

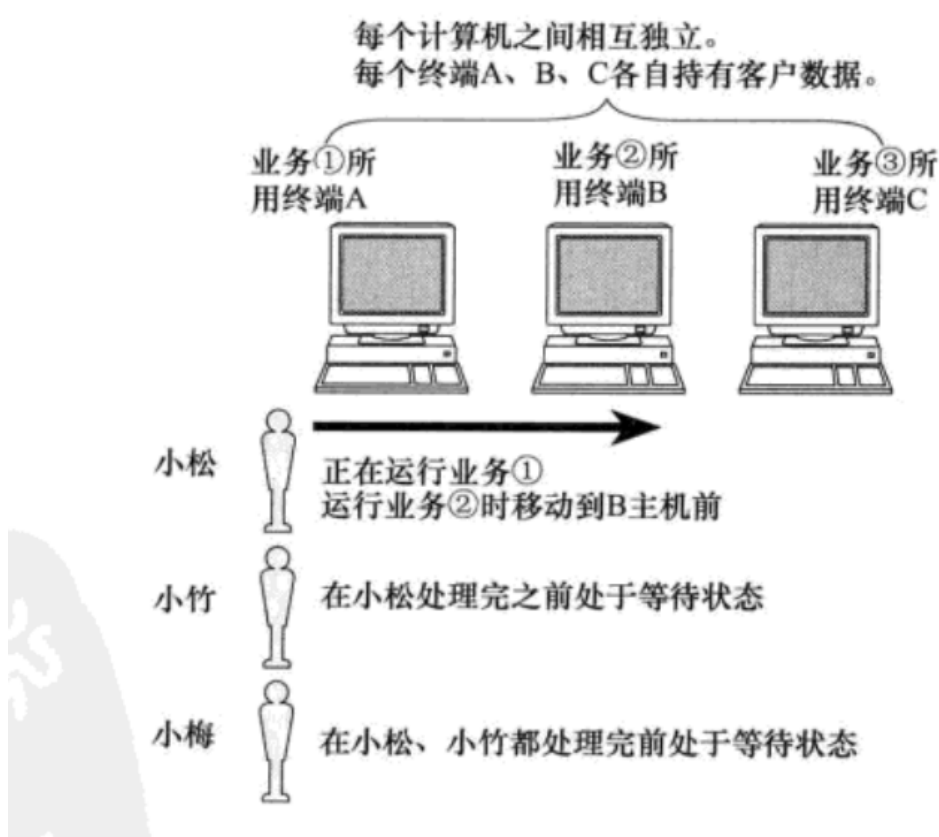
# 网络初识

## 目标

## 网络发展史

### 独立模式

独立模式：计算机之间相互独立；



### 网络互连

随着时代的发展，越来越需要计算机之间互相通信，共享软件和数据，即以多个计算机协同工作来完成业务，就有了网络互连。

网络互连：将多台计算机连接在一起，完成数据共享。

数据共享本质是**网络数据传输**，即计算机之间通过网络来传输数据，也称为**网络通信**。

根据网络互连的规模不同，可以划分为局域网和广域网。

### 局域网LAN

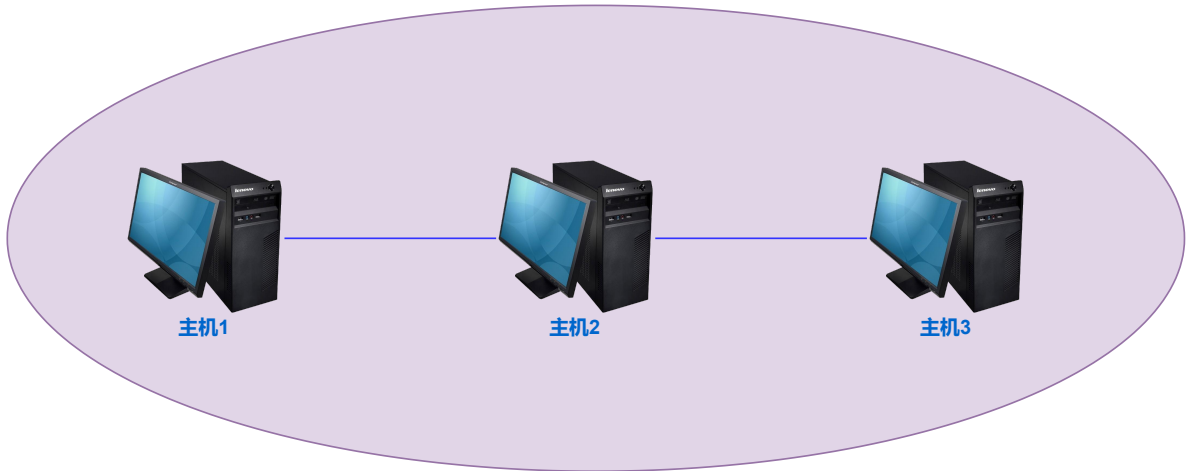
局域网，即 Local Area Network，简称LAN。

Local 即标识了局域网是本地，局部组建的一种私有网络。

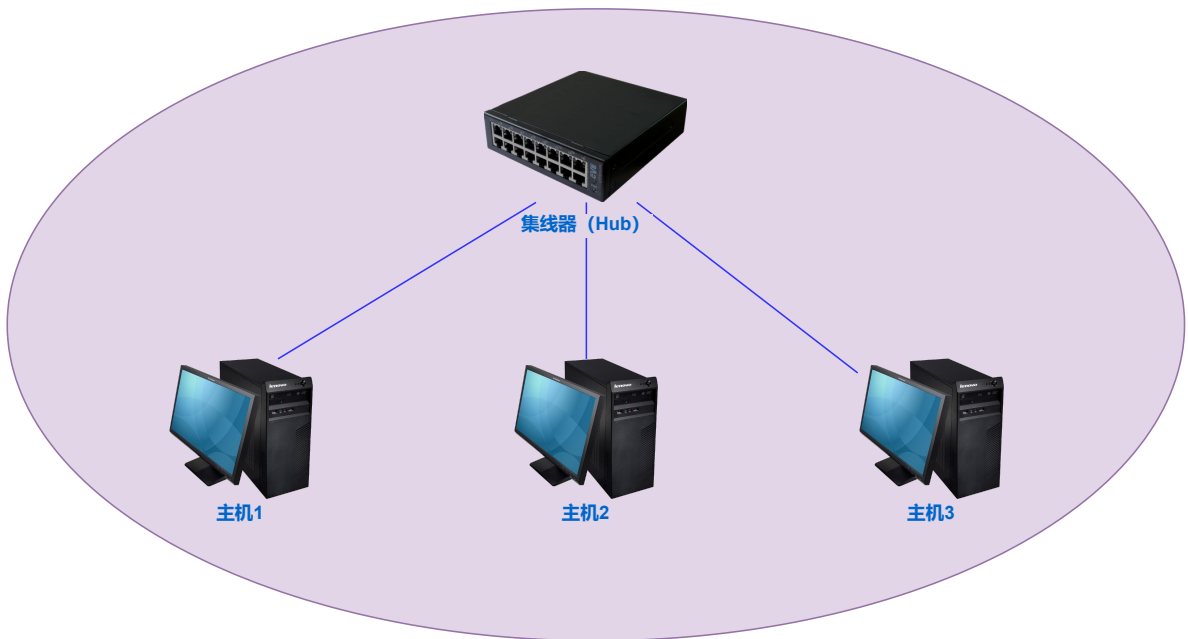
局域网内的主机之间能方便的进行网络通信，又称为内网；局域网和局域网之间在没有连接的情况下，是无法通信的。

