架设一个网站的基础流程：了解基础网络——>了解架站的目的——>Linux 安装硬盘规划——>了解欲架设网站的服务原理——>服务的套件安装、漏洞修补、套件升级——>主机设定、启动、观察与除错——>客户端设定、观察和除错——>安全性设定——>服务日志、登录文件和备份管理

网络:通过通信线路和通信设备将地理位置不同且具有独立功能的多个计算机系统相互连接在一起，由网络操作系统和协议软件进行管理，能实现信息交换、资源共享的系统称为计算机网络

以太网络：常见的网络硬件包括有最常见的以太网络(IEEE 制定的 802.3u 标准)，还有速度最快的光纤网络，以及蓝牙无线技术等

0、数据编码技术

(1)不归零编码 用两个不同的电位分别表示二进制数字代码 0 和 1，例如用高电位表示 1，低电位表示 0。

(2)曼彻斯特编码 将每比特信号周期 T 分为前 T/2 和后 T/2，用前 T/2 传送该比特的反（原）码，用后 T/2 传送该比特的原（反）码。

(3)差分曼彻斯特编码 对曼彻斯特编码的一种改进。每比特的取值则根据其开始处是否出现电平的跳变来决定。通常规定有跳变者代表二进制“0”，无跳变者代表二进制“1”

1、网络间数据传输方式

数据的传输是指数据在传输信道上的传输数据的传递方式，可分为串行传输和并行传输：

(1)并行传输原理：将 8 位、16 位或者 32 位的数据按数位宽度同时进行传输，每一个数位都有自己的数据传输线和发送、接受设备，并行传输一般只用于 1 米以内的极短距离的传输

(2)串行传输原理：在一根数据传输线上，每次传送一位二进制数据，即数据一位接一位地传送，在传输距离远和传输数字数据时，都采用串行通信方式。

计算机网络普遍采用串行通信方式传输数据，根据数据在传输线上的传输方向可分为：

(1)**单工通信**：在通信线路上，数据只可按一个固定的方向传送而不能进行相反方向传送的通信方式称为单工通信，如无线电广播或有线电广播、电视广播就属于这种类型。

(2)**半双工通信**：通信的双方都可以发送信息，但双方不能同时发送（当然也不能同时接收）。称为半双工通信。对讲机就属于这种类型。

(3)全双工通信：通信的双方可以同时发送和接收信息称为全双工通信。日常生活中使用的电话就属于这种类型。单项通信只需要一条信道，而双向交替通信或双向同时通信则都需要两条信道（每个方向各一条）。显然，双向同时通信的传输效率最高。

2、网络适配器(NIC，Network Interface Card，网卡)：

完成物理层和数据链路层的功能。用于实现计算机和传输介质之间的物理连接，为计算机之间相互通信提供一条物理通道，并通过这条通道进行高速数据传输。

MAC地址，又称网卡地址，物理地址，硬件地址。48位二进制表示，通常12位十六进制数表示。

3、中继器（Repeater）：

工作原理：由于数字信号在传输过程中，其高次谐波最易衰减而使信号变形，电缆上的阻抗、容抗也会使信号幅值和形状变小或失真，对弱信号进行放大或再生，以便延长传输距离，但不对信号作校验等其他处理，故即使是一个错误的信息帧或信号中含有噪声，均整形放大。

4、集线器（Hub）：

又称集中器，可以看成是一种多端口的中继器，集线器的主要功能是对接收到的信号进行再生整形放大，以扩大网络的传输距离，同时把所有节点集中在以它为中心的节点上。所有传到集线器的数据均被广播到与之相连的各个端口，容易形成网络风暴，造成网络堵塞。它工作于OSI参考模型的物理层和数据链路层的MAC子层。

其优点是：当网络系统中某条线路或某节点出现故障时，不会影响网上其他节点的正常工作，缺点是：用户带宽共享，带宽受限。其带宽由它的端口平均分配，集线器的同一时刻每一个端口只能进行一个方向的数据通信，而不能像交换机那样进行双向双工传输，网络执行效率低。

5、网桥(Bridge)：

又称桥接器，具有单个输入输出端口，是工作在数据链路层的一种网络互联设备，是在互联的局域网之间实现数据帧的存储和转发。

6、交换机（Switch）

又称交换式集线器，工作在数据链路层上的、基于MAC识别、能完成封装转发数据包功能的

网络设备。交换机是集线器的升级换代产品，局域网交换机拥有许多端口，可以连接不同网段。

交换机的特性：

（1）通过支持并行通信，提高了交换机的信息吞吐量。

（2）交换机采用全双工技术进一步提高端口的带宽。

（3）将传统的一个大局域网上的用户分成若干工作组，每个端口连接一台设备或连接一个工作组，有效地解决拥挤现像。

根据交换机完成的协议层功能可分为：

（1）二层交换机：二层交换机工作于 OSI 模型的第 2 层（数据链路层），故称为二层交换机。二层交换技术的发展已经比较成熟，二层交换机属数据链路层设备，可以识别数据包中的 MAC 地址信息，根据 MAC 地址进行转发，并将这些 MAC 地址与对应的端口记录在自己内部的一个地址表中。

（2）三层交换机：三层交换机就是具有部分路由器功能的交换机，三层交换机的最重要目的是加快大型局域网内部的数据交换，所具有的路由功能也是为这目的服务的，能够做到一次路由，多次转发。对于数据包转发等规律性的过程由硬件高速实现，而像路由信息更新、路由表维护、路由计算、路由确定等功能，由软件实现。三层交换技术就是二层交换技术+三层转发技术。传统交换技术是在 OSI 网络标准模型第二层——数据链路层进行操作的，而三层交换技术是在网络模型中的第三层实现了数据包的高速转发，既可实现网络路由功能，又可根据不同网络状况做到最优网络性能。

集线器和交换机的区别	集线器	交换机
工作层次不同	物理层和数据链路层的MAC子层	数据链路层
数据传输方式不同	广播方式	数据只对目的节点发送
带宽占用方式不同	各端口均分总带宽	每个端口具有自己的带宽
传输模式不同	共享传输介质，一次只能传输一个任务	全双工模式，可同时接受发送数据

7、路由器：

路由器（Router），是连接因特网中各局域网(LAN)、广域网(WAN)的网络层设备，它会根据信道的情况自动选择和设定路由，以最佳路径，按前后顺序发送信号。它依据网络层信息将数据包从一个网络向另一个网络转发。它能将不同网络或网段之间的数据信息进行“翻译”，以使它们能够相互“读”懂对方的数据，从而构成一个更大的网络。

路由器的功能：（1）网络互连。路由器支持各种局域网和广域网接口，主要用于互连局域网

和广域网，实现不同网络相互通信；（2）数据处理。提供包括分组过滤、分组转发、优先级、复用、加密、压缩和防火墙等功能；（3）网络管理。路由器提供配置管理、性能管理和流量控制等功能。

