

计算机网络概述

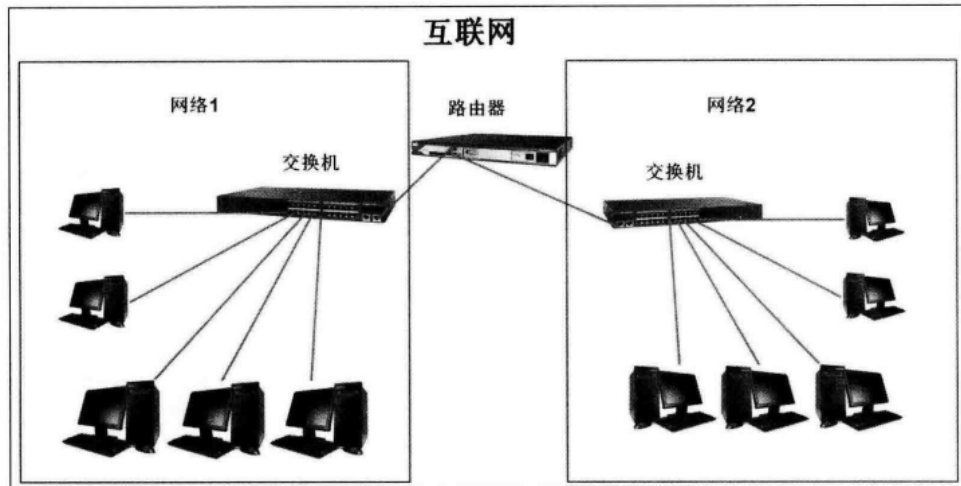
1 各种网络

网络（局域网）

一些计算机通过网线与交换机连接起来，这些计算机就构成了一个网络（同一个网段）

互联网（internet）

网络之间用路由器连接起来就构成了互联网，即网络的网络（不同网段连接在一起）

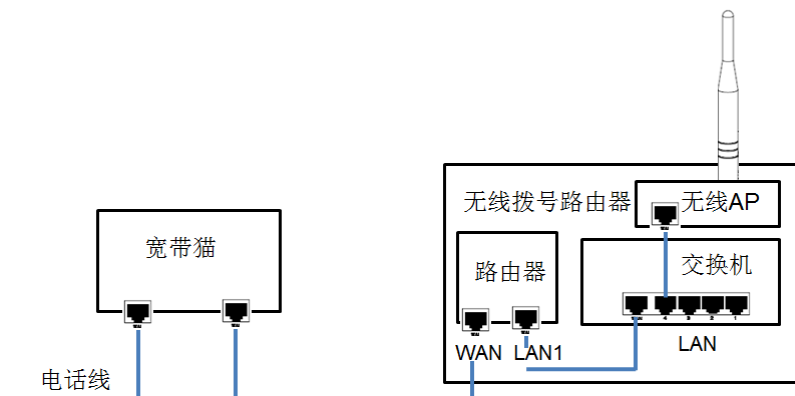


因特网（Internet）

全球最大的一个互联网

家庭互联网

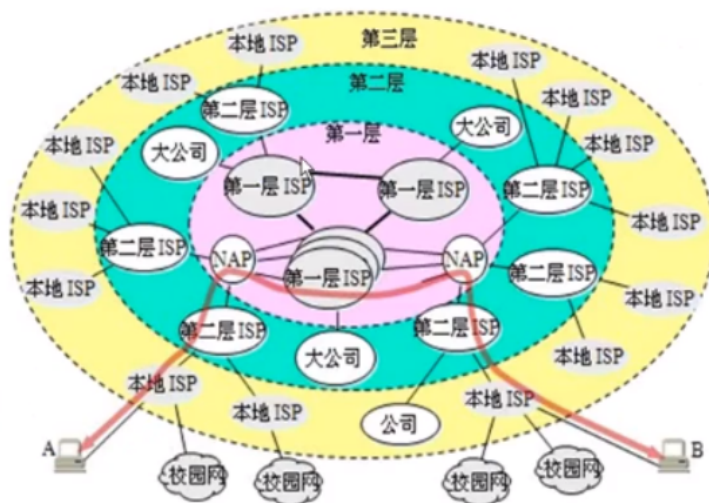
无线路由器相当于两个交换机和一个路由器的组合，无线AP相当于一个交换机将无线设备连成一个网络1，LAN口也相当于一个交换机将有线设备连成一个网络2，网络1和网络2同属一个网络，路由器将这个网络与广域网进行连接，形成互联网。



多层次ISP（网络服务运营商）因特网

- 计算机连入因特网需要IP地址，而IP地址就由ISP提供。
- IP地址总数有限，由某个组织统一管理，它将部分IP地址分配给第一层ISP，第一层ISP再将其拥有的IP地址分配给第二层ISP，第二层ISP再将其拥有的IP地址分配给第三层ISP，这就构成了多层次ISP。

