数据进行封装，然后添加附加包头用于数据在互联网络上的传输，但是PPTP是ip层的，L2TP是数据链路层的。

VPN是站点到站点或者客户端到站点，站点那里的设备叫做VPN网关。site-to-site 这种vpn一般都是一直连接着，而不是即时拨号接入。

原始IP数据包：IP头 TCP头 数据

传输模式下的IP数据包: IP头 IPsec头 TCP头 数据

隧道模式下的IP数据包: IP头 IPsec头 IP头 TCP头 数据

PPP协议：

1. 链路建立阶段(由LCP协议协商)

2. 验证阶段(可选)

存在2种PPP验证协议：PAP和CHAP以及EAP框架协议

PAP明文两部验证就好了，所以不能防止嗅探与重放攻击。

CHAP 客户端请求-》服务器发出一个挑战-》客户端使用已有的密码进行md5加密，在作为响应返回-》服务器最后给出结果

3. 网络层协议阶段(由NCP协议协商)

ftp服务端软件可以控制支持的命令，控制指令是通过服务器的TCP 21号端口发送。ftp的运行分为主动模式(port模式)以及被动模式(passive 模式)，主动于被动都是相对于服务器而言的。在过去，客户端缺省为active(port)模式；近来，由于Port模式的安全问题，许多客户端的FTP应用缺省为Passive模式。如果是Port模式，那么创建的服务器数据传输端口就会在20端口，如果是passive模式，那么服务器的数据输出端口是不定的。

FTP 的PORT指令可以指定别的服务器的ip以及端口，从而进行内网上面的扫描。

带外数据(Out of Band, OOB)方案就是将控制流以及数据流进行分离；对应的叫做in-band。并且认为传输数据并不是主要的数据，主要的数据对ftp而言就是文件的传送。

通过抓包发现浏览器访问ftp：

1.ip.src==128.119.245.12 or ip.dst==128.119.245.12

2.统计-》对话-》tcp 知道了端口以后可以进行端口过滤

3.找到合适的包，使用“跟踪流”，就能看到在这个会话流中的信息，比如说控制信息流

220 (vsFTPd 3.0.2)

USER anonymous

331 Please specify the password.

PASS mozilla@example.com 在获取数据之前，FTP客户端必须发送一个用户名到服务器。这一步由FTP命令USER来完成。FTP服务器响应一个报文，并通知客户端发送完整的电子邮件地址作为密码（但许多客户端代之发送一个标识客户端软件的字符串作为密码，一些FTP服务器检查密码是否是电子邮件格式，但大部分不会这么做。）

230 Login successful.

SYST

215 UNIX Type: L8

FEAT

211-Features:

EPRT

EPSV

MDTM

PASV

REST STREAM

SIZE

TVFS

UTF8

211 End

OPTS UTF8 ON

200 Always in UTF8 mode.

PWD FTP服务器保持对用户目录位置的跟踪。用户还可以请求从服务器获取文件或向服务器存储文件，对这些请求的解释与当前工作的目录有关。

257 "/"

TYPE I

200 Switching to Binary mode.

PASV

227 Entering Passive Mode (128,119,245,12,137,248).

SIZE /cs591h

550 Could not get file size.

MDTM /cs591h

550 Could not get file modification time.

RETR /cs591h

550 Failed to open file.

PASV

227 Entering Passive Mode (128,119,245,12,160,64). 这个包FTP层给出了Passive Port，之后客户端就会尽快建立tcp连接，用于传输数据

CWD /cs591h

250 Directory successfully changed.

LIST

150 Here comes the directory listing.

226 Directory send OK.

PASV

227 Entering Passive Mode (128,119,245,12,235,122). 这个包FTP层给出了Passive Port，之后客户端就会尽快建立tcp连接，用于传输数据

SIZE /cs591h/rfc821.txt.Z

213 36872

MDTM /cs591h/rfc821.txt.Z

213 19940210030404

RETR /cs591h/rfc821.txt.Z

150 Opening BINARY mode data connection for /cs591h/rfc821.txt.Z (36872 bytes). 从这里文件开始传输

226 Transfer complete. 服务器通知传输完毕

Motion JPEG:

Software and devices using the M-JPEG standard include web browsers, media players, game consoles, digital cameras, IP cameras, webcams, streaming servers, video cameras, and non-linear video editors.