路由器的工作原理：

（1）路由器接收来自它连接的某个网站的数据；

（2）路由器将数据向上传递，并且（必要时）重新组合 IP 数据报；

（3）路由器检查 IP 头部中的目的地址，如果目的地址位于发出数据的那个网络，那么路由器就放下被认为已经达到目的地的数据，因为数据是在目的计算机所在网络上传输；

（4）如果数据要送往另一个网络，那么路由器就查询路由表，以确定数据要转发到的目的地；

（5）路由器确定哪个适配器负责接收数据后，通过软件传递数据，以便通过网络来传送数据。

路由器和交换机的区别	路由器	交换机
工作的层次不同	网络层	数据链路层
数据转发依据的对象不同	利用地址IP确定数据转发的目的地址，IP地址一般由网络管理员或系统自动分配(DNS)	利用MAC地址确定转发数据的目的地址,MAC地址硬件自带，一般不可更改
分割网域不同	连接到路由器上的网段分配成不同的广播域，广播数据不会穿过路由器，提供了防火墙功能	交换机只能分割冲突域，不能分割广播域。

8、网关（Gateway）

网关，又叫网间连接器、协议转换器，是一个网络连接到另一个网络的“关口”，它可以支持不同协议之间的转换，实现不同协议网络之间的互连。网关的功能体现在 OSI 模型的高层，它将协议进行转换，将数据重新分组，以便在两个不同类型的网络系统之间通信，网关一般是软件产品，网关的功能就是在负责不同网域之间的封包转递(IP Forwarder)，由于路由器具有 IP Forwarder 的功能，并且具有管理路由的能力，所以可以将来自不同网域之间的封包进行转递的功能。此外，主机 IP 与设定的 Gateway 必定是在同一个网段内！

网关和多协议路由器组合在一起，可以连接多种不同的系统。和网桥一样，网关可以是本地的，也可以是远程的。主要有三类网关：协议网关、应用网关、安全网关

连接端口	服 务 名 称 与 内 容
20	FTP-data，档案传输协议所使用的主动数据传输端口
21	FTP，档案传输协议的命令通道
22	SSH，较为安全的远程联机服务器
23	Telnet，早期的远程联机服务器软件

25	SMTP，简单邮件传递协议，用在作为 mail server 的端口
53	DNS，用在作为名称解析的领域名称服务器
80	http，就是全球信息网服务器
110	POP3，邮件收信协议，办公室用的收信软件都是透过他
443	https，有安全加密机制的http+ssl服务器

10、OSI 七层协议模型：

(1)采用结构化思想，分为若干层，各层之间是相互独立的;

(2)层间的关系是服务与被服务的关系;

(3)网络上的节点间对应层遵守一致的规约。

OSI七层模型	
Layer 7 应用层 Application Layer	**完全与程序有关，包括定义出档案的读取、复制、开启、关闭等等， 常见的程序包括有浏览器、数据库处理系统与电子邮件系统等**
Layer 6 表示层 Presentation Layer	**在应用程序上面所制作出来的数据格式不一定符合网络传输的标准编码格式的！在这个层级当中，将来自本地端应用程序的数据格式转换(或者是重新编码)成为网络的标准格式， 然后再交给底下传送层等的协议来进行处理。所以，在这个层级上面主要定义的是网络服务(或程序)之间的数据格式的转换， 包括数据的加解密也是在这个分层上面处理。**
Layer 5 会话层 Session Layer	**在这个层级当中定义了两个地址之间的联机信道的连接与挂断，此外，亦可建立应用程序的会话、 提供其它加强型服务如网络管理、签到签退、会话的控制等。如果说传送层是在判断资料封包是否可以正确的到达目标，那么会话层则是在确定网络服务建立联机的确认，例如TCP的三次握手**
Layer 4 传输层 Transport Layer	**这一个分层定义了发送端与接收端的联机技术(如TCP/UDP技术)， 同时包括该技术的封包格式，数据封包的传送、流程的控制、传输过程的侦测检查与复原重新传送等，以确保各个资料封包可以正确无误的到达目的端**
Layer 3 网络层 Network Layer	**IP (Internet Protocol) 就是在这一层定义的，同时也定义出计算机之间的联机建立、终止与维持等，数据封包(packet)的传输路径选择等，因此这个层级当中最重要的除了IP之外，就是封包能否到达目的地的路由(route)概念了！此外，这一个网络层可以涵盖实体层与数据链路层，通常不需要设定硬件与相关MAC的数据，就是因为网络层已经隐藏了底下两层，只要设定好 IP 就能够上网**
Layer 2 数据链路层 Data-Link Layer	**数据要使用电子讯号传送，就需要制订各种网络型态的讯框 (frame) 才能确保数据可以在不同的网络媒体进行传送。在这一层当中就制订了 frame 的格式以及通过网络的方式。 包括讯框的数据格式、错误控制、流量控制、检查数据传输错误的方法等等。 既然与讯框有关，当然这个层级就与前面提到的 MAC 有很强烈的相关性，目前的网络使用的是 IP 来进行联机的！但硬件数据却是由讯框所传送的。为了要将两者对应 (MAC 与 IP 的对应)，就必须要经由 Address Resolution Protocol (ARP) 这个协定来解析！**
Layer 1 物理层 Physical Layer，PHY	**在这个层级当中主要定义了最基础的网络硬件标准，包括各种网络线、各种无线联机方式，各种设备规范、以及各种接头的规则，还有传输讯号的电压等等，与硬件有关的标准大多都在这个层级当中定义**

OSI，即开放系统互联参考模型(Open System Interconnection Reference Model,OSI/RM)七层协议当中，前两层(物理层与数据连接层)主要就是由一些硬件标准所规范出来的，网络层

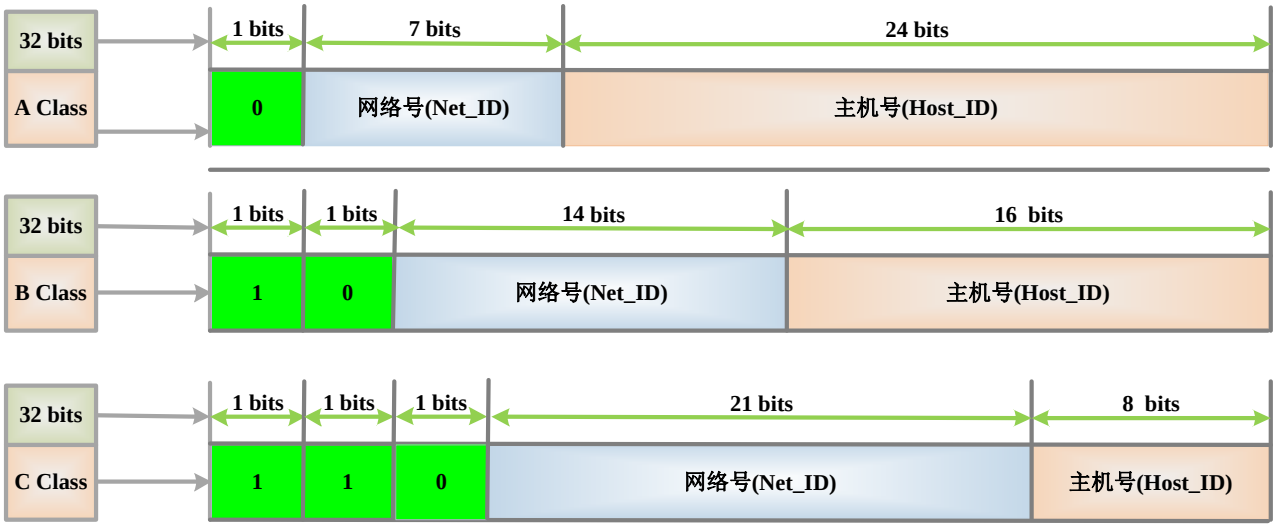
和传输层则与 TCP/IP 有关，会话、表示和应用层主要与操作系统及应用程序相关。每个分层都接受由它下一层所提供的特定服务，并且负责为自己的上一层提供特定的服务，上下层之间进行交互时所遵循的约定叫做“接口”。同一层之间的交互所遵循的约定叫做“协议”。