- 计算机的IP地址可由任一层的ISP进行分配，因此计算机对因特网的访问速度常常取决于其所连入的ISP。
 - 不同ISP网络之间的访问速度比较慢，这可以由双线机房来解决，即机房同时接入多个ISP的网络
 - 不同地区的ISP网络之间的访问速度也比较慢



2 因特网组成

2.1 边缘部分

指主机之间的通信方式。

- 客户端服务器CS方式：多对一
- 对等P2P方式：多对多，每台计算机既是客户端，也是服务器

2.2 核心部分

指数据之间的传输方式。

1. 电路交换方式：一般用于电话，通信双方采用专线连接，通话后处于占线，其他电话无法打进来
 - 延迟低，用于数据量很大的实时性传输
 - 核心路由器之间可以使用电路交换
2. 报文交换方式：不做分组，将数据完整进行传输。
3. 分组交换方式：将数据分成多组，每组都附上目标地址进行单独传输，传输到目标主机后将多组数据去首部并重组，形成完整数据。
 - 通信双方不需要建立连接，因此线路可以复用，不会出现占线的情况
 - 路由器具有存储转发功能，即数据包可以在路由器暂时保存，在选定最佳路径后再进行转发
 - 延迟高，但实现了并发

3 计算机网络类型

局域网（LAN）：自己花钱买设备，带宽固定

广域网（WAN）：花钱租带宽，带宽和费用相关

- 按覆盖范围分
 - 覆盖范围大：广域网

覆盖范围小：局域网

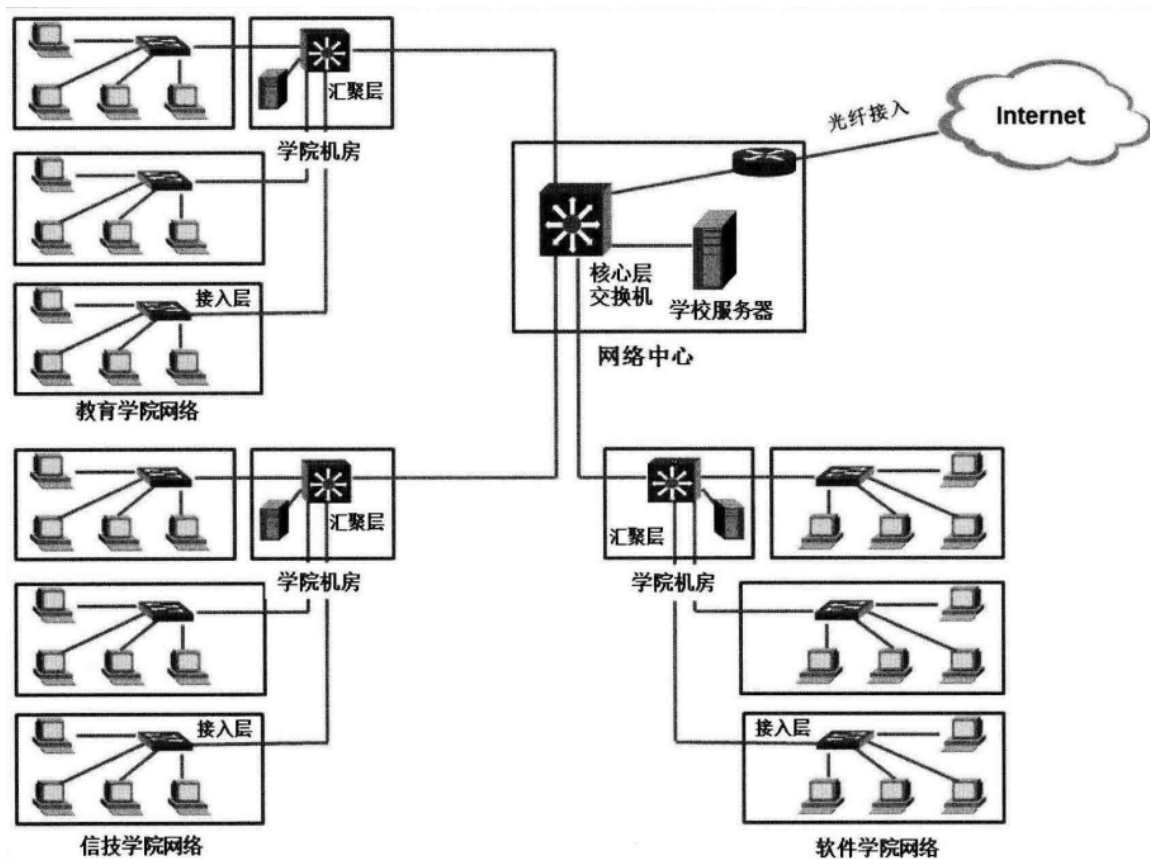
- 按所采用的技术分

采用了广域网技术：广域网

采用了局域网技术：局域网

4 局域网的设计

三层结构局域网设计：由接入层交换机、汇聚层交换机、核心层交换机三层结构组成。



交换机和路由器的区别：

- ① 交换机有路由功能，但只能用于实现局域网，不能用于连接广域网
- ② 路由器有广域网接口，可以用来实现局域网与广域网的连接

5 计算机网络性能指标

速率：单个数字信道上每秒传输的比特数量，即数据率。

带宽：单个数字信道上所能传输的最高速率。

- 带宽相同的两个设备才能进行通信，因为带宽不同每bit的传输信号长度不一样，就导致双方对信号的理解不同，就相当于编码和解码所采用的字符集不同，使得乱码产生。
- 两个设备之间带宽的具体数值通常由协商决定。
- ISP通过调整时钟频率调整用户的带宽。
- WIFI带宽随着信号的强弱变化会不断改变。

吞吐量：在单位时间内通过某个网络或接口的数据量，包括全部上传和下载的流量，即相当于所有信道上的速率之和。

