一些重要的计算机网络协议（IP、TCP、UDP、HTTP） - 风之之 - 博客园

www.cnblogs.com

目录

一、计算机网络的发展历程

1、计算机网络发展

与其说计算机改变了世界，倒不如说是计算机网络改变了世界。彼时彼刻，你我都因网络而有了交集，岂非一种缘分？

计算机与网络发展大致经历如下过程：

年代	内容
20 世纪 50 年代	批处理时代
20 世纪 60 年代	分时系统时代
20 世纪 70 年代	计算机间通信时代
20 世纪 80 年代	计算机网络时代
20 世纪 90 年代	互联网普及时代
2000 年	以互联网为中心的时代
2010 年	无论何时何地一切皆 TCP/IP 的网络时代

2、计算机的模式变化

起初的计算机主要以“单机”形式存在，计算机之间没有通信可言，而网络的出现才真正让计算机变得与以往的工具不同，信息的共享和交流让计算机成为划时代的产物。网络按照规模可划分为：

- 局域网（**LAN**）：局域网是某一区域内由多台计算机互联成的计算机组。一般在几公里范围内。
- 广域网（**WAN**）：广域网的范围很大，几十公里到几千公里，可以是一个区域的计算机网也可能是整个国家的计算机网。

注：我们常见的**WLAN**是LAN的一种，称为无线局域网。在无线局域网WLAN发明之前，人们要想通过网络进行联络和通信，必须使用物理线缆-铜绞线或光纤进行物理线路连接，但这样的局域网转载和拆卸都很麻烦。随着网络的发展，人民利用射频（**Radio Frequency**； **RF**）的技术，使用电磁波进行网络架构，实现了无线局域网互联，让信息随身化。

二、各种协议

1、协议的重要性

计算机之间要实现通信，除了技术支持还需要一些规则来进行信息匹配，方能进行交流。不同的厂商生产不同的计算机，其CPU等内部构造不尽相同，就好比两个外国人，那么计算机或者说外国人之间需要实现通信或交流，那么两者之间就需要学会同一种交流规则，对于两个外国人来说，这种交流规则就是语言，而对于计算机来说，这种交流规则就是各种协议。协议的出现让不同厂商之间生产的计算机只要能够支持同一种协议就能实现正常通信，进行交流。

下面是一些以前常用的协议（因为现在基本都是TCP/IP协议）：

网络体系结构	协　议	主要用途
TCP/IP	IP，ICMP，TCP，UDP，HTTP，TELNET，SNMP，SMTP…	互联网、局域网
IPX/SPX（NetWare）	IPX，SPX，NPC…	个人电脑局域网
AppleTalk	DDP，RTMP，AEP，ATP，ZIP…	苹果公司现有产品的局域网
DECnet	DPR，NSP，SCP…	前 DEC 小型机
OSI	FTAM，MOTIS，VT，CMIS/CMIP，CLNP，CONP…	—
XNS	IDP，SPP，PEP…	施乐公司网络

2、协议的标准：**OSI**七层模型

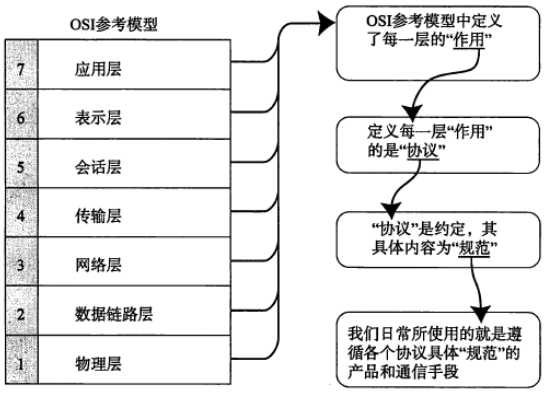
计算机通信诞生之初，系统化与标准化未收到重视，不同厂商只出产各自的网络来实现通信，这样就造成了对用户使用计算机网络造成了很大障碍，缺乏灵活性和可扩展性

为解决该问题，ISO（国际标准化组织）制定了一个国际标准**OSI**（开放式通信系统互联参考模型）。

注：TCP/IP并非ISO指定，是由IETF（国际互联网工程任务组）建议、致力推进标准化的一种协议，其中，大学等研究机构和计算机行业是推动标准化的核心力量，现已成为业界标准协议。协议的标准化也推动了计算机网络的普及。