IP cameras:

Many network-enabled cameras provide M-JPEG streams that network clients can connect to. Mozilla and Webkit-based browsers have native support for viewing these M-JPEG streams.

M-JPEG over HTTP:

HTTP streaming separates each image into individual HTTP replies on a specified marker. HTTP streaming creates packets of a sequence of JPEG images that can be received by clients such as QuickTime or VLC.

In response to a GET request for a MJPEG file or stream, the server streams the sequence of JPEG frames over HTTP. A special mime-type content type multipart/x-mixed-replace;boundary=<boundary-name> informs the client to expect several parts (frames) as an answer delimited by <boundary-name>. This boundary name is expressly disclosed within the MIME-type declaration itself.

The TCP connection is not closed as long as the client wants to receive new frames and the server wants to provide new frames. Two basic implementations of a M-JPEG streaming server are cambozola and MJPG-Streamer. The more robust ffmpeg-server also provides M-JPEG streaming support.

Email 相关的四种协议：IAMP4、POP3、HTTP、SMTP

MIME Multipurpose Internet Mail Extensions，默认邮件中只允许包含ASCII码第一位为0的字符，使用这个以后可以设置自己的编码，从而通过base64等编码传输中文等。同时这个扩展提供了对附件内容、相关图片的支持。

MIME-Version: 1.0

MIME标准

Content-Type: 指明数据的类型;类型/子类型 （不清楚Contente-Description存在的意义）

Content-Transfer-Encoding:base64

Content-ID：报文内容标识号

Content-Description: Description of MIME message。老师ppt上面写的是指明报文是文字，照片还是对话，但是我觉得是加上自己想要表达的内容，也就是说这个部分是给人看的。

content-type: multipart/alternative text 正文和HTML正文选择其一

content-type: multipart/mixed 邮件带有附件

content-type: multipart/related 文档的多个部分相关联,描述正文与其相关的图片

HTTP协议中有一个multipart/form-data头部。

我本一直觉得HTTP一个包与一个包之间一定是分开发送的。但是建立了TCP连接以后，应用层的包就是靠识别来自己分开。不过真正的问题是一个请求对应一个响应包，如果返回了两个那么一般的操作应该是第二个被无视掉。

HTTP1.0/1.1/2.0 之间的区别：<https://blog.csdn.net/linsongbin1/article/details/54980801>

SSO英文全称Single Sign On，单点登录。SSO是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统。

DHT全称为分布式哈希表(Distributed Hash Table)，是一种分布式存储方法。在不需要服务器的情况下，每个客户端负责一个小范围的路由，并负责存储一小部分数据，从而实现整个DHT网络的寻址和存储。从而构成种子市场。下载者在下载的同时，为其它人提供上载

使用安全RTP（sRTP）或者使用TLS、SSL、IPsec等传输过程中的加密支持来防止RTP流量被窃听解析。

隧道拆分：ss作为系统的局部代理的时候。并不是所有的流量都走ss过。

NAT：

最简单的NAT就是将内网地址与一个公网地址对应。使用不同端口来区别内容实际上是PAT（port adress translation）

一个Basic NAT映射一个内在的私有IP地址到一个公网IP地址，但当数据包穿过NAT时，不更换它的TCP/UDP端口号。Basic Nat通常是只用在一些具备公共IP地址池的NAT上，通过它可以地址绑定，即代表一台内部主机。

Full Cone NAT: 在给一个新的外部会话建立了一个公共/私有的端口绑定后，一个full cone NAT就可以通过这个公共端口从公网上的任何外部端点接收数据通讯了。

Restricted Cone NAT：限制外网主机只能够向先前已经向它发送过数据包的内网主机发送包。

Port Restricted Cone NAT：比上面再加上一个端口号的限制。

Symmetric NAT：如果同一台主机使用相同的源地址和端口号发送包，但是发往不同的目的地，NAT将会使用不同的映射。此外，只有收到数据的外网主机才可以反过来向内网主机发送包。

无奈的是我的路由器的有线与无线之间处于隔离状态，ping都接不起来。主机ping不到无线设备，无线设备之间可以互相ping，虚拟机处于连路由器都ping不到状态。