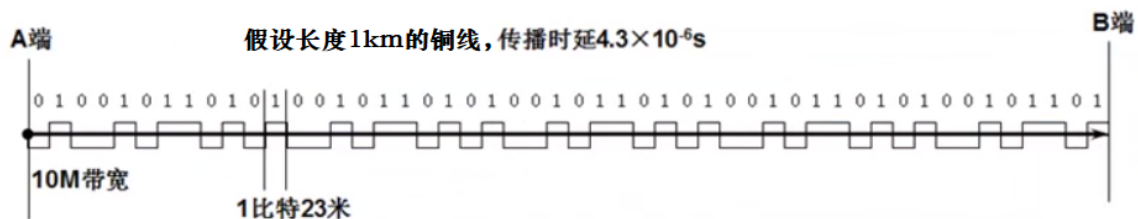
- 最大吞吐量和带宽和通信模式（全双工、半双工、单工）有关

- 单工：只能发不能收，单方向通信（电台、电视台）
- 半双工：同一时间只能是发或者收（对讲机、集线器联网）
- 全双工：同一时间可以收和发（打电话、交换机联网）

时延：数据（1个数据包或bit）从网络的一端传送到另一端所需要的时间。

- **发送时延：**从发送数据帧的第1个bit开始，到最后1个bit发送完毕所需要的时间，即数据帧长度 / 网卡带宽
- **传播时延：**从最后1bit发送完毕到最后1bit到达路由器接口所需要的时间，即信道长度 / 信道速率
 - 带宽只能改变发送时延，不能改变传播时延。只要传播介质和介质长度确定，传播时延就确定，也无法再改变。
- **排队时延：**多个数据包堵在一个路由器中，后来的数据包就要等待路由器处理完先到的数据包
- **处理时延：**路由器对数据包进行处理所需要的时间，比如首部校验、TTL减1、查找路由表为数据包选定转发出口等
- **总时延：**以上的总和

时延带宽积：指一个链路的最大数据承载量，即该链路包括了多少bit的数据，用传播延迟 × 带宽计算。该指标可用于求1比特信号的长度：



往返时间：从发送端发送数据开始，到发送端接收到来自接收端的确认总共经历的时间。

- 往返时间带宽积

利用率：指网络有百分之几的时间是被利用的（有数据通过）。

- 信道利用率：有数据通过的时间 / （有+无）数据通过的时间
- 网络利用率：全网络的信道利用率的加权平均值
- 网络利用率越高，时延也会越大（和堵车一个道理）

6 计算机网络的分层模型

6.1 分层的意义

- OSI参考模型将计算机通信过程分为7层，即分成了7个步骤，并规定了每个步骤实现的功能；
- 分层有利于标准化，也可以降低耦合性（即每层都是独立的，某层的变化不会影响其他层）；
- 每层为上一层提供服务。

6.2 OSI参考模型

----- 以下4层在通信主机上实现 -----

- **应用层：**指能够产生网络流量并和用户交互的应用程序。
- **表示层：**处理数据的加密、压缩和编码等问题，即数据预处理。

① 字符编码和解码：假设有abd三个字符要存储在计算机中，如果采用某一个字符集（a:00，b:01，c:10，d:11）编码，那么计算机存储的就是000111，这时如果用其他字符集进行解码就会出现乱码现象；