IP 和 MAC：网络计算中以二进制的 bit 为单位，1bit=8bits，数据传输速率是指每秒线路上传输的二进制数据位 bit 数，单位 bps(bit per second)，10/100Mbps=1.25/12.5Mbytes。MAC(Media Access Control 媒体存取控制)，MAC 常用来作为硬件地址(Hardware address)的代称。当主机想要找出目标 IP 时，就会对整个局域网络进行广播封包(broadcast)的传送， 这个广播封包可以对所有局域网络内的计算机要求回报 IP 与 MAC ， 当目标 IP 看到这个广播封包时，就会响应主机相关的 MAC 信息，如果非目标主机接到这个封包，就会主动的忽略！如此一来，就可以取得目标主机的 MAC！而这个目标主机的 MAC 就会被记录到主机内的 ARP table (ARP table 在内存中)，MAC 是不能跨路由的！如下是 MAC 信息的头部内容：

查阅 ARP 记录，可以使用 arp 命令：

```
[root@linux ~]# arp -s <hostname[IP]> [Hardware_address]
参数:
-n : 将主机名称以 IP 的型态显示
-d : 将 hostname 的 hardware_address 由 ARP table 当中删除掉
-s : 设定某个 IP 或 hostname 的 MAC 到 ARP table 当中
```

11、IP 地址分类：



A 类公网 IP 地址：地址范围 0.0.0.0 -- 127.0.0.0 默认子网掩码 255.0.0.0

B 类公网 IP 地址：地址范围 128.0.0.0 -- 191.255.0.0 默认子网掩码 255.255.0.0

C 类公网 IP 地址：地址范围 192.0.0.0 -- 223.255.255.0 默认子网掩码 255.255.255.0

D 类是组播地址：地址范围 224.0.0.0 -- 239.255.255.255(第一个字节的前四位固定为 110)

E 类作为保留研究：地址范围 240.0.0.0--255.255.255.255(第一个字节的前五位固定 11110)

保留私网地址：

A 类私网：10.0.0.0 -- 10.255.255.255 默认子网掩码 255.0.0.0

B 类私网：172.16.0.0 -- 172.31.255.255 默认子网掩码 255.255.0.0

C 类私网：192.168.0.0 -- 192.168.255.255 默认子网掩码 255.255.255.0

特殊地址：

本地还回地址：127.0.0.1 -- 127.0.0.255

主机请求无法被分配时候，自己生成的地址：169.254.0.0

子网掩码：用二进制方式表示则是一个 32 位的数字。它对应 IP 地址网络标识部分的位全部为“1”，对应 IP 地址主机标识的部分则全部为“0”。由此，一个 IP 地址可以不再受限于自己的类别，而是可以用这样的子网掩码自由地定位自己的网络标识长度。当然，子网掩码必须是 IP 地址的首位连续的“1”。对于子网掩码，目前有两种表示方式：一种是用 32 位数字表示；另一种，则是在每个 IP 地址后面追加网络地址的位数，用“/”隔开。

子网掩码作用：用来区分 IP 地址的网络号与主机号；当 TCP/IP 网络上的主机相互通信时，就可以利用子网掩码得知这些主机是否在相同的网络区段内；用来将网络分割为多个子网。

在 IP 的 32bits 中，分为 HOST_ID 和 Net_ID，以 C 类网域 192.168.0.0~192.168.0.255 为例：

```
192.168.0.0~192.168.0.255 这个 C Class 的说明:
11000000.10101000.00000000.00000000
11000000.10101000.00000000.11111111
|----------Net_ID---------|-host--|
以二进制说明 Network 第一个数字的定义:
A Class : 0xxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx ==> NetI_D 的开头是 0
          |--net--|---------host------------|
B Class : 10xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx ==> NetI_D 的开头是 10
          |------net-------|------host------|
C Class : 110xxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx ==> NetI_D 的开头是 110
          |-----------net-----------|-host--|
三种分级在十进制的表示:
A Class : 0.xx.xx.xx ~ 126.xx.xx.xx
B Class : 128.xx.xx.xx ~ 191.xx.xx.xx
C Class : 192.xx.xx.xx ~ 223.xx.xx.xx
```

将 127.xx.xx.xx 这个 A Class 的网段拿到操作系统当中，来做为内部循环网络(loopback)的

回路测试。

Netmask(子网掩码)可实现子网络的切分，既然 Net_ID 是不可变的，那就假设他所占据的 bits 已经被用光了（全部为 1），而 Host_ID 是可变的，就将他想成是保留着（全部为 0），所以， Netmask 的表示就成为：如下以 C 类网段为例：

```
192.168.0.0~192.168.0.255 这个 C Class 的 Netmask 说明
11000000.10101000.00000000.00000000
11000000.10101000.00000000.11111111
|----------Net_ID---------|-host--|
11111111.11111111.11111111.00000000 <== Netmask 二进制
255 . 255 . 255 . 0 <== Netmask 十进制
Class A, B, C 三个等级的 Netmask 表示方式：
A Class : 11111111.00000000.00000000.00000000 ==> 255. 0. 0. 0
B Class : 11111111.11111111.00000000.00000000 ==> 255.255. 0. 0
C Class : 11111111.11111111.11111111.00000000 ==> 255.255.255. 0
Network/Netmask
192.168.0.0/255.255.255.0
192.168.0.0/24 <==因为 Net_ID 共有 24 个 bits
Netmask: 255.255.255.0 <==网域定义中，最重要的参数
Network: 192.168.0.0 <==第一个 IP
Broadcast: 192.168.0.255 <==最后一个 IP
可用以设定成为主机的 IP 数：
192.168.0.1 ~ 192.168.0.254
```