Neighbor Discovery Protocol NDP

NDP用来发现直接相连的邻居信息，包括邻接设备的设备名称、软/硬件版本、连接端口等，另外还可提供设备的ID、端口地址、硬件平台等信息。ARP replaced by NDP in IPv6。

broadband技术即宽带技术即ADSL技术。

银行国家这种传输数据都是使用专线(租或者自己建立)。

没想到无线的Ether包真的是其中包含3-4 Address，其实只需要两个就好，但是可能之前有原因导致不得不这样子兼容。

NFS在文件传送或信息传送过程中依赖于RPC协议，所以NFS可以算是RPC的一个程序。输出目录是指NFS系统中需要共享给客户机使用的目录。metasploiotable2 中输出目录设置为根并且可以rw就导致能够将sshkey直接加入验证文件。相关守护进程如下：

nfsd：它是基本的NFS守护进程，主要功能是管理客户端是否能够登录服务器；

mountd：它是RPC安装守护进程，主要功能是管理NFS的文件系统。当客户端顺利通过nfsd登录NFS服务器后，在使用NFS服务所提供的文件前，还必须通过文件使用权限的验证。它会读取NFS的配置文件/etc/exports来对比客户端权限。

portmap：主要功能是进行端口映射工作。当客户端尝试连接并使用RPC服务器提供的服务（如NFS服务）时，portmap会将所管理的与服务对应的端口提供给客户端，从而使客户可以通过该端口向服务器请求服务。

为了使NFS服务器能正常工作，需要启动portmap和nfs两个服务，并且portmap一定要先于nfs启动。

WebDAV是一组基于超文本传输协议的技术集合，有利于用户间协同编辑和管理存储在万维网服务器文档。通俗一点儿来说，WebDAV 就是一种互联网方法，应用此方法可以在服务器上划出一块存储空间，可以使用用户名和密码来控制访问，让用户可以直接存储、下载、编辑文件。通常在国外网盘服务中使用。如果是私人使用就作为文件服务器。apache在配置文件中可以直接开启这个功能，之后apache的标语改变成Apache/2.2.8 (Ubuntu) DAV/2 Server，多了个DAV/2。

计算机传输文件类型有：

CS（Client-server）

P2P

CS以及P2P的混合（Skype、QQ、迅雷）

服务器响应并发请求有两种实现方案：重复服务器方案和并发服务器方案：

Concurrent Server（并发服务器）：服务器是可以同时执行多个进程的，并发服务器在同一时刻可以处理多个客户的连接，实现主要分三种方式：多进程并发服务器，多线程并发服务器，I/O复用并发服务器

Iterative Server（重复服务器）：看样子应该就是使用了多个一样的服务器

streaming multimedia协议有HTTP以及RTP。

Internet telephony以UDP实现。

HTTP本身是无状态的协议

Web应用=http协议+session、cookies等状态机制+其他辅助的机制。

RTT: time for a small packet to travel from client to server and back 在计网的题目中经常会说Nonpersistent、persistent、Nonpersistent with 5 parallel connections 三种情况然后计算时间的关系

Without pipelining: the client issues a new request only when the previous response has been received 正是短连接

With pipelining: the default mode of HTTP 1.1 正是长连接（HTTP/1.0只允许短连接）

两层c/s模型：

文件服务器

应用服务器也可以有这种架构（其实是简单的数据请求服务器，比如说专门返回json的数据），这时候的特点是服务器与客户端共同完成对应用程序和数据需求的处理。将一个应用程序分成若干个部分，由客户机和服务器分别执行、协同工作。通常在服务器上配置的是一个数据库系统，如ORACLE、SYBASE。由客户机负责向服务器发出“应用和数据请求”，由服务器根据请求的内容，完成应用处理和数据操纵，然后将处理结果返回给客户机。