- ② 共享文件夹：处于同一个网络的计算机可以通过共享文件夹传输文件（CMD -> 输出目标主机IP地址）；
- ③ 抓包：一台计算机可以通过抓包捕获所有流经该计算机网络的通信数据，但一般不会捕获不属于自己的数据包，如果要捕获所有数据，要将网络接口设置为混杂模式；
- ④ 广播：当用户发送一个数据包时，如果为广播包，则可达到局域网中的所有机器，如果为单播包，则只能到达处于同一碰撞域中的机器。

- **会话层**：实现应用程序之间的会话建立，即主机端口之间的连接。

- ① netstat -nb：查看端口会话的命令，可用于查木马端口；
- ② netstat /?：查看具体命令用法；
- ③ 木马与病毒的区别：木马是为了窃取信息（因此需要联网），病毒是为了破坏主机。

- **传输层**：实现可靠传输（TCP，需要建立会话）和不可靠传输（UDP，不建立会话）以及流量控制。

- ① 可靠传输过程：发送端不断将数据分段放入缓存中，数据段也不断被发送给接收端，接收端收到某一个数据段后进行回复，发送端就将接收端确认收到的数据段从缓存中删除；
- ② 流量控制过程：当接收端来不及处理大量传输来的数据时，就会向发送端发送一个停车请求，让发送端慢点发送；
- ③ 丢包：当一个路由器接收太多数据包，忙不过来了，就会把部分数据包给扔掉。

----- 以下3层在网络设备上实现 -----

- **网络层**：给数据包填写IP地址，使得数据包可以准确到达目标主机，同时路由器也可以根据数据包的IP地址选择最佳路由路径（路由表）。

tracert www.baidu.com：跟踪路由路径

- **数据链路层**：数据包从源主机到达目标主机的实际线路被分成多个链路，数据包需要从链路的一个结点传输到下一个结点也需要地址，不然不知道要传输给哪个结点上，而数据链路层就负责把数据包加上结点的物理地址（MAC）并封装成帧，从而让数据可以沿着正确的路径传输。

- ① CSMA/CD协议：负责在广播链路上指定目标MAC地址；
- ② PPP协议：用于点到点链路，即无需指定MAC地址；
- ③ 广播链路：数据帧传递到交换机时，交换机会将数据帧发送给连接在交换机上的所有计算机，但每台计算机的网卡都会将数据帧上的MAC地址和自己的MAC地址进行比较，看是不是传给自己的，如果不是，就直接拒之门外，从而不会被计算机接收；
- ④ IP地址指出了数据的最终目的地，MAC地址指出了数据的下一跳目的地。如果数据在局域网内传输，那么就不需要IP地址，只需要MAC地址。

- **物理层**：提供传递介质（有线、无线），规定电压标准（即多少是高电平，多少是低电平）以及接口标准（比如网卡要接多少根网线）。

6.3 分层的应用

- 采用分层的思想解决问题

将各层的内容关联在一起容易产生耦合问题

- 网络排错

从底层向上逐一排查（网线插没插 -> 发送或接收字节是否为0 -> MAC地址有没有被屏蔽 -> 与交换机的带宽是否一致 -> ADSL是否欠费 -> IP地址或网关对不对 -> 浏览器配置是否正确）

- 网络安全

物理层安全：网线接口暴露、WIFI信号暴露（即给了其他人接入网络的机会）

数据链路层安全：ADSL拨号密码暴露、WIFI密码暴露（身份认证安全）

网络层安全：存储重要信息的主机与外网断开，只能在内网进行访问，从而保护信息不被窃取；
在路由器控制数据包转发

应用层安全：SQL注入漏洞、上传漏洞

6.4 TCP/IP四层模型

OSI参考模型只是表明了网络通信应该要有哪些东西，并不是具体的某些协议，也没有说明该如何实现，而TCP/IP协议体系结构则是OSI参考模型的具体实现，有具体的协议。

OSI参考模型	TCP/IP分层	TCP/IP协议组						本书按5层讲解
应用层	应用层	HTTP	FTP	SMTP	DNS	TFTP	DHCP	应用层
表示层								
会话层								
传输层	传输层	TCP			UDP			传输层
网络层	网络层	ARP	IP		ICMP	IGMP		网络层
数据链路层	网络接口层	CSMA/CD		HDCL	PPP	Frame Relay		数据链路层
物理层								物理层