ISO对协议进行分层，每个分层可以独立使用，其实系统中某些分层发生变化，也不会影响整个系统，因此可以构造一个扩展性和灵活性都比较强的系统；此外，通过分层可以细分通信功能，更易于单独实现每个分层的协议，界定各个分层的具体责任和义务。

但过分模块化，处理变得更加沉重，以及每个模块都不得不事先相似的处理逻辑等。



以下是各层功能描述：

	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。	针对每个应用的协议 电子邮件 ↔ 电子邮件协议 远程登录 ↔ 远程登录协议 文件传输 ↔ 文件传输协议
6	表示层	设备固有数据格式和网络标准数据格式的转换。	网络标准格式 接收不同表现形式的信息，如文字流、图像、声音等
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）。 管理传输层以下的分层。	何时建立连接，何时断开连接以及保持多久的连接？
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）。	是否有数据丢失？
3	网络层	地址管理与路由选择。	经过哪个路由传递到目标地址？
2	数据链路层	互连设备之间传送和识别数据帧。	0101 数据帧与比特流之间的转换 分段转发
1	物理层	以“0”、“1”代表电压的高低、灯光的闪灭。 界定连接器和网线的规格。	0101 → 0101 比特流与电子信号之间的切换 连接器与网线的规格

1. 物理层协议：

   负责0、1 比特流（0/1序列）与电压的高低、逛的闪灭之间的转换。规定了激活、维持、关闭通信端点之间的机械特性、电气特性、功能特性以及过程特性；该层为上层协议提供了一个传输数据的物理媒体，只是说明标准。

   在这一层，数据的单位称为比特（**bit**）（注：bit和字节Byte，我们常说的1字节8位2进制即：1B=8bit）。属于物理层定义的典型规范代表包括：EIA/TIA RS-232、EIA/TIA RS-449、V.35、RJ-45、fddi令牌环网。
2. 数据链路层协议：

   负责物理层面上的互联的、节点间的通信传输（例如一个以太网项链的2个节点之间的通信）；该层的作用包括：物理地址寻址、数据的成帧、流量控制、数据的检错、重发等。

在这一层，数据的单位称为帧（**frame**）。数据链路层协议的代表包括：ARP、RARP、SDLC、HDLC、PPP、STP、帧中继等。

3. 网络层协议：
   将数据传输到目标地址；目标地址可以使多个网络通过路由器连接而成的某一个地址，主要负责寻找地址和路由选择，网络层还可以实现拥塞控制、网际互连等功能。
   在这一层，数据的单位称为数据包（**packet**）。网络层协议的代表包括：IP、IPX、RIP、OSPF等。
4. 传输层协议（核心层）：
   传输层是OSI中最重要、最关键的一层,是唯一负责总体的数据传输和数据控制的一层；传输层提供端到端的交换数据的机制，检查分组编号与次序，传输层对其上三层如会话层等，提供可靠的传输服务,对网络层提供可靠的目的地站点信息主要功能。在这一层，数据的单位称为数据段（segment）。主要功能：

   ①：为端到端连接提供 传输服务。

   ②：这种传输服务分为可靠和不可靠的,其中Tcp是典型的可靠传输,而Udp则是不可靠传输。

   ③：为端到端连接提供流量控制,差错控制,服务质量(Quality of Service,QoS)等 管理服务。

   包括的协议如下：

   TCP：传输控制协议，传输效率低，可靠性强。

   UDP：用户数据报协议，适用于传输可靠性要求不高，数据量小的数据。

   DCCP、SCTP、RTP、RSVP、PPTP等协议。

5. 会话层协议：
   负责建立和断开通信连接（数据流动的逻辑通路），记忆数据的分隔等数据传输相关的管理。
6. 表示层协议：将数据格式转换为标准格式
   　　将应用处理的信息转换为适合网络传输的格式，或将来自下一层的数据转换为上层能够处理的格式；主要负责数据格式的转换，确保一个系统的应用层信息可被另一个系统应用层读取。具体来说，就是将设备固有的数据格式转换为网络标准传输格式，不同设备对同一比特流解释的结果可能会不同；因此，主要负责使它们保持一致。
7. 应用层协议：