局域网组建网络的方式有很多种：

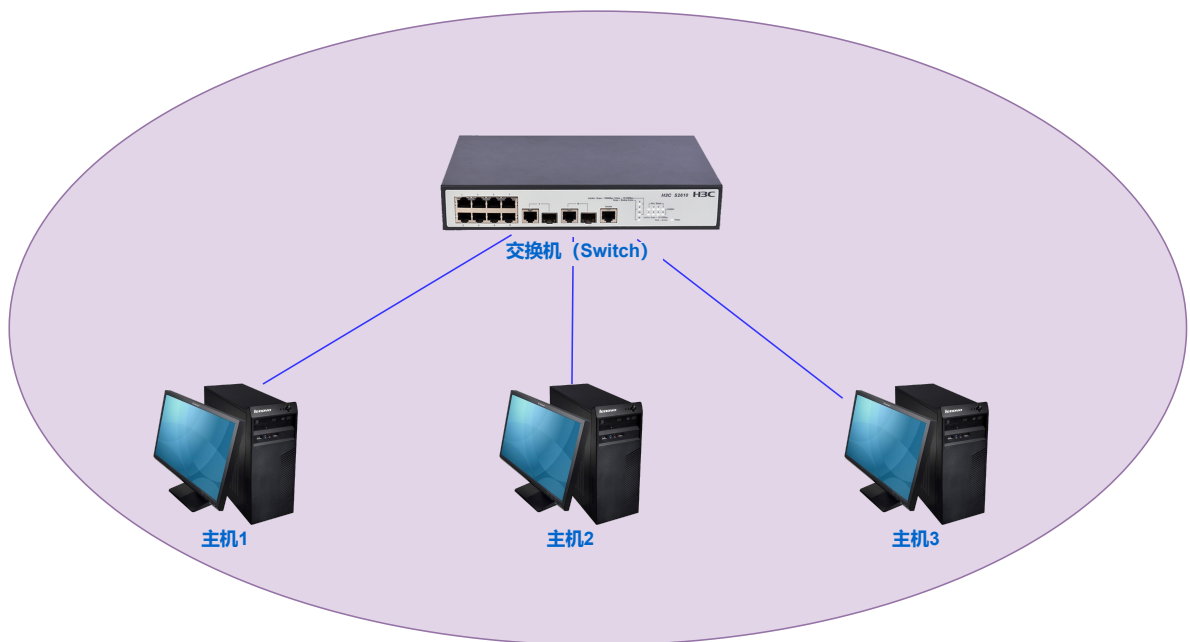
### (1) 基于网线直连



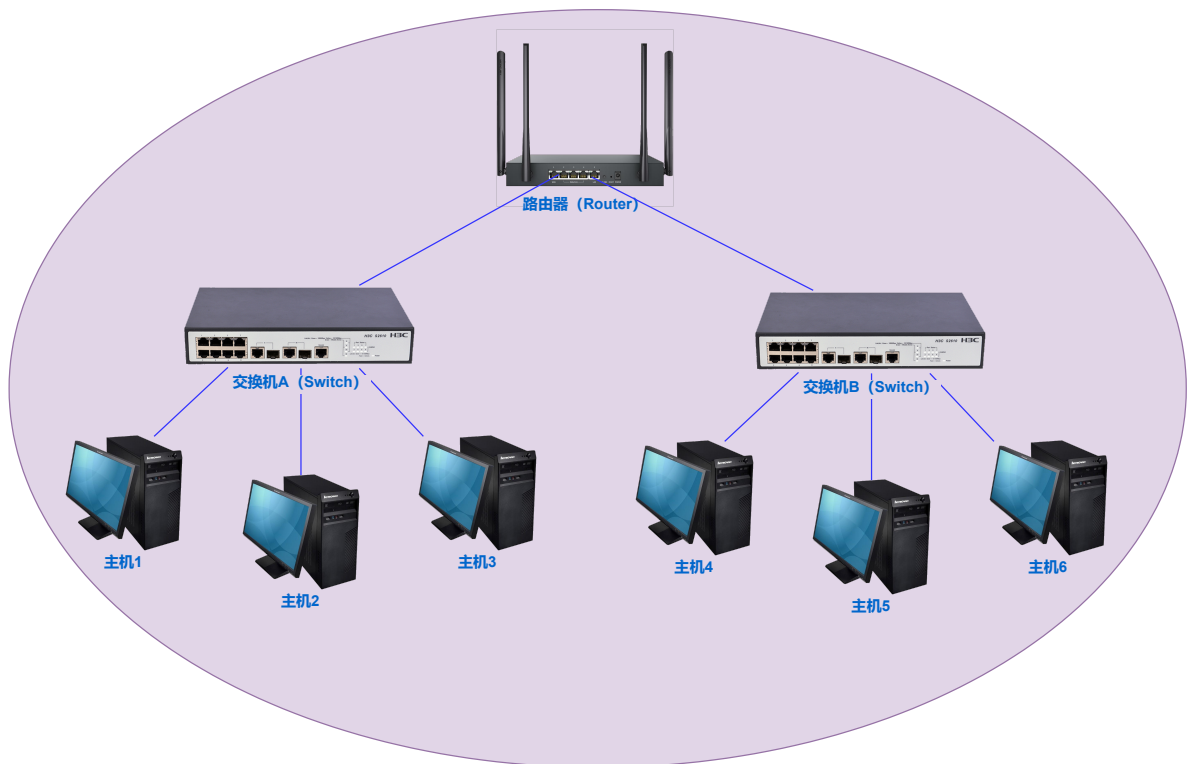
### (2) 基于集线器组建



### (3) 基于交换机组建



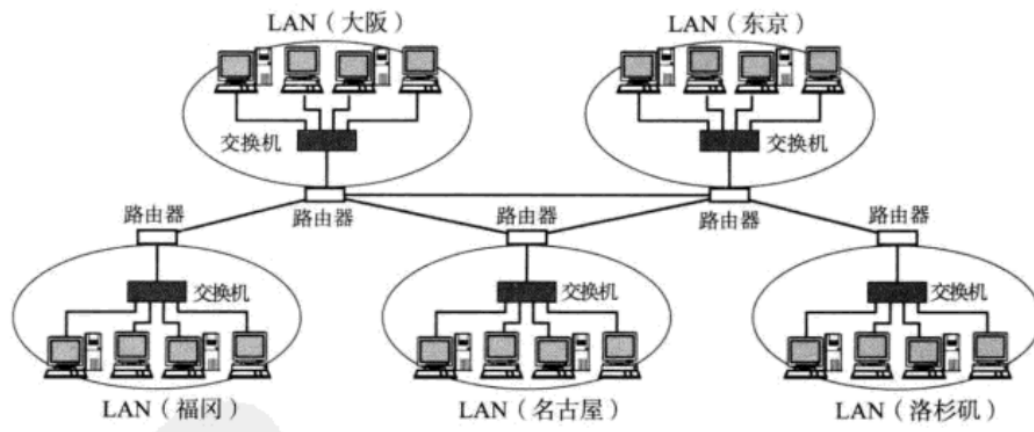
#### (4) 基于交换机和路由器组建



## 广域网WAN

广域网，即 Wide Area Network，简称WAN。

通过路由器，将多个局域网连接起来，在物理上组成很大范围的网络，就形成了广域网。广域网内部的局域网都属于其子网。



如果有北、中、南等分公司，甚至海外分公司，把这些分公司以专线方式连接起来，即称为“广域网”。

如果属于全球化的公共型广域网，则称为互联网（又称公网，外网），属于广域网的一个子集。

有时在不严格的环境下说的广域网，其实是指互联网。

所谓“局域网”和“广域网”只是一个相对的概念。比如，我们有“天朝特色”的广域网，也可以看做一个比较大的局域网。



你知道的太多了

## 网络通信基础

网络互连的目的是进行网络通信，也即是网络数据传输，更具体一点，是网络主机中的不同进程间，基于网络传输数据。

那么，在组建的网络中，如何判断到底是从哪台主机，将数据传输到那台主机呢？这就需要使用IP地址来标识。

## IP地址

### 概念

IP地址主要用于标识网络主机、其他网络设备（如路由器）的网络地址。简单说，**IP地址用于定位主机的网络地址**。

就像我们发送快递一样，需要知道对方的收货地址，快递员才能将包裹送到目的地。

### 格式

IP地址是一个32位的二进制数，通常被分割为4个“8位二进制数”（也就是4个字节），如：  
01100100.00000100.00000101.00000110。

通常用“点分十进制”的方式来表示，即 a.b.c.d 的形式（a,b,c,d都是0~255之间的十进制整数）。如：100.4.5.6。