6.5 层与层之间的通信过程

几个基本概念

实体：交换信息的硬件或软件进程

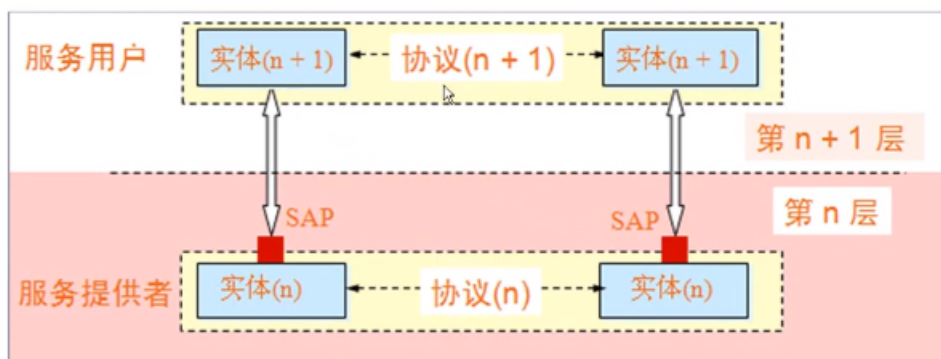
- 对等实体：采用相同协议的两个实体

协议：控制两个对等实体通信的规则

服务：下层向上层提供服务，上层需要使用下层提供的服务来实现本层功能

服务访问点：相邻两层实体间交换信息的地方

层与层之间的通信过程



6.6 各层协议的数据单元

应用层：传输数据单元PDU（电影、图片、MP3等）

运输层：运输层报文，即将PDU加上TCP或UDP协议的内容

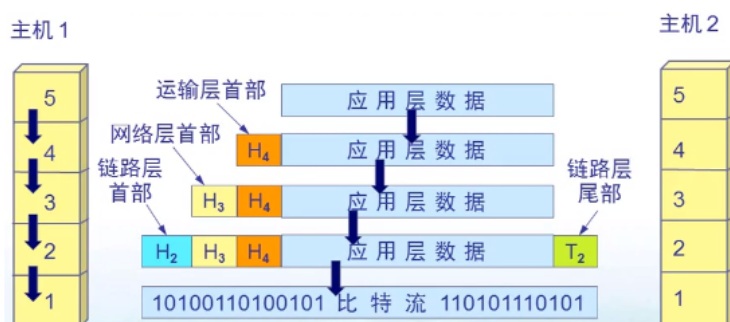
网络层：IP数据报（IP分组），即将运输层报文分组后加上IP地址

数据链路层：数据帧，即将数据包加上MAC地址

物理层：01（比特）

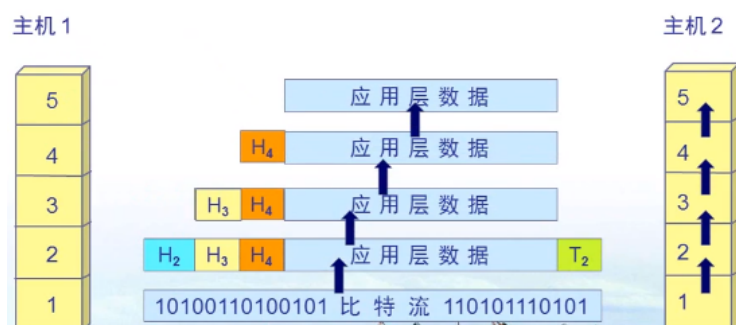
数据发送过程

发送数据时，数据每经过一层网络模型，都会被该层按照该层所采用的协议增添一些该协议规定的字段（即首部），然后将它们进行打包，形成一个新的数据包。



数据接收过程

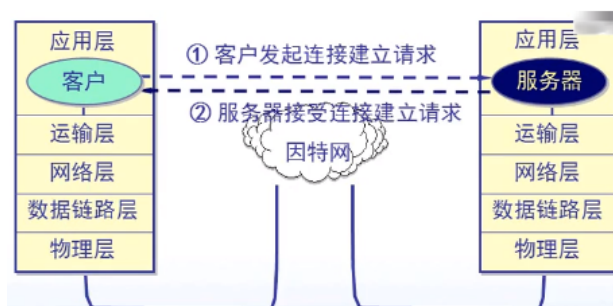
接收数据时，数据每经过一层网络模型，都会被该层按照该层所采用的协议进行解包，取出该层协议所规定的字段（即首部），然后进行校验，并把这些字段从数据包上删除后，发给下一层。