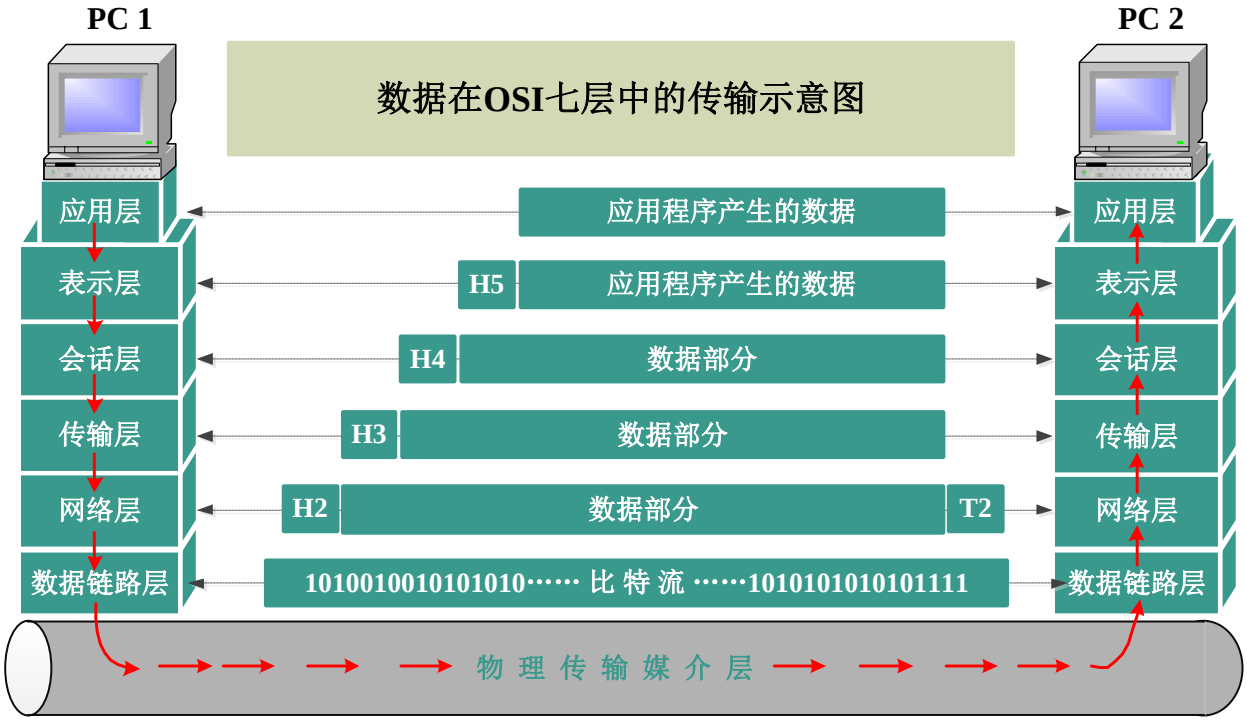
12、数据在各主机间的传送介绍：

<table>
<tr><th colspan="3">OSI七层模型</th><th colspan="2">TCP/IP模型</th></tr>
<tr><td rowspan="3">第5~7层完成信息处理服务的功能，称之为网络高层</td><td>为应用程序提供子网络服务</td><td>应用层</td><td rowspan="3">应用层</td><td rowspan="3">应用层是所有用户所面向的应用程序的统称。TCP/IP协议族在这一层有着很多协议支持不同的应用，如httpd服务、ftp服务等</td></tr>
<tr><td>设定固有数据格式和网络标准数据格式转换,加密等</td><td>表示层</td></tr>
<tr><td>通信管理，确保建立和断开通信连接，管理传输层以下的分层</td><td>会话层</td></tr>
<tr><td>第4层完成高层和底层的衔接</td><td>管理两个节点之间的数据传输。确保数据被可靠的传送到目标地址</td><td>传输层</td><td>传输层</td><td>提供应用程序间的通信，TCP/IP协议族在这一层有TCP和UDP等</td></tr>
<tr><td rowspan="3">第1~3层完成数据的交换以及传输，称之为网络底层，即通信子网</td><td>地址管理和路由选择，实现拥塞控制、网络互联</td><td>网络层</td><td>网络层</td><td>定义IP地址格式，保证数据在网上传播</td></tr>
<tr><td>互连设备之间传送和识别数据帧</td><td>数据链路层(MAC层)</td><td rowspan="2">网络接口层</td><td rowspan="2">负责接收IP数据包并通过网络发送，或者从网络上接受物理帧，抽出IP数据报，交给IP层</td></tr>
<tr><td>利用传输介质建立物理连接，为数据链路层提供数据传输服务，数据传输单元是比特</td><td>物理层
Physical Layer</td></tr>
</table>

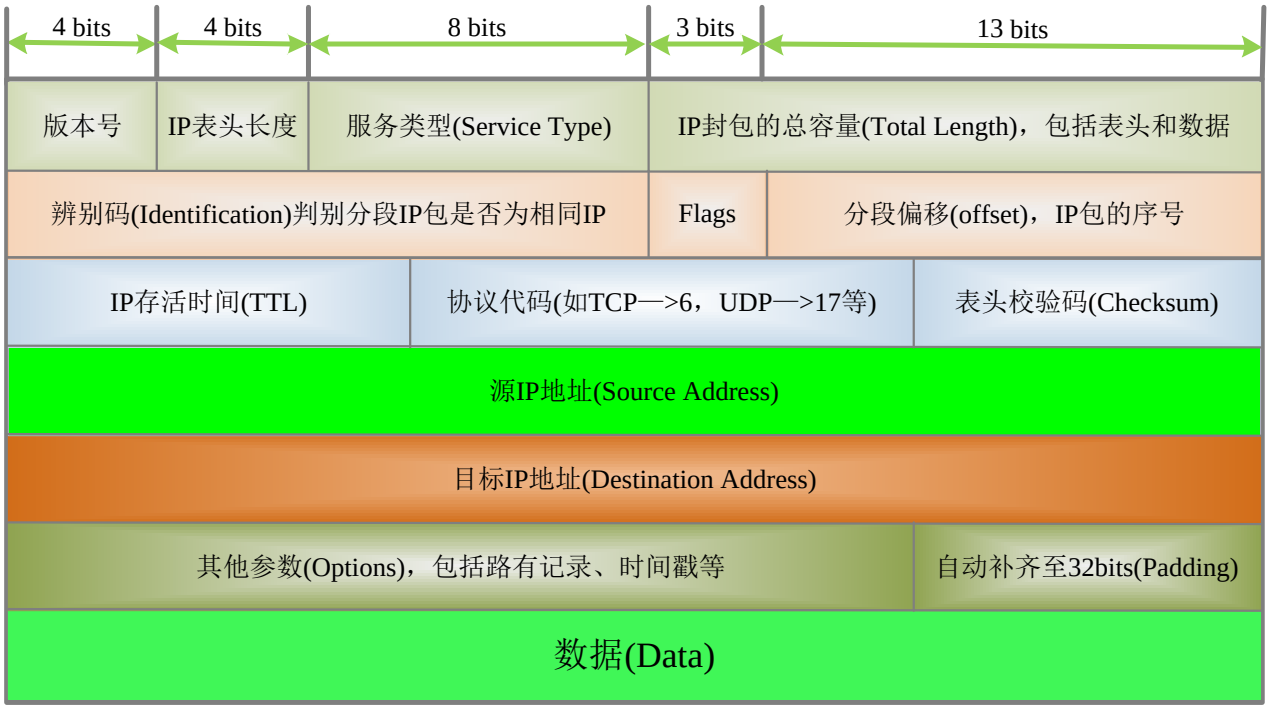
数据的封装(发送方)： AP1 先将其数据交给第 7 层（应用层）。第 7 层加上必要的控制信息 H5 就变成了下一层的数据单元。第 4 层收到这个数据单元后，加上本层的控制信息 H4，再交