## 特殊IP

127.\*的IP地址用于本机环回(loop back)测试，通常是127.0.0.1

本机环回主要用于本机到本机的网络通信（系统内部为了性能，不会走网络的方式传输），对于开发网络通信的程序（即网络编程）而言，常见的开发方式都是本机到本机的网络通信。

IP地址解决了网络通信时，定位网络主机的问题，但是还存在一个问题，传输到目的主机后，由哪个进程来接收这个数据呢？这就需要端口号来标识。

## 端口号

### 概念

在网络通信中，IP地址用于标识主机网络地址，端口号可以标识主机中发送数据、接收数据的进程。简单说：**端口号用于定位主机中的进程。**

类似发送快递时，不光需要指定收货地址（IP地址），还需要指定收货人（端口号）。

### 格式

端口号是0~65535范围的数字，在网络通信中，进程可以通过绑定一个端口号，来发送及接收网络数据。

### 注意事项

两个不同的进程，不能绑定同一个端口号，但一个进程可以绑定多个端口号。

了解：

一个进程启动后，系统会随机分配一个端口（启动端口）

程序代码中，进行网络编程时，需要绑定端口号（收发数据的端口）来发送、接收数据。

进程绑定一个端口号后，fork一个子进程，可以实现多个进程绑定一个端口号，但不同的进程不能绑定同一个端口号。

问题：

有了IP地址和端口号，可以定位到网络中唯一的一个进程，但还存在一个问题，网络通信是基于二进制0/1数据来传输，如何告诉对方发送的数据是什么样的呢？

网络通信传输的数据类型可能有多种：图片，视频，文本等。同一个类型的数据，格式可能也不同，如发送一个文本字符串“你好！”：如何标识发送的数据是文本类型，及文本的编码格式呢？

基于网络数据传输，需要使用协议来规定双方的数据格式。

## 认识协议

### 概念

协议，网络协议的简称，网络协议是网络通信（即网络数据传输）**经过的所有网络设备都必须共同遵从**的一组约定、规则。如怎么样建立连接、怎么样互相识别等。只有遵守这个约定，计算机之间才能相互通信交流。通常由三要素组成：