   ①：超文本传输协议**HTTP**： 这是一种最基本的客户机/服务器的访问协议；浏览器向服务器发送请求，而服务器回应相应的网页。

   ②：文件传送协议**FTP** ：提供交互式的访问，基于客户服务器模式，面向连接 使用TCP可靠的运输服务。主要功能:减少/消除不同操作系统下文件的不兼容性。

⑬：远程登录协议TELNET：客户服务器模式，能适应许多计算机和操作系统的差异，网络虚拟终端NVT的意义。

④：简单邮件传送协议**SMTP** ：Client/Server模式，面向连接。基本功能：写信、传送、报告传送情况、显示信件、接收方处理信件。

⑤：**DNS**域名解析协议：DNS是一种用以将域名转换为IP地址的Internet服务。

⑥：简单文件传送协议TFTP：客户服务器模式，使用UDP数据报，只支持文件传输，不支持交互，TFTP代码占内存小。

⑦：简单网络管理协议（SNMP）: SNMP模型的4个组件：被管理结点、管理站、管理信息、管理协议。SNMP代理：运行SNMP管理进程的被管理结点。

⑧DHCP动态主机配置协议: 发现协议中的引导文件名、空终止符、属名或者空,DHCP供应协议中的受限目录路径名 Options –可选参数字段，参考定义选择列表中的选择文件。

PS：其实协议分层只是为了更好地理解我运用协议的作用，而不是绝对的分层，有的层之间协议也是可以共用的，特别是会话层、表示层和应用层这三层。

三、**TCP/IP**协议

1、介绍

Transmission Control Protocol/Internet Protocol的简写，中译名为传输控制协议/因特网互联协议，又名网络通讯协议，是Internet最基本的协议、Internet国际互联网络的基础，由网络层的**IP**协议和传输层的**TCP**协议等组成（当然还有其他后来发展起来的网络协议，还包括 ARP，ICMP，IGMP，UDP，以及让域名访问成为可能的DNS，以及电脑/手机可以自动获取IP地址的DHCP。当然还有形形色色的应用层的协议如 HTTP / SMTP / FTP 等。）。TCP/IP定义了电子设备如何连入因特网，以及数据如何在它们之间传输的标准。协议采用了4层的层级结构，每一层都呼叫它的下一层所提供的协议来完成自己的需求。通俗而言：TCP负责发现传输的问题，一有问题就发出信号，要求重新传输，直到所有数据安全正确地传输到目的地。而IP是给因特网的每一台联网设备规定一个地址。

2、TCP/IP协议的发展历程

（以下内容来自百度百科）：

为了减少网络设计的复杂性，大多数网络都采用分层结构。而TCP/IP协议采用了4层的层级结构。

在阿帕网（ARPA）产生运作之初，通过接口信号处理机实现互联的电脑并不多，大部分电脑相互之间不兼容。在一台电脑上完成的工作，很难拿到另一台电脑上去用，想让硬件和软

件都不一样的电脑联网，也有很多困难。当时美国的状况是，陆军用的电脑是DEC系列产品，海军用的电脑是Honeywell中标机器，空军用的是IBM公司中标的电脑，每一个军种的电脑在各自的系里都运行良好，但却有一个大弊病：不能共享资源。

当时科学家们提出这样一个理念："所有电脑生来都是平等的。"为了让这些"生来平等"的电脑能够实现"资源共享"就得在这些系统的标准之上，建立一种大家共同都必须遵守的标准，这样才能让不同的电脑按照一定的规则进行"谈判"，并且在谈判之后能"握手"。

在确定今天因特网各个电脑之间"谈判规则"过程中，最重要的人物当数瑟夫（Vinton G.Cerf）。正是他的努力，才使今天各种不同的电脑能按照协议上网互联。瑟夫也因此获得了与克莱因罗克（"因特网之父"）一样的美称"互联网之父"。