三层C/S模型：

客户端<->服务器上面的应用<->后台数据库

客户端不是直接请求数据而是请求网页，但是应用程序自动向后台查询，因为有cookie等信息。

B/S模型类似。

reliable data transfer protocol (rdt 链路层协议)： 了解就能知道如果自己封装udp做到9应用层上的可靠传输

知道的版本有rdt1.0,rdt2.0,rdt2.1,rdt2.2,rdt3.0

学习网站：http://blog.csdn.net/qq\_33936481/article/details/53152903

rdt 1.0版本直接假设底层使用了可靠传输：没有bits error，同时不存在分组的丢包。

rdt2.0 版本 增加了ACK/NAK 机制。既然是基于1.0版本，所以依旧是认为不存在分组的丢包（可能发生bits error）。

rdt 2.1 增加序号机制，用于处理接收到被破坏的ACK/NAK包，因为收到破坏的包以后不知道到底是不是应该进行重传，如果贸然进行重传就可能引起“重复分组”的问题，解决办法是指定序号以后重发，接收端于是能够判断这个包是不是重复的。通过序号表明这是不是一个重复的包，即通过序号进行两个相邻包之间内容的区分。

rdt2.2 是对2.1 不必要部分的简化，没有必要传输NAK，再不会发生丢包的情况下，等待ACK0时候接收到的是ACK1。直接通过ACK来判断，多个重复的同序号ACK相当于一个NAK。减少协议实现算法的复

rdt3.0 对丢包增加了考虑，增加了倒数计时器。

tcp中使用的ACK包被简化了，像ftp数据传输的时候，多个数据包你只需要一个ACK进行确认就好。

TCP的差错恢复方法（Pipeline Protocols）：GBN 以及 SR。都是滑动窗口协议，收到一个ACK就会滑动发送方的窗口。 引入这两种是为了能够解决RDT中的等待问题，做到能够连续多个发包。

学习网址：http://blog.csdn.net/macdroid/article/details/49070185

GBN Go-Back-N：从出错处重发已发出过的N个帧。适合用于需要一定时间内传输大量数据的情景，比如说ftp传输文件。GBN序号访问受到窗口的大小限制。接收方接收的时候每一次返回是接收到的最大序号。后退N帧ARQ就是从出错处重发已发出过的N个帧。接收窗口=1，没有接受缓存。

SR选择重传： 只会重传错误的分组。所以每一个分组都会有自己的计时器以及自己的ACK。与GBN相比，需要的序号范围也变得大了，SR的序号需要保证一定时间内不会重复就好，倒也是可以重用的。接收窗口大于1。

TCP为了实现网络通信的可靠性，使用了复杂的拥塞控制算法，建立了繁琐的握手过程以及重传策略。由于TCP内置在系统协议栈中，极难对其进行改进。

tcp 的 seq next 序号=seq序号+TCP Segment Len，TCP握手结束的时候Seq与Ack number都为1。

UDP还支持一个端口与多个端口“同时”进行通信，因为socket不会被绑定。广播的 时候一定只能使用udp。

蠕虫流 量的特点：

存在对内网ipo中端口的扫描，所以发包的数量会远远大于收包的数量。使用流量关于时间的统计分析就容易看到。

蠕虫病毒是程序，而程序的发包行为是固定的。

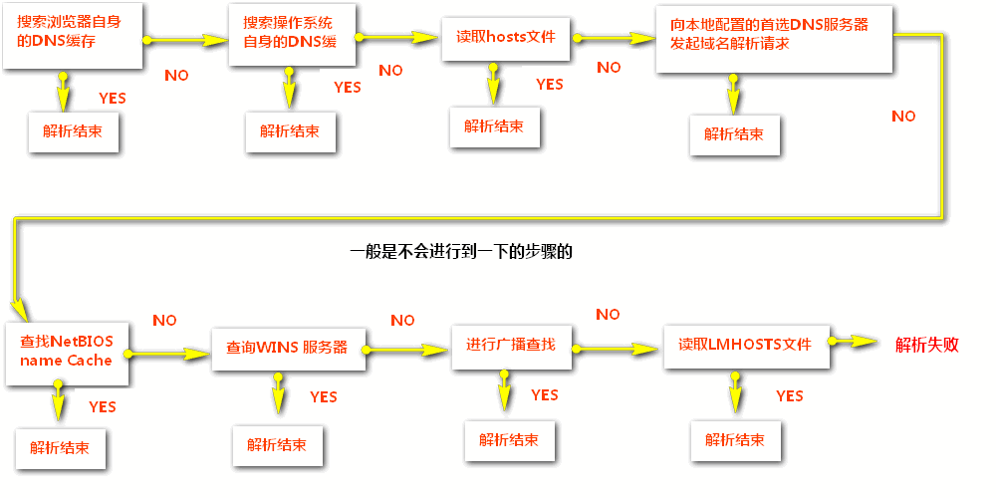
wake on lan：

一种电源管理功能，如果存在网络活动，则允许设备将操作系统从待机或休眠模式中唤醒。就是通过LAN口将电脑唤醒。这意味着这时候电脑还是在接收与处理数据包的。所以这种功能需要网卡提供支持。可以设置唤醒password。