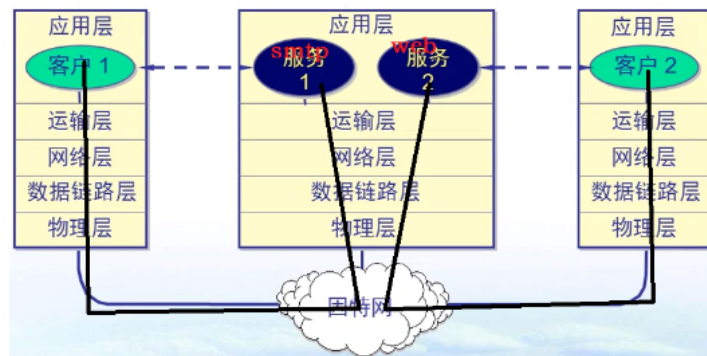
可见，模型的每一层都只关心该层协议所规定的的字段。

6.7 主机之间的通信过程

主机之间需要经过网络模型的各层才能进行通信，但在实际开发中，某一层的通信不需要考虑其他层的问题。



单个主机可以在多个端口同时运行多个服务进程，从而同时与多个主机进行通信。



7 网络协议

7.1 协议的样子

网络模型中每一层的协议都规定了该层数据的传输格式：

- 要有什么字段
- 各字段的位置和长度

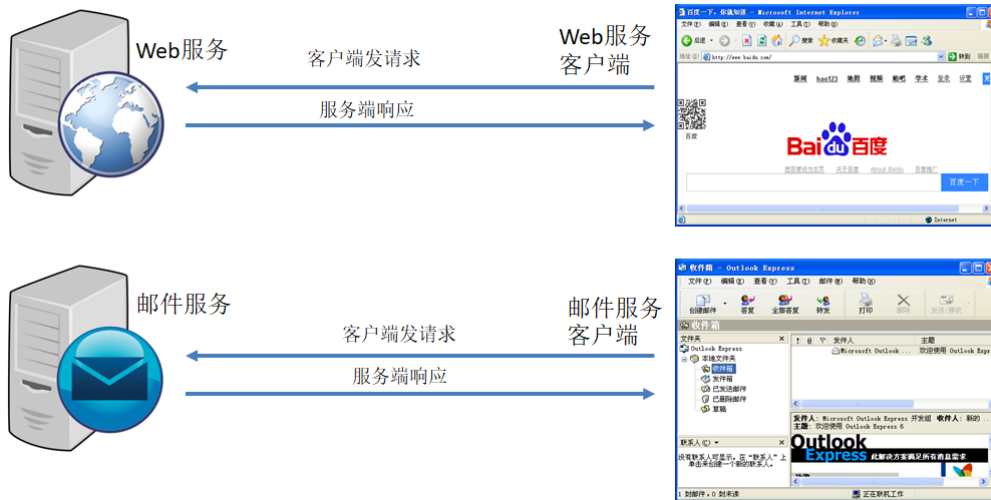
4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

7.2 应用层协议举例

应用层协议规定了用户可以使用哪些请求命令，即协议规定了用户的通信功能。相反，服务器要想提供哪些功能，就要采用有这些功能的协议进行数据传输。当然，服务器也可以禁用一些协议中规定的命令，使客户端只可以使用该协议的部分功能。

服务端程序

客户端程序



禁用FTP协议的某些命令

FTP 请求筛选

使用此功能可配置 FTP 服务的筛选规则。



7.3 协议的标准化

应用程序通信需要协议（应用层协议），通信双方（服务端程序、客户端程序）只有采用同一种协议才能够明白对方发送的数据是什么意思。

因此，通信双方可以定义私有协议，从而进行通信，但如果通信双方的程序由不同厂家开发，这就产生了多种多样的协议，导致只有采用同一种协议的两个厂家开发的程序才能互相通信。

如果将协议进行标准化，所有厂家都遵守这个协议，那么不同厂家开发的所有程序都可以进行相互通信。

7.4 协议的三要素

计算机通信使用的协议包含三要素：语法、语义和同步。

- 语法

定义协议中每种报文的格式：有哪些字段，字段是定长还是变长，如果是变长，字段分割符是什么，都要在协议中定义。一个协议有可能需要定义多种报文格式，比如ICMP协议，定义了ICMP请求报文格式、ICMP响应报文格式、ICMP差错报告报文格式。