1. 语法：即数据与控制信息的结构或格式；

类似打电话时，双方要使用同样的语言：普通话

2. 语义：即需要发出何种控制信息，完成何种动作以及做出何种响应；

语义主要用来说明通信双方应当怎么做。用于协调与差错处理的控制信息。

类似打电话时，说话的内容。一方道：你瞅啥？另一方就得有对应的响应：瞅你咋的！

3. 时序，即事件实现顺序的详细说明。

时序定义了何时进行通信，先讲什么，后讲什么，讲话的速度等。比如是采用同步传输还是异步传输。

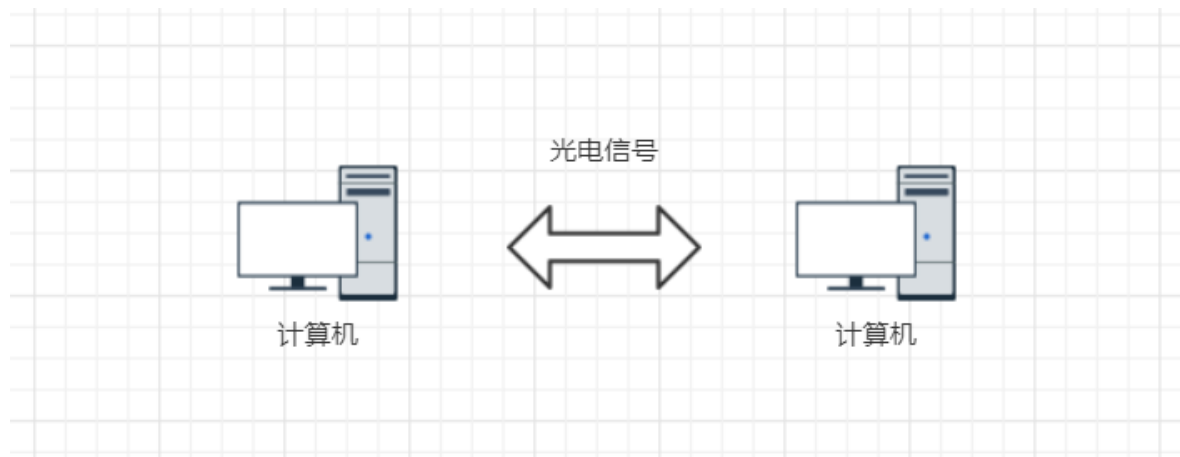
女生和男生的通话，总是由男生主动发起通话，而总是在男生恋恋不舍的时候，由女生要求结束通话。

协议 (protocol) **最终体现为在网络上传输的数据包的格式。**

## 作用

为什么需要协议？

就好比见网友，彼此协商胸口插支玫瑰花见面，这就是一种提前的约定，也可以称之为协议。



计算机之间的传输媒介是光信号和电信号。通过 "频率" 和 "强弱" 来表示 0 和 1 这样的信息。要想传递各种不同的信息，就需要约定好双方的数据格式。

- 计算机生产厂商有很多；
- 计算机操作系统，也有很多；
- 计算机网络硬件设备，还是有很多；
- 如何让这些不同厂商之间生产的计算机能够相互顺畅的通信？就需要有人站出来，约定一个共同的标准，大家都来遵守，这就是 **网络协议**；

## 知名协议的默认端口

系统端口号范围为 0 ~ 65535，其中：0 ~ 1023 为**知名端口号**，这些端口预留给服务端程序绑定广泛使用的应用层协议，如：

- 22端口：预留给SSH服务器绑定SSH协议
- 21端口：预留给FTP服务器绑定FTP协议
- 23端口：预留给Telnet服务器绑定Telnet协议
- 80端口：预留给HTTP服务器绑定HTTP协议
- 443端口：预留给HTTPS服务器绑定HTTPS协议

需要补充的是：

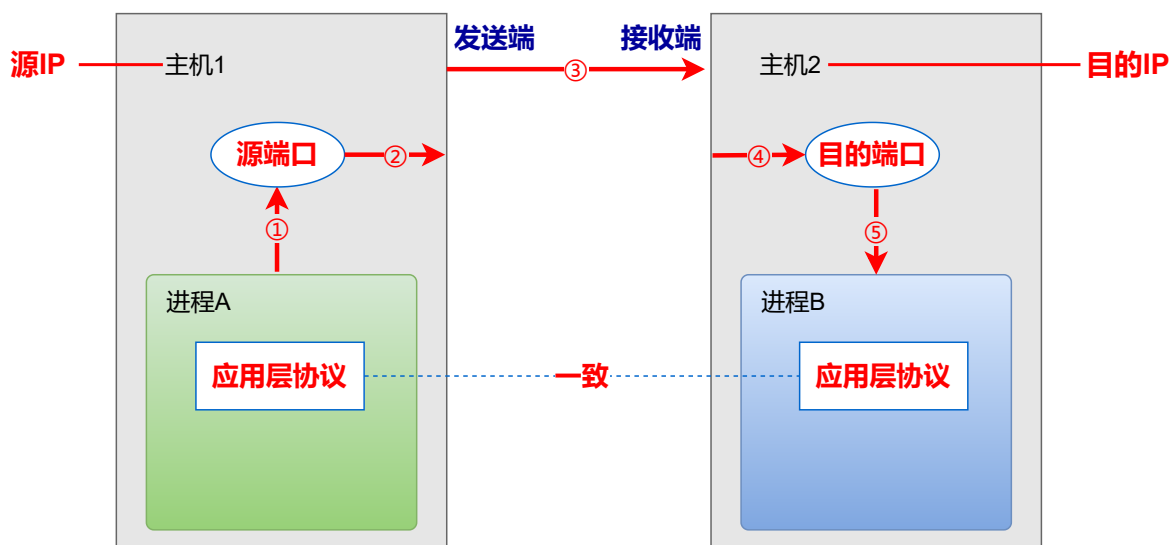
以上只是说明 0 ~ 1023 范围的知名端口号用于绑定知名协议，但某个服务器也可以使用其他 1024 ~ 65535 范围内的端口来绑定知名协议。