瑟夫从小喜欢标新立异，坚强而又热情。中学读书时，就被允许使用加州大学洛杉矶分校的电脑，他认为"为电脑编程序是个非常激动人心的事，.....只要把程序编好，就可以让电脑做任何事情。"1965年，瑟夫从斯坦福大学毕业到IBM的一家公司当系统工程师，工作没多久，瑟夫就觉得知识不够用，于是到加州大学洛杉矶分校攻读博士，那时，正逢阿帕网的建立，"接口信号处理机"（IMP）的研试及网络测评中心的建立，瑟夫也成了著名科学家克莱因罗克手下的一位学生。瑟夫与另外三位年轻人（温菲尔德、克罗克、布雷登）参与了阿帕网的第一个节点的联接。此后不久，BBN公司对工作中各种情况发展有很强判断能力、被公认阿帕网建成作出巨大贡献的鲍伯·卡恩（Bob Kahn）也来到了加州大学洛杉矶分校。在那段日子里，往往是卡恩提出需要什么软件，而瑟夫则通宵达旦地把符合要求的软件给编出来，然后他们一起测试这些软件，直至能正常运行。

当时的主要格局是这样的，罗伯茨提出网络思想设计网络布局，卡恩设计阿帕网总体结构，克莱因罗克负责网络测评系统，还有众多的科学家、研究生参与研究、试验。69年9月阿帕网诞生、运行后，才发现各个IMP连接的时候，需要考虑用各种电脑都认可的信号来打开通信管道，数据通过后还要关闭通道。否则这些**IMP**不会知道什么时候应该接收信号，什么时候该结束，这就是我们所说的通信"协议"的概念。1970年12月制定出来了最初的通信协议由卡恩开发、瑟夫参与的"网络控制协议"（**NCP**），但要真正建立一个共同的标准很不容易，72年10月国际电脑通信大会结束后，科学家们都在为此而努力。

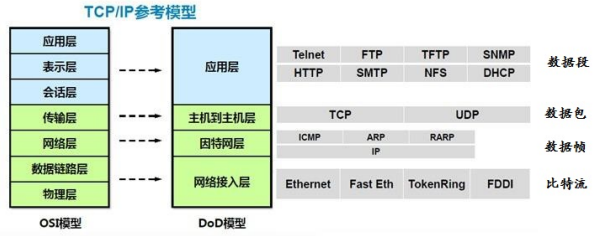
"包切换"理论为网络之间的联接方式提供了理论基础。卡恩在自己研究的基础上，认识到只有深入理解各种操作系统的细节才能建立一种对各种操作系统普适的协议，73年卡恩请瑟夫一起考虑这个协议的各个细节，他们这次合作的结果产生了在开放系统下的所有网民和网管人员都在使用的"传输控制协议"（TCP，Transmission-Control Protocol）和"因特网协议"（IP，Internet Protocol）即**TCP/IP**协议。

通俗而言：TCP负责发现传输的问题，一有问题就发出信号，要求重新传输，直到所有数据安全正确地传输到目的地。而IP是给因特网的每一台电脑规定一个地址。1974年12月，卡恩、瑟夫的第一份TCP协议详细说明正式发表。当时美国国防部与三个科学家小组签定了完成TCP/IP的协议，结果由瑟夫领衔的小组捷足先登，首先制定出了通过详细定义的TCP/IP协议标准。当时作了一个试验，将信息包通过点对点的卫星网络，再通过陆地电缆，再通过卫星网络，再由地面传输，贯串欧洲和美国，经过各种电脑系统，全程9.4万公里竟然没有丢失一个数据位，远距离的可靠数据传输证明了TCP/IP协议的成功。

1983年1月1日，运行较长时期曾被人们习惯了的NCP被停止使用，TCP/IP协议作为因特网上所有主机间的共同协议，从此以后被作为一种必须遵守的规则被肯定和应用。2005年9月

9日卡恩和瑟夫由于他们对于美国文化做出的卓越贡献被授予总统自由勋章。

3、TCP/IP四层模型



四、IP协议

1、介绍

IP协议（Internet Protocol）是将多个包交换网络连接起来，它在源地址和目的地址之间传送一种称之为数据包的东西，它还提供对数据大小的重新组装功能，以适应不同网络对包大小的要求。IP协议在OSI参考模型中应用于网络层，以“数据包（Package）”为单位。其中，IP地址的定义是确认唯一端口号和路由选择的关键，IP地址相当于每台电话的电话号码，唯一且是我们互相联系的关键，因此IP协议也是网络互连的关键。