- 语义

客户端能够向服务器发送那些请求（方法或命令），服务器有哪些响应（状态代码），每种状态代码代表什么意思。

- 同步

客户端访和服务器命令交互顺序，比如POP3协议，需要先验证用户身份才能收邮件。

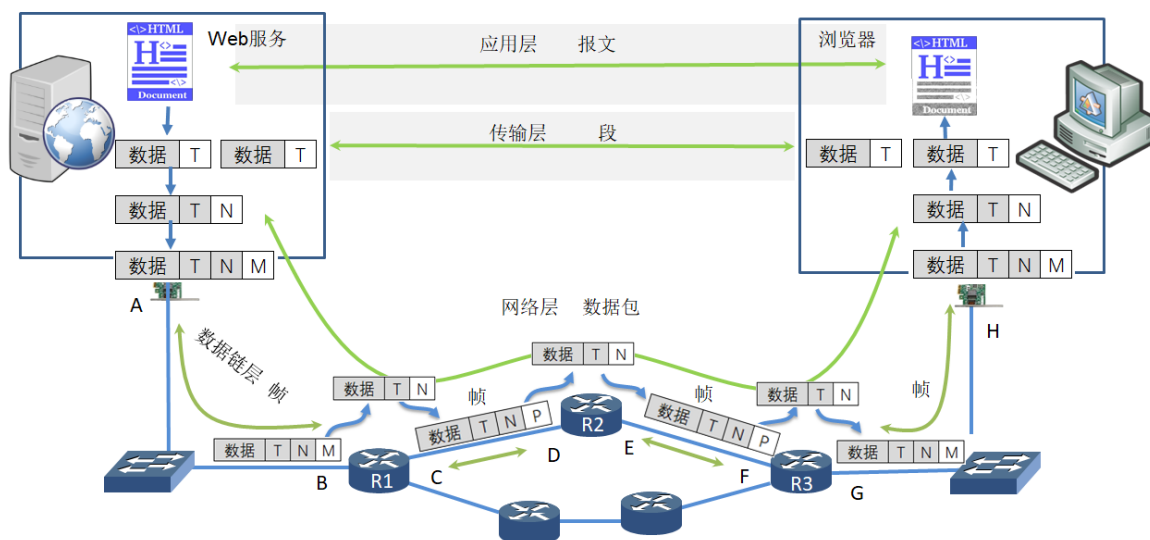
8 TCP/IP协议体系结构

体系结构就是指包括哪些协议以及他们之间的关系。

8.1 TCP/IP协议的分层

TCP/IP协议代表的是一组协议，这些协议包括应用层协议、传输层协议、网络层协议、数据链路层协议。

8.2 TCP/IP协议各层协议的作用



应用层协议

负责实现用户和服务端之间具体的交互功能，比如访问网页、发送邮件、上传文件、解析域名、分配IP地址等。

传输层协议

用于实现可靠传输，即在数据传输过程中可能会出现丢包、传输顺序错误等情况，而传输层的TCP协议就用来解决这类问题，比如丢包后重传、对数据包的顺序重排列等。

传输层协议会给数据添加一个首部，封装成数据段，这个首部用于数据的检验，从而实现可靠传输。

网络层协议

负责填快递单，即为数据包加上IP地址，使得路由器可以根据该地址找到目标主机。

网络层协议会给数据段加上网络层首部，即包含目标IP地址的一段信息，然后封装成数据包，路由器根据数据包的首部从而准确发送到目标主机。

数据链路层协议

用于添加不同链路之间的物理传输地址，比如a路由器到b路由器，则a路由器填写b路由器的物理地址，b路由器接收到后，去掉该地址首部，然后填写下一个路由器的地址，重新封装成数据帧进行发送。

数据链路层协议给数据包加上最近路由器的物理地址，封装成数据帧，而且每经过一个链路，数据包都要解包后重新封装成帧。

可以将数据链路层协议比作快递的每个接收点之间的传输协议。

各层相互独立

每层之间的功能相互独立，互不干扰，比如数据出现丢包、顺序错误等问题，应用层协议并不关心，丢了就丢了，这些问题只能由传输层协议管理。

封装和解封

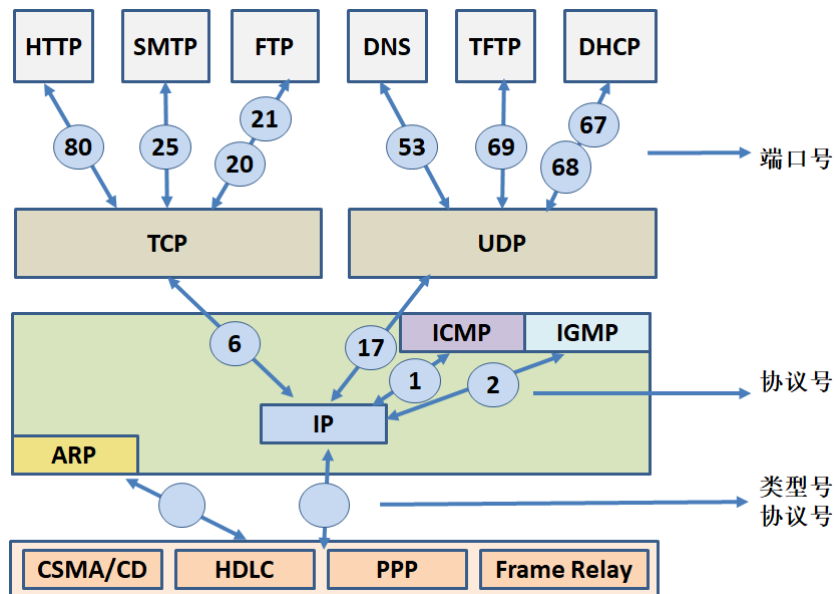
发送端加上传输层首部、网络层首部、数据链路层首部的过程叫做封装；接收端收到后，去掉数据链路层首部、去掉网络层首部、去掉传输层首部的过程叫做解封。