到第 3 层，称为第 3 层的数据单元。依此类推。到了第 2 层后，控制信息分成两部分，分别加到本层数据单元的首部（H2）和尾部（T2），而第 1 层由于是比特流的传送，不加控制信息。

数据的拆封（接收方）：当这一串的比特流经网络的物理媒体传送到目的站时，就从第 1 层依次上升到第 5 层。每一层根据控制信息进行必要的操作，然后将控制信息剥去，将该层剩下的数据单元上交给更高的一层。最后，把应用进程 AP1 发送的数据交给目的站的应用进程 AP2。



13、IP 报文的首部



(1)、Version(版本) 这个 IP 封包的版本，目前的还是 IPv4 这个版本；

(2)、IHL(Internet Header Length, IP 表头的长度)，告知这个 IP 封包的表头长度，单位为字节(bytes)，此 IHL 长度的范围为 5~15；

(3)、Type of Service(服务类型)可选内容为『PPPDTRUU』，表示这个 IP 封包的服务类型，主要分为：PPP：表示此 IP 封包的优先度；

D：若为 0 表示一般延迟(delay)，若为 1 表示为低延迟；

T：若为 0 表示为一般传输量 (throughput)，若为 1 表示为高传输量；

R：若为 0 表示为一般可靠度(reliability)，若为 1 表示高可靠度；UU：保留尚未被使用；

(4)、Total Length(总长度)IP 封包的总容量，包括表头与内容 (Data) 部分。最大可达 65535

(5)、Identification(辨别码)判别拆分后的小 IP 包是否来自与同一个数据报文

(6)、Flags(特殊标记)内容为『0DM』，其意义为：

D：若为 0 表示可以分段，若为 1 表示不可分段；

M：若为 0 表示此 IP 为最后分段，若为 1 表示非最后分段；

(7)、Fragment Offset(分段偏移) 表示目前这个 IP 分段在原始的 IP 封包中所占的位置 IP 报文的序号，有这个序号才能将所有的小 IP 分段组合成为原本的 IP 封包大小，透过 Total Length，Identification，Flags 以及这个 Fragment Offset 就能够将小 IP 分段在收受端组合起来；

(8)、Time To Live(TTL, 存活时间)表示这个 IP 封包的存活时间，范围为 0-255。当这个 IP 封包通过一个路由器时， TTL 就会减一，当 TTL 为 0 时，这个封包将会被直接丢弃；

(9)、Protocol Number(协议代码)IP 内含有的数据的协议类型的代号。 常见的网络协议代号：

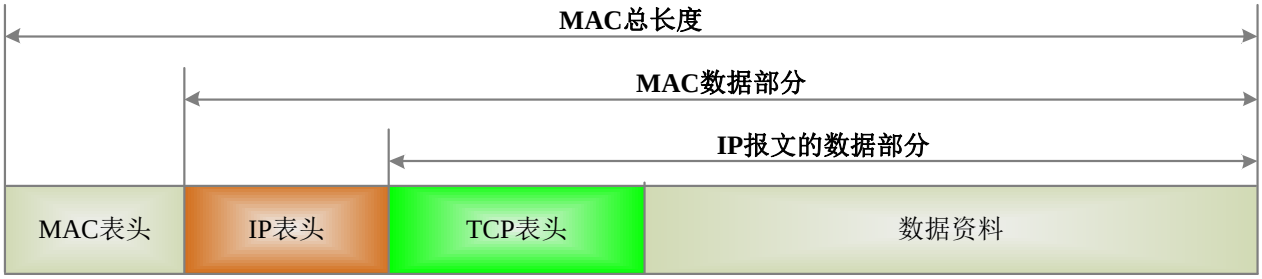
```
1------ICMP    (Internet Control Message Protocol)
4------IP      (IP in IP encapsulation)
6------TCP     (Transmission Control Protocol)
17-----UDP     (User Datagram Protocol)
```

(10)、Header Checksum(表头校验码)用来检查这个 IP 表头的错误检验之用；

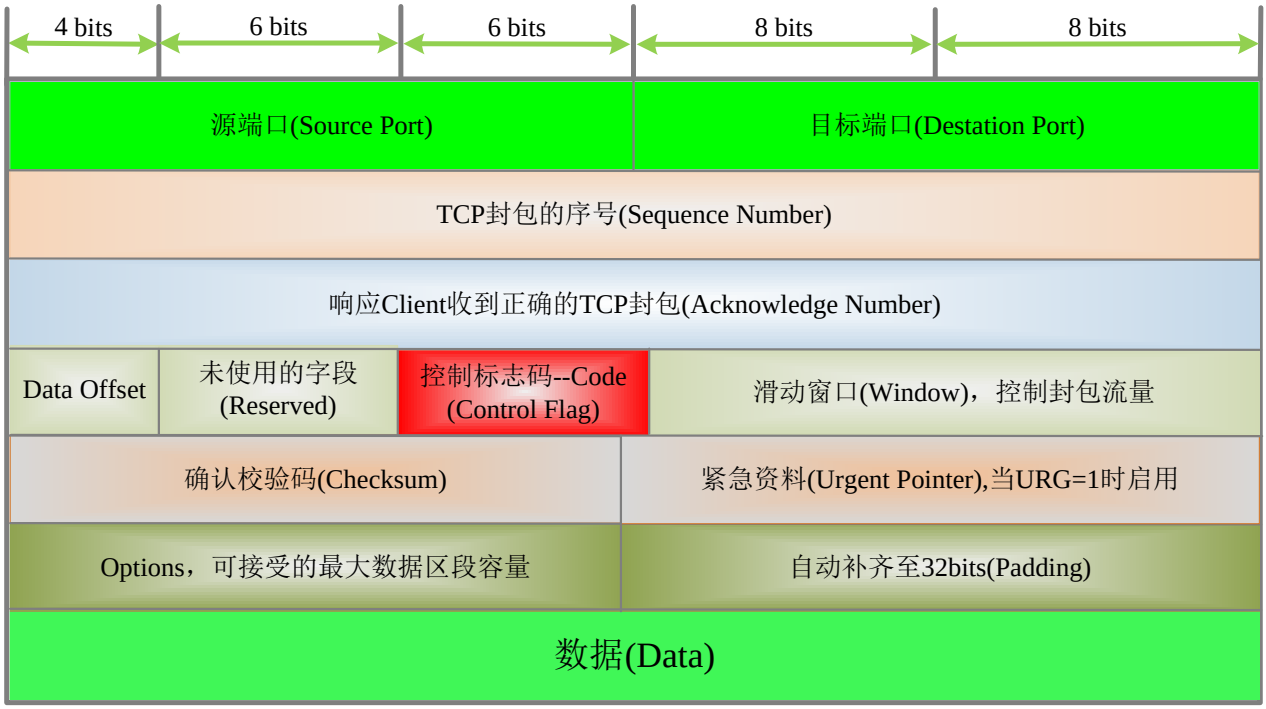
(12)、Padding(补齐项目)IP 每个数据是 32 bits，若 Options 长度不足 32bits 时，则自动补齐；