餐厅的VIP包房是给会员使用，但会员也可以不坐包房，坐其他普通座位。

## 五元组

在TCP/IP协议中，用五元组来标识一个网络通信：

1. 源IP：标识源主机
2. 源端口号：标识源主机中该次通信发送数据的进程
3. 目的IP：标识目的主机
4. 目的端口号：标识目的主机中该次通信接收数据的进程
5. 协议号：标识发送进程和接收进程双方约定的数据格式



五元组在网络通信中的作用，类似于发送快递：



可以在cmd中，输入 `netstat -ano` 查看网络数据传输中的五元组信息：

## 活动连接

| 协议  | 本地地址            | 外部地址            | 状态          |
|-----|-----------------|-----------------|-------------|
| TCP | 127.0.0.1:50629 | 127.0.0.1:50630 | ESTABLISHED |
| TCP | 127.0.0.1:50630 | 127.0.0.1:50629 | ESTABLISHED |
| TCP | 127.0.0.1:50631 | 127.0.0.1:50632 | ESTABLISHED |
| TCP | 127.0.0.1:50632 | 127.0.0.1:50631 | ESTABLISHED |
| TCP | 127.0.0.1:50633 | 127.0.0.1:50634 | ESTABLISHED |
| TCP | 127.0.0.1:50634 | 127.0.0.1:50633 | ESTABLISHED |
| TCP | 127.0.0.1:60530 | 127.0.0.1:60531 | ESTABLISHED |
| TCP | 127.0.0.1:60531 | 127.0.0.1:60530 | ESTABLISHED |
| TCP | 127.0.0.1:60532 | 127.0.0.1:62817 | ESTABLISHED |
| TCP | 127.0.0.1:60940 | 127.0.0.1:62811 | TIME_WAIT   |
| TCP | 127.0.0.1:60941 | 127.0.0.1:62810 | TIME_WAIT   |

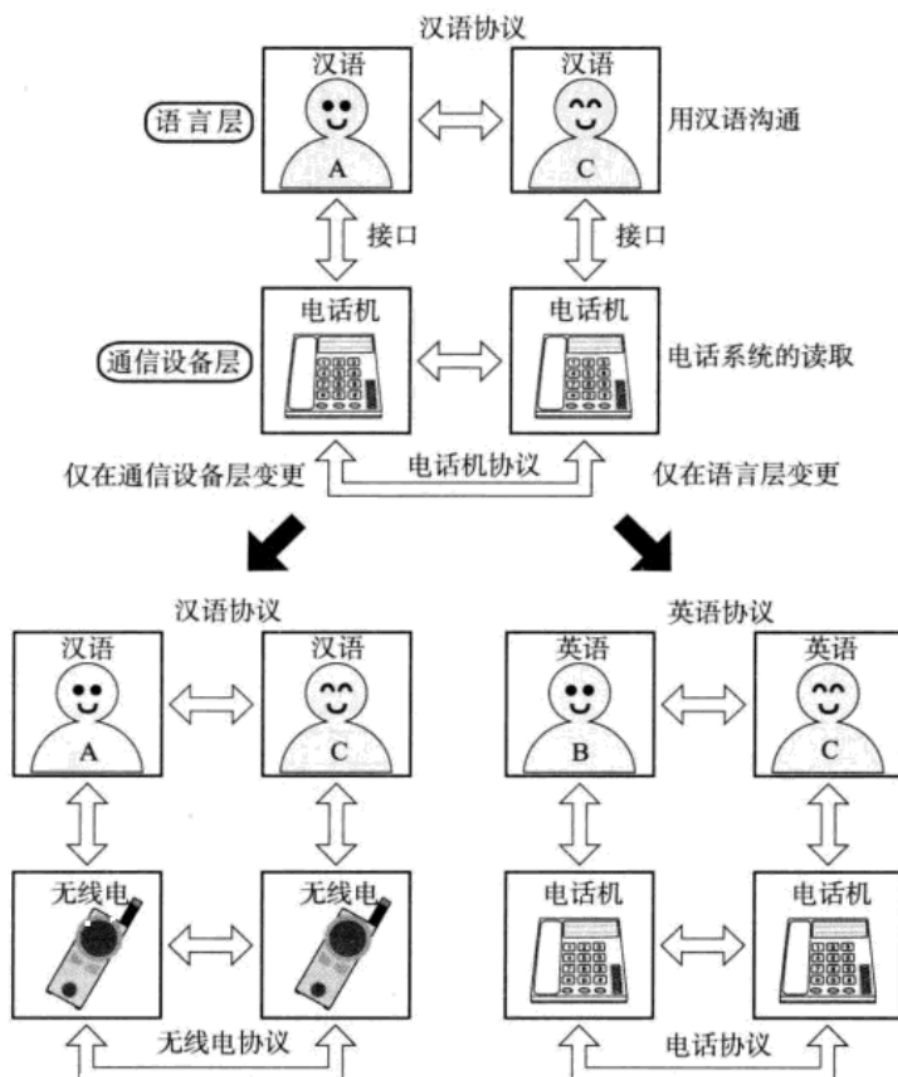
如果需要过滤（一般是通过端口号或进程PID过滤），可以使用 `netstat -ano | findstr 过滤字符串`