8.3 协议的标识字段

如果底层协议能够支持多种上层协议的封装，在底层协议必然要有一个字段用来指明封装的是哪种上层协议，否则接收端就不知道提交给上层的那个协议来进行下一步的处理。

各层协议用来标识上层协议的字段

从图中可以看出，网络层协议内还是可以继续分层的，但不是每一层的协议都需要用到，是否需要用到还取决于底层或上层采用了什么协议。



举例

Example of a packet capture in Wireshark showing the structure of a packet and its headers:

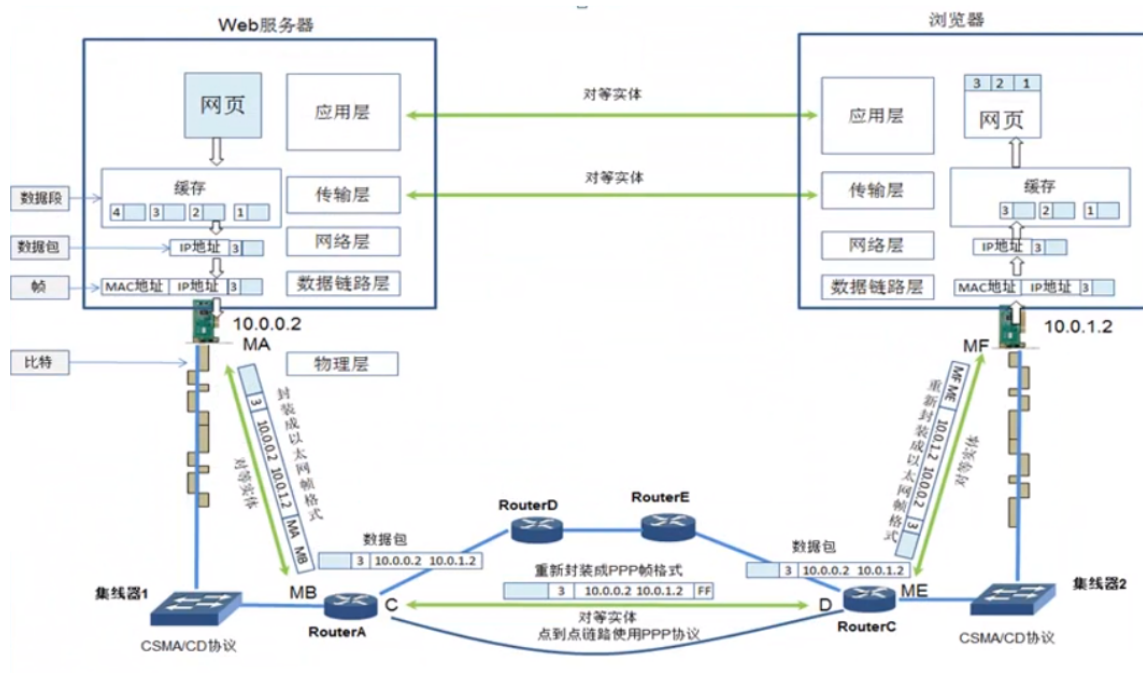
- Ethernet II:** Src: IntelCor_f9:d2:51 (00:db:df:f9:d2:51), Dst: PhicommS_96:a4:61. Type: IPv4 (0x0800).
- Internet Protocol Version 4:** Src: 192.168.2.163, Dst: 47.93.94.253. Header Length: 20 bytes (5). Total Length: 1396. Identification: 0x2ac7 (10951). Flags: 0x4000, Don't fragment. Time to live: 128. Protocol: TCP (6). Header checksum: 0x7917 [validation disabled]. Source: 192.168.2.163. Destination: 47.93.94.253.
- Transmission Control Protocol:** Src Port: 55182, Dst Port: 80, Seq: 4333, Ack: 5689. Window size: 259. Flags: 0x018 (PSH, ACK). Checksum: 0x2ad1 [unverified]. Urgent pointer: 0. [SEQ/ACK analysis]. [Timestamps]. TCP payload (1356 bytes). TCP segment data (1356 bytes).

Annotations in the image:

- 以太网数据链路层首部的Type字段标明网络层是什么协议** (Ethernet II header Type field indicates network layer protocol).
- IP协议首部的Protocol字段标明传输层是什么协议** (IP header Protocol field indicates transport layer protocol).
- TCP首部的Protocol字段标明应用层是什么协议** (TCP header Protocol field indicates application layer protocol).

8.4 TCP/IP协议的通信过程

TCP/IP协议的通信过程



各种网络设备的工作层次

集线器：只负责转发，实际效果就和网线一样，工作在物理层

交换机：工作在数据链路层，可以看懂数据链路层协议

路由器：工作在网络层，负责选择最佳路由线路