14、TCP 协议

TCP 协议，传送层的数据打包成常见的 TCP 封包,MAC、IP 和 TCP 的封包数据示意图：



IP 除了表头之外的 Data 内容其实就是 TCP 封包的表头与内容；而 MAC 的 Data 内容，就是一个完整的 IP 封包数据！最终还是得以 MAC 能够支持的最大容许容量， 才能够决定 IP 与 TCP 封包是否需要再进行分段的工作。那么既然 MAC 与 IP 都有表头数据， 想当然尔，TCP 也有表头数据来记录该封包的相关信息, TCP 封包的表头如下：



上图就是一个 TCP 封包的表头数据，各个项目以 Source Port, Destination Port 及 Code 为重点，Code (Control Flag, 控制标志码)，显示两个主机间的联机状态的字段，共有 6 个 bits，分别代表 6 个句柄，若为 1 则为启动。分别说明如下：

(1)、URG(Urgent)：若为 1 则代表该封包为紧急封包， 接收端应该要紧急处理；

(2)、ACK(Acknowledge)：若为 1 代表这个封包为响应封包，则与 Acknowledge Number 有关。

(3)、PSH(Push function)：若为 1 时，要求对方立即传送缓冲区内的其它对应封包，无须等待缓冲区满了才送。

(4)、RST(Reset)：如果 RST 为 1 的时候， 表示联机会被马上结束，而无需等待终止确认手续。这也就是说，这是个强制结束的联机， 且发送端已断线。

(5)、SYN(Synchronous)：若为 1，表示发送端希望双方建立同步处理，也就是要求建立联机。通常带有 SYN 标志的封包表示『主动』要连接到对方的意思。

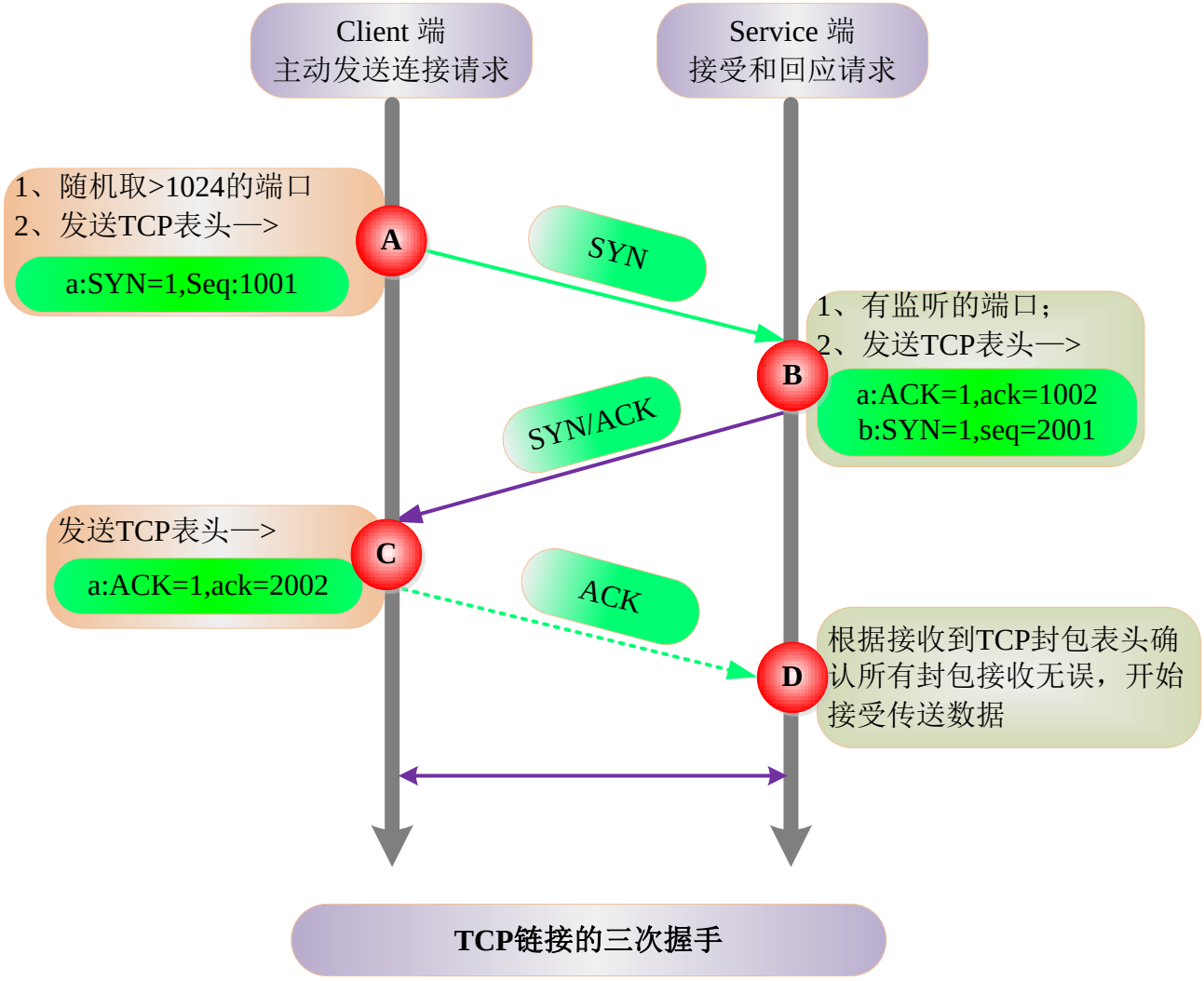
(6)、FIN(Finish)：若为 1 ，表示传送结束， 所以通知对方数据传毕，是否同意断线，只是发送者还在等待对方的响应而已。

其中比较常见到的应该是 ACK/SYN/FIN。通讯端口和 Socket，主机上共有 2^16(65536)个

端口，对于小于 1023 的端口启动需要 root 权限。在下面的封包连接模式当中，在建立联机之前都必须要通过三个确认的动作，所以这种联机方式也就被称为三次握手(Three-way handshake)。分为以下四个阶段介绍如下：