## 协议分层

对于网络协议来说，往往分成几个层次进行定义。

### 什么是协议分层

协议分层类似于打电话时，定义不同的层次的协议：





在这个例子中，我们的协议只有两层；但是实际的网络通信会更加复杂，需要分更多的层次。

## 分层的作用

为什么需要网络协议的分层？

分层最大的好处，类似于面向接口编程：定义好两层间的接口规范，让双方遵循这个规范来对接。

在代码中，类似于定义好一个接口，一方为接口的实现类（提供方，提供服务），一方为接口的使用类（使用方，使用服务）：

- 对于使用方来说，并不关心提供方是如何实现的，只需要使用接口即可
- 对于提供方来说，利用封装的特性，隐藏了实现的细节，只需要开放接口即可。

这样能更好的扩展和维护，如下图：



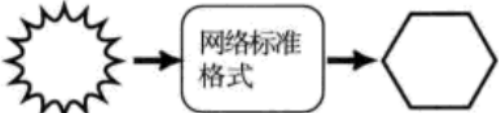
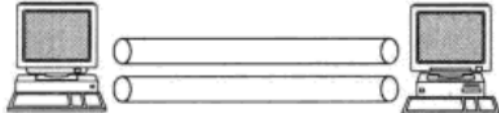
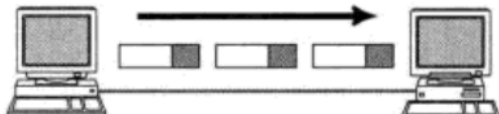
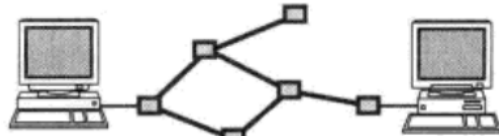
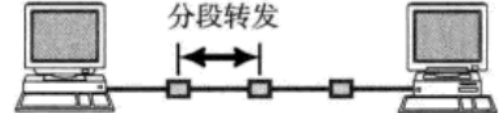

## OSI七层模型

**OSI**：即Open System Interconnection，开放系统互连

- OSI 七层网络模型是一个逻辑上的定义和规范：把网络从逻辑上分为了7层。
- OSI 七层模型是一种框架性的设计方法，其最主要的功能使就是帮助不同类型的主机实现数据传输；
- 它的最大优点是将**服务**、**接口**和**协议**这三个概念明确地区分开来，概念清楚，理论也比较完整。通过七个层次化的结构模型使不同的系统不同的网络之间实现可靠的通讯。



OSI 七层模型划分为以下七层:

|   | 分层名称  | 功    能                                      | 每层功能概览   |
|---|-------|---|--|
| 7 | 应用层   | 针对特定应用的协议。                                  | <p>针对每个应用的协议</p> <p>电子邮件 ↔ 电子邮件协议</p> <p>远程登录 ↔ 远程登录协议</p> <p>文件传输 ↔ 文件传输协议</p>  |
| 6 | 表示层   | 设备固有数据格式和网络标准数据格式的转换。                       |  <p>接收不同表现形式的信息，如文字流、图像、声音等</p>  |
| 5 | 会话层   | 通信管理。负责建立和断开通信连接（数据流动的逻辑通路）。<br>管理传输层以下的分层。 | <p>何时建立连接，何时断开连接以及保持多久的连接？</p>   |
| 4 | 传输层   | 管理两个节点之间的数据传输。负责可靠传输（确保数据被可靠地传送到目标地址）。      | <p>是否有数据丢失？</p>   |
| 3 | 网络层   | 地址管理与路由选择。                                  | <p>经过哪个路由传递到目标地址？</p>    |
| 2 | 数据链路层 | 互连设备之间传送和识别数据帧。                             | <p>数据帧与比特流之间的转换</p> <p>分段转发</p>                                        |
| 1 | 物理层   | 以“0”、“1”代表电压的高低、灯光的闪灭。<br>界定连接器和网线的规格。      | <p>0101 → [waveform] → 0101</p> <p>比特流与电子信号之间的切换</p>  <p>连接器与网线的规格</p> |

OSI 七层模型既复杂又不实用：所以 OSI 七层模型没有落地、实现。

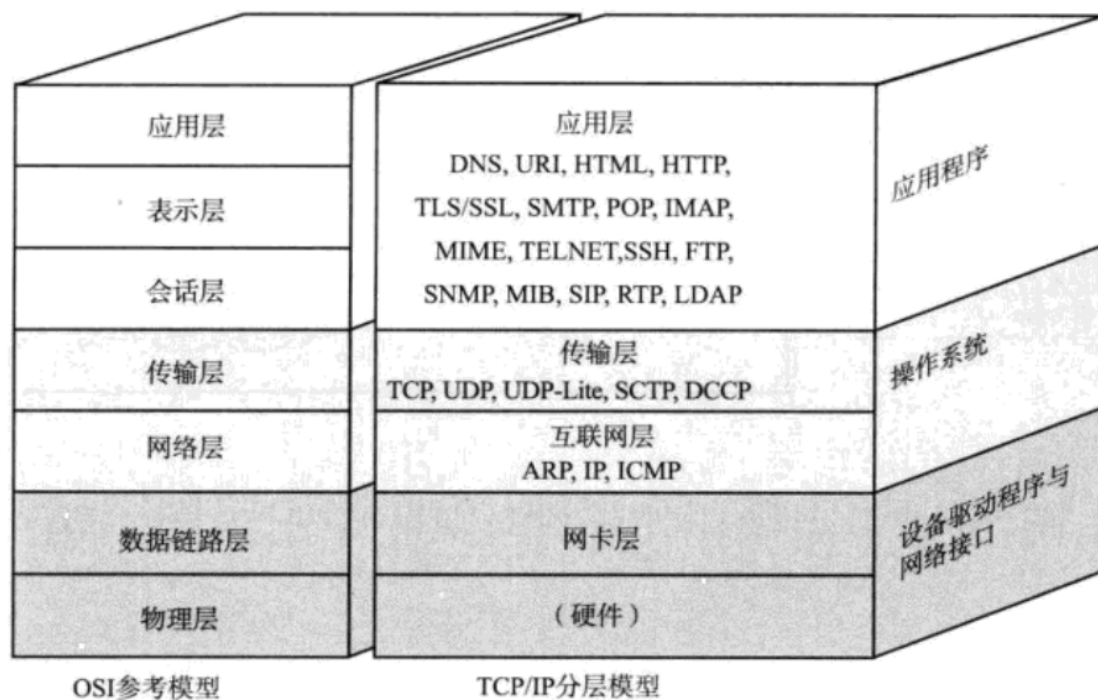
实际组建网络时，只是以 OSI 七层模型设计中的部分分层，也即是以下 TCP/IP 五层（或四层）模型来实现。