设置：d (disabled), p (PHY activity), u (unicast activity), m (multicast activity), b (broadcast activity), a (ARP activity), and g (magic packet activity)，Ignore

子网划分：掩码为 255.255.255.252 时 只可以划分2台主机 比如 192.168.1.1~192.168.1.2( 192.168.1.0为该网段网络号,192.168.1.3为该网段广播号)

设置关于默认静态路由的静态路由设置：0.0.0.0/0 via 192.xxxxxxx(下一跳地址)



路由器dns出现问题，不会进行响应。同时不能访问192.168.1.1。不管是访问什么网页都会出现502 Server dropped connection。网上搜索到502 Bad Gateway

1.开着路由器但是拔掉所有的连接线，并重置

2.将电脑连接到上面，先不连接网口，应该可以解决问题了。

我真的很无语，直接访问IP都能502 GateWay，我是路由器自己好了，估计和寝室每天断电重启有关系。

电脑连接着Internet，手机连接到另一个无线网卡的wifi，使用wireshark抓包我遇到的都只能抓有线网卡的包，不过理论上可以直接对无线网卡进行过滤性抓包。

360杀毒会不断向一个服务器发送POST /msvquery。内容还是二进制流，疑似存在后台交流。

用360测速可以测出是哪里的网络，什么营业厅的。应该是借助IP。360测速原理是无视DNS的多线程测试，有次测速正常但是打开网页不行。无视DNS应该是使用了临时的DNS

HTTP persistence connection有两种：

Without pipelining：同步发送数据包

With pipelining: 异步发送数据包（确实这个叫做流水线形式，原因未知）

HTTP 1.0 只有与GET、POST、HEAD这三种方法。

Interior Gateway Protocols (IGP)：AS系统之间的路由器选择协议，常见的有RIP、OSRF、IGRP

RIP（基于UDP）: Routing Information Protocol:RIP 主要设计来利用同类技术与大小适度的网络一起工作，因此通过速度变化不大的接线连接。RIP 比较适用于简单的校园网和区域网，不适于复杂网络的情况。 RIP有两个版本：RIPv1和RIPv2，它们均基于经典的距离向量路由算法，最大跳数为15跳。使用UDP数据包更新路由信息。路由器每隔30s更新一次路由信息，如果在180s内没有收到相邻路由器的回应，则认为去往该路由器的路由不可用，该路由器不可到达。如果在240s后仍未收到该路由器的应答，则把有关该路由器的路由信息从路由表中删除。不会存在cost一样的多条路径。使用算法名字叫做距离向量，根据跳数最少来进行选择。

OSPF（基于IP）：Open Shortest Path First：OSPF协议由三个子协议组成：Hello协议（需要经过认证更加具有安全性）、交换协议和扩散协议。其中Hello协议负责检查链路是否可用，并完成指定路由器及备份指定路由器；交换协议完成“主”、“从”路由器的指定并交换各自的路由数据库信息；扩散协议完成各路由器中路由数据库的同步维护 。传送方位是整一个AS系统，并没有限制最大跳数，所以在大型网络系统里面是分层的。可以存在多条cost一样的路径。支持组播，这时候名字叫做MOSPF。使用算法名字叫做 链路状态，根据代价最低进行选择。属于一种IGP（内部网关协议）。

BGP：Border Gateway Protocol（基于TCP，边界网关协议）：在BGP网络中，可以将一个网络分成多个自治系统。自治系统间使用eBGP广播路由，自治系统内使用iBGP在自己的网络内广播路由。 使用算法名字叫做路径-向量，根据较好的路径来选择。存在四种消息机制：OPEN（建立TCP连接）、UPDATE（更新信息）、KEEPALIVE、NOTIFICATION（关闭连接、报告错误的消息）