• A:封包发起：当客户端想要对服务器端联机时，就必须要送出一个要求联机的封包， 此时客户端必须随机取用一个大于 1024 以上的端口来做为程序沟通的接口。 然后在 TCP 的表头当中，必须要带有 SYN 的主动联机(SYN=1)，并且记下发送出联机封包给服务器端的序号(Sequence number = 10001) 。

• B:封包接收与确认封包传送：当服务器接到这个封包，并且确定要接收这个封包后，就会开始制作一个同时带有 SYN=1, ACK=1 的封包， 其中那个 acknowledge 的号码是要给 client 端确认用的，所以该数字会比(A 步骤)里面的 Sequence 号码多一号 (ack = 10001+1 = 10002)，服务器也必须要确认客户端确实可以接收服务器的封包才行，所以也会发送出一个 Sequence (seq=20001) 给客户端，并且开始等待客户端给我们服务器端的回应



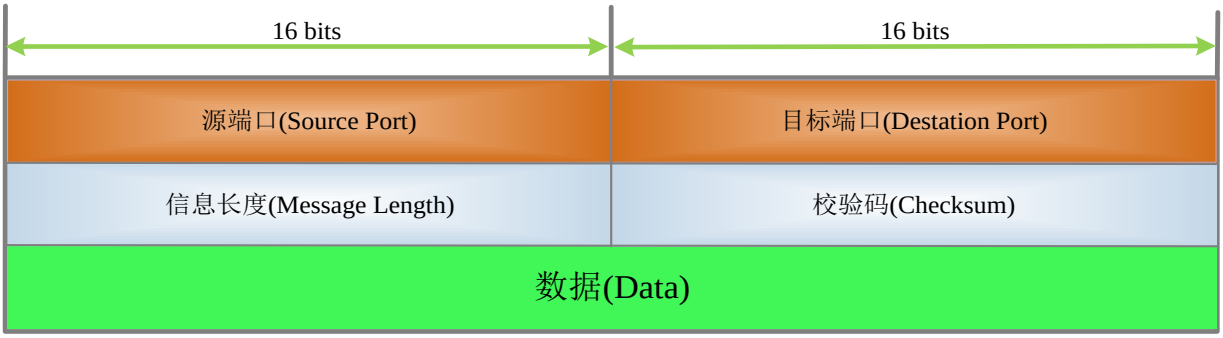
• C:回送确认封包：当客户端收到来自服务器端的 ACK 数字后 (10002) 就能够确认之前那个要求封包被正确的收受了， 接下来如果客户端也同意与服务器端建立联机时，就会再次的发送一个确认封包 (ACK=1) 给服务器，亦即是 acknowledge = 20001+1 = 20002

• D:取得最后确认：若一切都顺利，在服务器端收到带有 ACK=1 且 ack=20002 序号的封包后，就能够建立起这次的联机了。

在建立了联机之后，该次联机通道就可以在客户端与服务器端建立起一对 socket pair，然后通过该 socket pair 进行 TCP 封包的 PSH、FIN 等数据传输与联机中断等动作。Socket pair 由 IP 封包的 IP address 与 TCP 封包的 port number 构成。

15、UDP 协议

UDP 协议，UDP『User Datagram Protocol, 用户数据流协议』，UDP 与 TCP 不一样， UDP 不提供可靠的传输模式，因为不是联机导向的一个机制，在 UDP 的传送过程中，接受端在接受到封包之后，不会回复响应封包（ACK）给发送端，所以封包并没有像 TCP 封包有较为严密的验证机制。UDP 的表头资料如下表所示：



由于 UDP 协议不存在验证和三次握手故而比 TCP 的传输速率快，所以对于数据完整性要求不是太高的情况下，可选择 UDP 协议。许多服务例如 DNS，都开启了 TCP 和 UDP 的端口，优先选择 UDO 协议进行数据的传送。

16、TCP 和 UDP 协议的区别：

(1)、UDP 消耗的计算机资源少，而 TCP 消耗的资源较多。

(2)、应用程序本身如果已经提供数据完整性检查机制，因此不需要再由传输层来执行相同的功能，或者应用程序传输的并非是关键数据。这两种情况下如果使用 TCP 则在不必要的情况下造成延迟较大（TCP 建立、释放连接，发送确认信号灯），故多采用简单、快速的 UDP 协议。

(3)、要使用多点传送或广播发送时等一对多的传送方式时，必须使用 UDP。因为使用面向连接的 TCP 传输时只限于一对一的传送方式。

(4)、TCP 可靠性高，UDP 可靠性低。但是 TCP 也就是为了保证高可靠，使之实现起来比较复

杂；而 UDP 因为不需要考虑可靠性所以实现起来简单、灵活。

(5)、TCP 是面向连接的，而 UDP 是面向非连接的。

传输层协议	应用层协议及端口		描述
TCP	FTP	21	文件传输协议
	Telnet	23	远程登录协议
	SMTP	25	简单邮件传输协议
	HTTP	80	超文本传输协议
	POP3	110	邮件接受协议
	NNTP	119	新闻传输协议
TCP - UDP	DNS	53	域名解析协议
UDP	BOOT	67/68	引导协议
	TFTP	69	简单文件传输协议
	SNMP	161	简单网络管理协议

17、TCMP 协议

ICMP 协议，全称是『 Internet Control Message Protocol, 因特网讯息控制协议 』，基本上，ICMP 是一个错误侦测与回报的机制，最大的功能就是可以确保我们网络的联机状态与联机的正确性！ 同样的，ICMP 封包也是必须要装在 IP 封包的 Data 内才行喔！ 因为在 Internet 上面有传输能力的就是 IP 封包，ICMP 有相当多的类别可以侦测与回报，底下是比较常见的几个 ICMP 的类别 (Type)：

类别代号	类别名称与意义(ping响应的返回码)
0	Echo Reply（代表一个响应信息）
3	Distination Unreachable（表示目的地不可到达）
4	Source Quench（当 router 的负载过高时，此类别码可用来让发送端停止发送讯息）
5	Redirect（用来重新导向路由路径的信息）
8	Echo Request（请求响应讯息）

ping 和 traceroute 命令可以透过 ICMP 封包的辅助来确认与回报网络主机的状态。

MTU，最大传输单元 (Maximum Transmission Unit, MTU)，规范 TCP 以及 IP 数据包封包打包的最大容量。

Internet 使用的是 TCP/IP 通讯协议

网络基础参考文章：http://www.study-area.org/network/network.htm

局域网络 (Local Area Network, LAN)、广域网络 (Wide Area Network, WAN)、城域网 (Metropolitan Area Network，MAN)