## TCP/IP五层（或四层）模型

TCP/IP是一组协议的代名词，它还包括许多协议，组成了TCP/IP协议簇。

TCP/IP通讯协议采用了5层的层级结构，每一层都呼叫它的下一层所提供的网络来完成自己的需求。

- **应用层**：负责应用程序间沟通，如简单电子邮件传输（SMTP）、文件传输协议（FTP）、网络远程访问协议（Telnet）等。我们的网络编程主要就是针对应用层。
- **传输层**：负责两台主机之间的数据传输。如传输控制协议（TCP），能够确保数据可靠的从源主机发送到目标主机。
- **网络层**：负责地址管理和路由选择。例如在IP协议中，通过IP地址来标识一台主机，并通过路由表的方式规划出两台主机之间的数据传输的线路（路由）。路由器（Router）工作在网路层。
- **数据链路层**：负责设备之间的数据帧的传送和识别。例如网卡设备的驱动、帧同步(就是说从网线上检测到什么信号算作新帧的开始)、冲突检测(如果检测到冲突就自动重发)、数据差错校验等工作。有以太网、令牌环网，无线LAN等标准。交换机（Switch）工作在数据链路层。
- **物理层**：负责光/电信号的传递方式。比如现在以太网通用的网线(双绞线)、早期以太网采用的的同轴电缆(现在主要用于有线电视)、光纤，现在的wifi无线网使用电磁波等都属于物理层的概念。物理层的能力决定了最大传输速率、传输距离、抗干扰性等。集线器（Hub）工作在物理层。



物理层我们考虑的比较少。因此很多时候也可以称为 TCP/IP四层模型。

### 参考资料

[TCP/IP四层模型和OSI七层模型的概念](#)

## 网络设备所在分层

- 对于一台**主机**，它的**操作系统**内核实现了从传输层到物理层的内容，也即是TCP/IP五层模型的**下四层**；
- 对于一台**路由器**，它实现了从网络层到物理层，也即是TCP/IP五层模型的**下三层**；
- 对于一台**交换机**，它实现了从数据链路层到物理层，也即是TCP/IP五层模型的**下两层**；
- 对于**集线器**，它只实现了**物理层**；

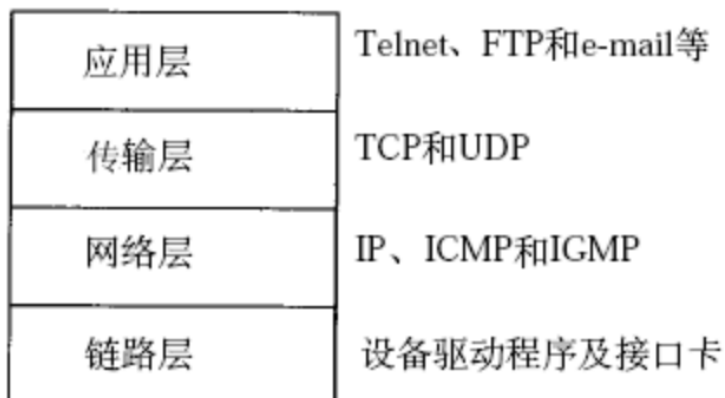
注意我们这里说的是传统意义上的交换机和路由器，也称为二层交换机（工作在TCP/IP五层模型的下两层）、三层路由器（工作在TCP/IP五层模型的下三层）。