EGP（外部网关协议）：是一个在自治系统网络中两个邻近的网关主机（每个都有它们自己的路由）间交换路由信息的协议，EGP常常被用来在英特网的两个主机间交换路由表信息。由于EGP存在很多的局限性，IETF边界网关协议工作组制定了标准的边界网关协议（BGP），当前被广泛使用。

区分距离向量路由协议与链路状态路由协议的区别。前者计算简单，后者收敛到最佳更快（因为不会产生环）。距离矢量最大的缺点是会产生路由环路。可以用跳数，水平分割等避免。信息传输量应该是DV协议更多。链路状态算法（也称最短路径算法）发送路由信息到互联网上所有的结点，然而对于每个路由器，仅发送它的路由表中描述了其自身链路状态的那一部分。距离向量算法（也称为Bellman-Ford算法）则要求每个路由器发送其路由表全部或部分信息，但仅发送到邻近结点上。从本质上来说，链路状态算法将少量更新信息发送至网络各处，而距离向量算法发送大量更新信息至邻接路由器。 ——由于链路状态算法收敛更快，因此它在一定程度上比距离向量算法更不易产生路由循环。但另一方面，链路状态算法要求比距离向量算法有更强的CPU能力和更多的内存空间，因此链路状态算法将会在实现时显得更昂贵一些。除了这些区别，两种算法在大多数环境下都能很好地运行。

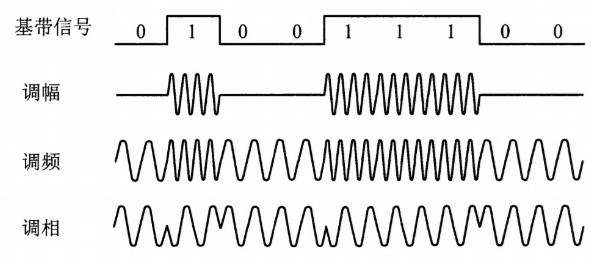
模二运算即将运算结果最后还要在模二。

WFQ：加权公平队列 http://blog.sina.com.cn/s/blog\_5de56eee0102v30o.html

基带传输：在数据通信中，表示计算机二进制的比特序列的[数字数据](https://www.baidu.com/s?wd=%E6%95%B0%E5%AD%97%E6%95%B0%E6%8D%AE&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)信号是典型的矩形脉冲信号; 矩形脉冲信号的固有频带称做基本频带,简称为基带,矩形脉冲信号就叫做[基带信号](https://www.baidu.com/s?wd=%E5%9F%BA%E5%B8%A6%E4%BF%A1%E5%8F%B7&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)。

宽带传输中的所有信道都可以同时发送信号。传输的是模拟信号。

模拟信号与数字信号之间的转换叫“调制/解调”——调幅 调频  调相



直通线、交叉线、全反线：https://jingyan.baidu.com/article/b7001fe172c59b0e7382dd6e.html 一般来说，如果没有特殊指明，平常所做的普通网线都是指两端水晶头都采用568A线序或者都采用568B线序的网线（现在全是568B），即两端都是一样的线序的网线，叫做直通线。http://tieba.baidu.com/p/4019283372 说明了使用这些标准的原因，同时有一张RJ-45 中连线颜色的图片。

网线水晶头水晶头一头采用568a另一头采用568b。这种方法就叫做交叉线，同种设备使用的。两台电脑都是用这种线连接。不同设备用直通线，两头都是a或者b。

udp/tcp checksum头部：假如checksum的运算结果为0，那么传输的checksum将全为1（即用0xFFFF代替0x0000），因为这在补码运算中是等效的。而如果传输的checksum为全0(0x0000)，就说明发送方并没有计算checksum

1.把伪首部添加到UDP上；

2.计算初始时是需要将检验和字段添零的；

3.把所有位划分为16位（2字节）的字

4.把所有16位的字相加（如果总长度为奇数个字节，则在最后增添一个位都为0的字节），如果遇到进位，则将高于16字节的进位部分的值加到最低位上，举例，0xBB5E+0xFCED=0x1 B84B，则将1放到最低位，得到结果是0xB84C