17、Linux 上的网络管理命令

(1)、网络参数的设定：ifconfig、ifup、ifdown、route、ip、ss

(2)、网络帧错和观察：ping、traceroute、netstat、host、dig(nslookup)反查主机名

(3)、远程连接指令：telnet、ftp、lftp

(4)、文字接口网页浏览：lynx、wget、curl

(5)、封包获取功能：tcpdump、wireshark(需安装：yum install ethereal ethereal-gnome -y)

ip 命令介绍：

(1)关于装置接口的相关设定：ip link

```
[root@linux ~]# ip [-s] link show <== 查阅该装置相关的信息 ,[-s]详细显示
[root@linux ~]# ip link set [device] [动作与参数] <== 设备参数的设定
##Example1: 查看所有的网络接口详细信息
[root@linux ~]# ip -s link show
##Example2: 启动、关闭与设定装置的相关信息
[root@linux ~]# ip link set eth0 up/down
##Example3: 更改指定设备 eth0 的 MTU 的预设值
[root@linux ~]# ip link set eth0 mtu 1000
```

(2)关于 IP 的相关设定：ip address

```
[root@linux ~]# ip address show <==查阅 IP 参数
[root@linux ~]# ip address [add|del] [IP 参数] [dev] [相关参数] <== 设备参数的设定
##Example1: 显示所有接口的 IP 参数:
[root@linux ~]# ip address show
##Example2: 新增一个接口，名称假设为 eth0:0，"broadcast +"代表使用默认的广播地址
[root@linux ~]# ip address add 192.168.5.1/24 broadcast + dev eth0 label eth0:0
##Example3: 删除设定的网络接口信息
[root@linux ~]# ip address del 192.168.5.1/24 dev eth0
```

(3)关于路由的相关设定：ip route

```
[root@linux ~]# ip route show <== 单纯的显示出路由的设定而已
[root@linux ~]# ip route [add|del] [IP|net] [gateway] [dev] <== 设备参数的设定
##Example1: 显示目前的路由信息:
[root@linux ~]# ip route show
##Example2: 增加路由，主要是本机直接可沟通的网域，不需要透过外部路由器
[root@linux ~]# ip route add 192.168.5.0/24 dev eth0
##Example3: 增加可以通往外部的路由，需透过 router。访问 10 需透过 5 这个路由
[root@linux ~]# ip route add 192.168.10.0/24 via 192.168.5.100 dev eth0
##Example4: 增加默认路由，默认路由仅设置一个
[root@linux ~]# ip route add default via 192.168.1.2 dev eth0
##Example5: 删除路由
```

```
[root@linux ~]# ip route del 192.168.10.0/24
```

ss 命令介绍：

```
[root@linux ~]#ss  [options]  [ FILTER ]
###选项：
#         -t：TCP 协议的相关连接
#         -u：UDP 相关的连接
#         -w：raw socket 相关的连接
#         -l：监听状态的连接
#         -a：所有状态的连接
#         -n：数字格式
#         -p：相关的程序及其 PID
#         -e：扩展格式信息
#         -m：内存用量
#         -o：计时器信息
###常用组合：    -tan,  -uan,  -tnl,  -unl,  -tunlp
##Example1：显示目标和源端口是 22 的连接
[root@linux ~]#ss  -tan   '(  dport = :22 or sport = :22  )'
##Example2：显示状态是已建立/监听(ESTABLISHED/TIME-WAIT)
[root@linux ~]#ss  -tan  state  ESTABLISHED/TIME-WAIT
```

netstat 命令介绍：

```
[root@linux ~]# netstat -[rn]         <==与路由有关的参数
[root@linux ~]# netstat -[antulpc]    <==与网络接口有关的参数
##选项说明
-r ：列出路由表(route table)
-a ：列出所有的联机状态，包括 tcp/udp/unix socket
-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 的文件名
-c ：可以设定几秒钟后自动更新一次，例：-c 1 #1 秒更新一次
```

tcpdump 命令介绍：

```
[root@linux ~]# tcpdump [-nn] [-i 接口] [-w 存储文件名称] [-c 次数] [-Ae][-qX] [-r
 档案] [所欲获取的数据内容]
参数介绍：
-nn：直接以 IP 及 port number 显示，而非主机名与服务名称
-i ：后面接要『监听』的网络接口，例如 eth0, lo, ppp0 等等的界面；
-w ：如果你要将监听所得的封包数据储存下来，后面接文件名
-c ：监听的封包数，如果没有这个参数， tcpdump 会持续不断的监听，直到使用者输入[ctrl]-c 为止
```

```
-A ：封包的内容以 ASCII 显示，通常用来捉取 WWW 的网页封包资料。
-e ：使用资料连接层 (OSI 第二层) 的 MAC 封包数据来显示；
-q ：仅列出较为简短的封包信息，每一行的内容比较精简
-X ：可以列出十六进制 (hex) 以及 ASCII 的封包内容，对于监听封包内容很有用
##Example1：监听某一个设备上的某一个端口的数据包
[root@linux ~]# tcpdump -i eth0 -nn port 21
##Example2：监听主机的 FTP 服务
[root@linux ~]# tcpdump -i lo -nn -X 'port 21'
##Example3:tcpdump 监听来自 eth0 且通讯协议为 port 22 目标来源为 192.168.1.10 的封包资料
[root@linux ~]# tcpdump -i eth0 -nn 'port 22 and src host 192.168.1.100'
```

附录：

1、常见的局域网(LAN)的拓扑结构：总线型、环形结构、星型结构、树形结构

2、常见的局域网(LAN)的传输介质：双绞线、同轴电缆、光纤、无线传输(采用无线电波、红外线、微波)

3、通信子网：网络中实现网络通信功能的设备及其软件的集合，通信设备、网络通信协议、通信控制软件是网络的内层，负责信息的传输。主要提供数据的传输，转接加工，变换等。

4、资源子网：实现资源共享的设备，由网络的服务器、工作站、共享的打印和其他设备及相关软件组成。

OSI七层模型	对应层级的协议
应用层	FTP、DNS、Telnet、SMTP、HTTP、WWW、NFS
表示层	JPEG、MPEG、ASII
会话层	NFS、SQL、NETBIOS、RPC
传输层	TCP、UDP、SPX
网络层	IP、ICMP、ARP、RARP、OSFP、IPX、RIP、IGRP(路由器)
数据链路层	PPP、FP、HDLC、VLAN、MAC(网桥、交换机)
物理层	RJ45、CLOCK、IEEE802.3(中继器、集线器)

5、网络通信的 三个地址：IP 地址(界定两个两个通信的主机)、MAC 地址(真正实现通信的地址)、进程地址=IP:Port(Socket 套接字)，其中套接字由计算机的内核提供，端口是用来标识本地主机上每一个进程的唯一的数字标识。

6、Linux 网络配置命令：

```
ifcfg 类别：
    ifconfig：配置 IP，NETMASK
    route：路由
    netstat：状态及统计数据查看
iproute2 类别：（重点）
```

```
    ip OBJECT(有如下三种)
        addr：地址和掩码;
        link：接口
        route：路由
    ss：状态及统计数据查看
CentOS 7：nm(Network Manager)家族
    nmcli：命令行工具
    nmtui：text window 工具
```

参考链接：http://www.178linux.com/43588

参考链接：http://www.178linux.com/64019