随着现在网络设备技术的不断发展，也出现了很多3层或4层交换机，4层路由器。我们以下说的网络设备都是传统意义上的交换机和路由器。

## 网络分层对应

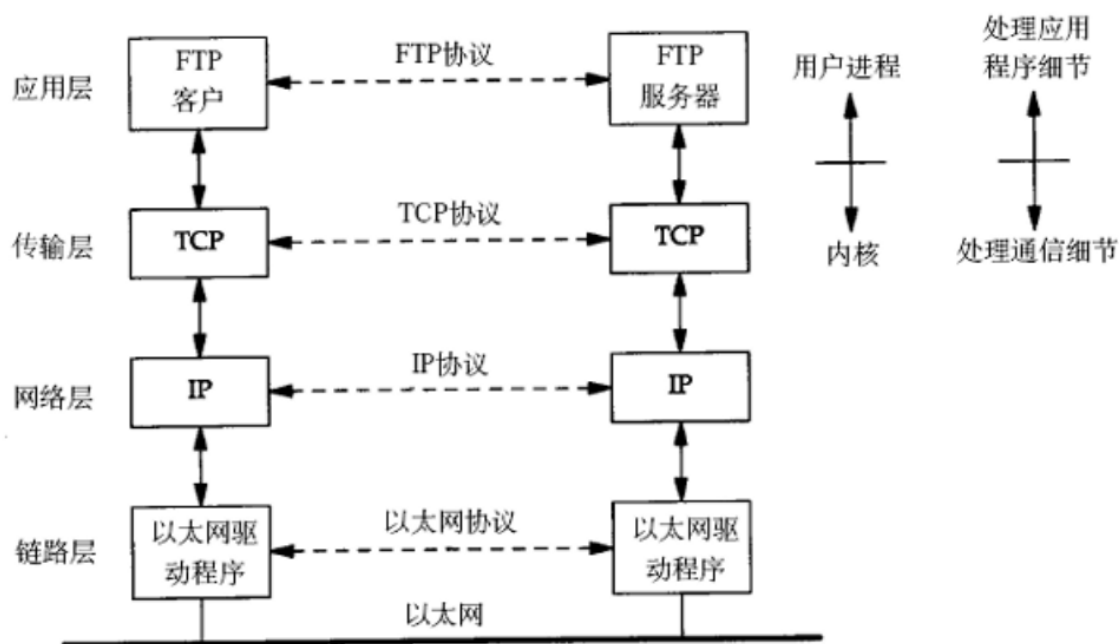
网络数据传输时，经过不同的网络节点（主机、路由器）时，网络分层需要对应。

以下为同一个网段内的两台主机进行文件传输：

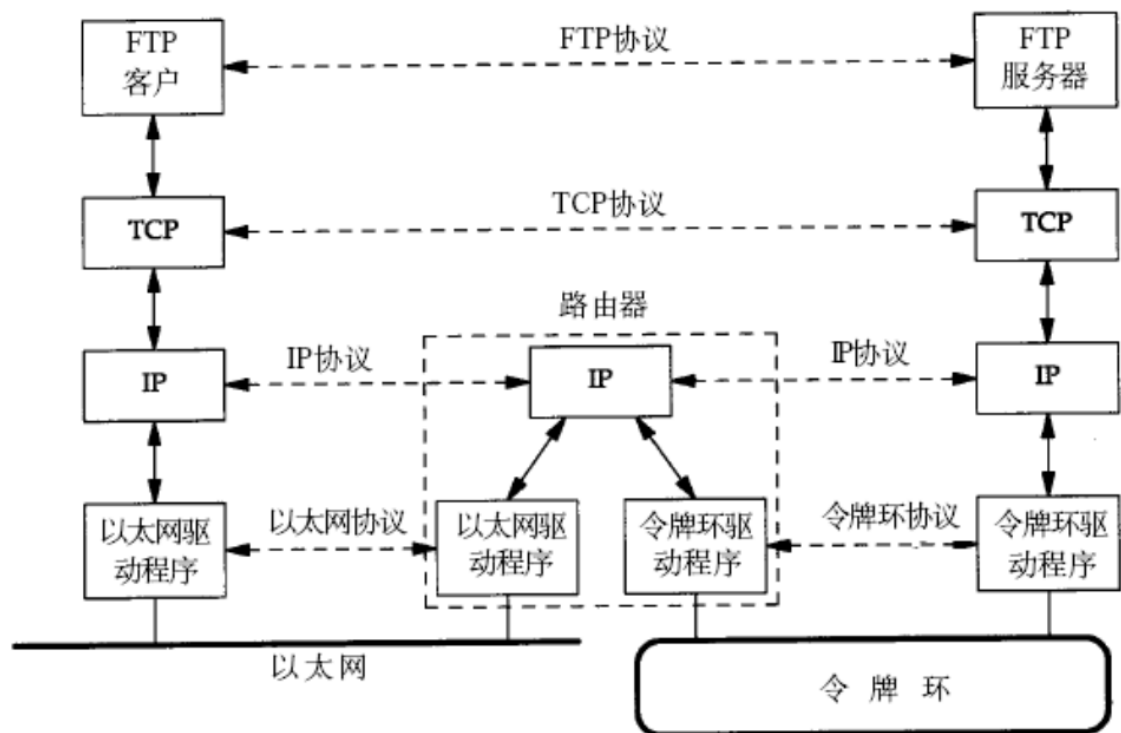


两台计算机通过TCP/IP协议通讯的过程如下所示

TCP/IP通讯过程



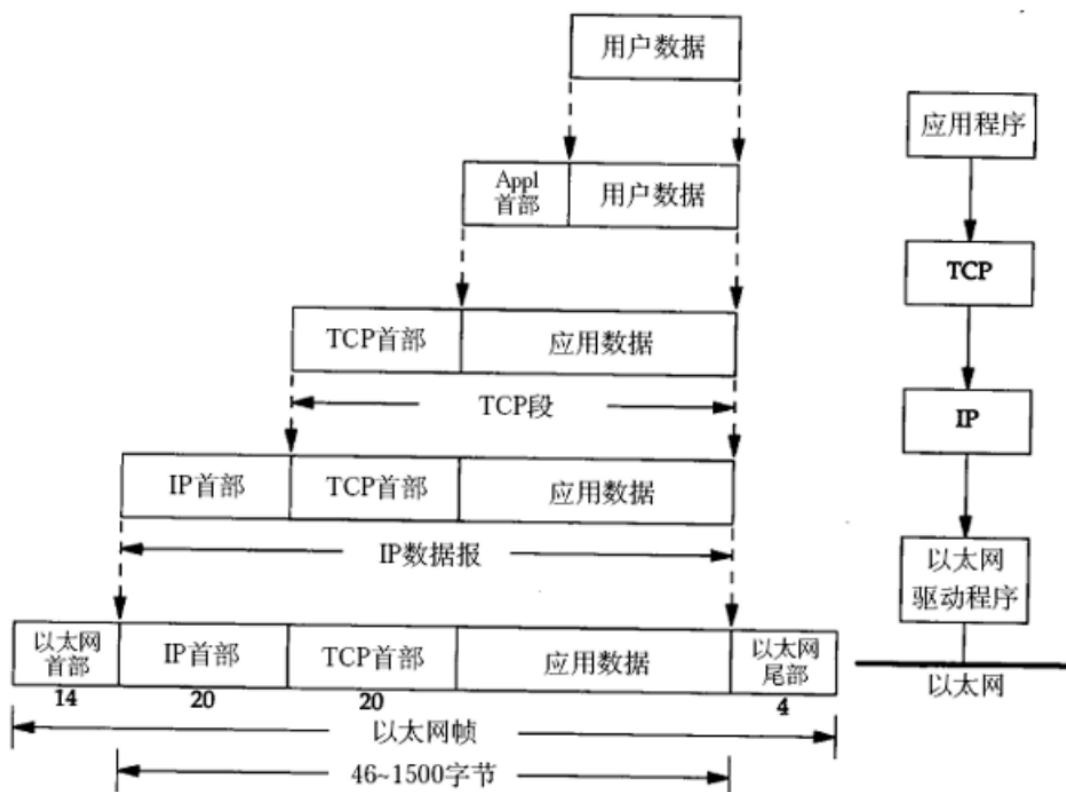
以下为跨网段的主机的文件传输：数据从一台计算机到另一台计算机传输过程中要经过一个或多个路由器



## 封装和分用

- 不同的协议层对数据包有不同的称谓，在传输层叫做段(segment)，在网络层叫做数据报(datagram)，在链路层叫做帧(frame)。
- 应用层数据通过协议栈发到网络上时，每层协议都要加上一个数据首部(header)，称为封装(Encapsulation)。
- 首部信息中包含了一些类似于首部有多长，载荷(payload)有多长，上层协议是什么等信息。
- 数据封装成帧后发到传输介质上，到达目的主机后每层协议再剥掉相应首部，根据首部中的"上层协议字段"将数据交给对应的上层协议处理。

下图为数据封装的过程



下图为数据分用的过程