5.将所有字相加得到的结果应该为一个16位的数，将该数取反则可以得到检验和checksum。

tcp四次挥手的时候，两个方向的断开是独立的，每个方向发送一个FIN，对方回复一个ACK，但同时，TCP规定ACK可以捎带在其他数据包当中，所以你看到的主动断开连接一方本应收到的ACK，是被对方的FIN包捎带过来的，就变成了三个包。并不是所有的情况下都是这样，典型的一种情况是，主动断开的一方发送FIN之后，被动一方仍然有数据要继续发送，就会先ACK这个FIN，然后继续发送数据（在此过程中主动断开一方仍然会继续ACK这些数据），直到数据发送完毕之后再发送FIN并接收对方的ACK

校园中通常采用锐捷的RG SAM+portal(如果是二代协议叫做eportal)作为无线WiFi mac认证、计费系统。这套系统由Supplicant（客户端）、NAS设备（支持802.1x端口认证功能的交换机，Authenticator）、Radius Server（Authentication Server，SAM服务器）。客户端要么安装了客户端，要么需要通过IE访问被重定向的网址。Windows服务器系统会采用MSQL，linux系统一般是redhat+Oracle。EAP在802.1x中改为EAPOL（EAP over LAN）。

公用端口：0-1023

注册端口：1024-49151

动态端口/私有端口：49152-65535

镜像端口：把端口的流量完全复制一份到另外一个端口。好一点的交换机有此功能。公司内部可以用于流量的完全统计。而这需要一块专门的网卡。如果流量已经超过桐庐，那么就多个网卡和交换机上的多个端口双方捆绑成一个网卡和一个虚拟端口。公司可以达到一天50G

FTP 数据传输 tcp 20

FTP 控制 tcp 21

ssh tcp22 sftp、scp

Telnet tcp 23

Smtp tcp 25

DNS udp 53

Tftp udp 69

finger tcp 79

http tcp 80

POP3 tcp 110端口

msrpc tcp 135（还有1025端口）

NetBios tcp udp137-138 tcp139

HTTPS tcp 443

windows文件共享端口SMB（microsoft-ds） 445

rlogin远程连接 监听512,513,514

tcpwrapped 514（可以进行爆破，msfconsole的rsh\_login支持）

RMI registry 1099/tcp

M$ SQL 1433 TCP and UDP

SQL server浏览器 udp 1434

ingreslock 1524 使用telnet连接就获取root

radius 默认使用1812tcp作为认证端口、1813tcp用于记账

nfs tcp/udp 2049

Mysql 3306

远程桌面RDP 3389

distccd tcp 3632

SIP tcp/udp 5060 一般推荐使用tcp

postgresql 5432

VNC over HTTPP 5800

VNC 5900

UnreaIRCD IRC 6667

Nvidia Mental Ray Satellite Service 7414

http-proxy 8080（不过Tomacat的服务端口就是8080）

tor 9150

tor-orport 9001

serv-u 43958

win7 5357 （Function Discovery Resource Publication服务 发布该计算机和连接到该计算机的资源）（默认处于手动开启状态）

海康网络摄像机IPC的端口为:HTTP（80），服务端口1（8000），服务端口2（8200），RTSP端口（默认554）

H.232 是端口群

瞬杀漏洞：

Linux：

NIS（Network Infomation System）网络信息系统，一旦被得到服务其的NIS域名，使用RPC查询就可通过NIS映射（主机关键信息）返回信息。

如果有resersd服务开着那么就修使用rusers 获取用户信息

如果有rwhod，可以使用rwho查出当期登录的用户

如果有nfs开着，就使用showmount -e 获取挂载信息

如果有xhost +在X服务器上打开了，这会允许任何本地或远程用户来访问自己的服务器。rhost -表示禁止将来的一切连接。xscan就是扫描整个子网并寻找一台开放的X服务器，并记录它的键击。被推荐应该启动X服务器的安全键盘选项。

Windows：

如果有finger开着，就是用finger 获取系统信息

n-PDU 第n层layer的协议数据单元

protocol stack 这么多层layer的总和

point-to-point 点到点 这是形容对等实体之家你的连接，比end-to-end更加精确