2、IP协议特点

- ㈠IP协议是一种无连接、不可靠的分组传送服务的协议。
- ㈡IP协议是点-点线路的网络层通信协议。IP协议是针对原主机-路由器、路由器-路由器、路由器-目的主机之间的数据传输的点-点线路的网络层通信协议。
- ㈢IP协议屏蔽了网络在数据链路层、物理层协议与实现技术上的差异。：通过IP协议，网络层向传输层提供的是统一的**IP**分组，传输层不需要考虑互联网在数据链路层、物理层协议与实现技术上的差异，IP协议使得异构网络的互联变得容易了。

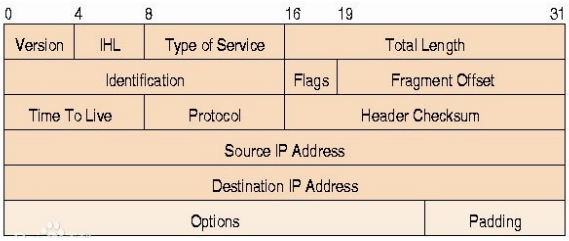
3、IPV4和IPV6

目前，主流IP是基于IPv4的，但IPV4网络难以实现网络实名制，一个重要原因就是因为IP资源的共用，因为IP资源不够（IPV4为32位），所以不同的人在不同的时间段共用一个IP，IP和上网用户无法实现一一对应。而IPv6（128位，足够长）的普及将改变现状，因为IPv6一个重要的应用将是实现网络实名制下的互联网身份证/VIeID，在IPv4下，现在根据IP查人也比较麻烦，电信局要保留一段时间的上网日志才行，通常因为数据量很大，运营商只保留三个月左右的上网日志，比如查前年某个IP发帖子的用户就不能实现。IPv6的出现可以从技术上一劳永逸地解决实名制这个问题，因为那时IP资源将不再紧张，运营商有足够多的IP资源，那时候，运营商在受理入网申请的时候，可以直接给该用户分配一个固定IP地址，这样实际就实现了实

名制，也就是一个真实用户和一个IP地址的一一对应。当一个上网用户的IP固定了之后，你任何时间做的任何事情都和一个唯一IP绑定，你在网络上做的任何事情在任何时间段内都有据可查，并且无法否认。因此你可能昨晚刚浏览过非法网站后，第二天早上就会有人上门给你开罚款单（啧啧）。

IPv4，是互联网协议（Internet Protocol，IP）的第四版，也是第一个被广泛使用，构成现今互联网技术的基础的协议。1981年 Jon Postel 在RFC791中定义了IP，Ipv4可以运行在各种各样的底层网络上，比如端对端的串行数据链路(PPP协议和SLIP协议），卫星链路等等。局域网中最常用的是以太网。目前的全球因特网所采用的协议族是TCP/IP协议族。IP是TCP/IP协议族中网络层的协议，是TCP/IP协议族的核心协议之一。目前IP协议的版本号是4(简称为IPv4，v，version版本)，它的下一个版本就是IPv6。

数据报格式如下：首部的长度是以**4**个字节为单位，长度可以是**20-60**字节，这跟首部的HLEN字段有关。



- 首部长度：这个4位字段定义了数据报首部的长度，以**4**字节的字为单位。当首部没有选项时，首部长度位**20**字节；当这个字段值位最大值F时，首部长度最大为**60**字节。

- 服务类型：在最初这个字段有一部分用于定义数据报的优先级，剩下的一部分定义了服务类型。IETF已经改变了这个8位字段的解释，现在定义了一组区分服务。在这种解释种，前6位构成了码点（codepoint），最后两位未使用。当码点字段最右边的3位不全为0时，这6位定义了54种服务，低延时，高吞吐量等等。

- 总长度：这个16位字段定义了数据报总长度，其以字节为单位。故IPv4数据报总长度上限值位65536字节。注：为什么需要这个字段？在许多情况下，我们确实不需要这个字段值。但是有些情况下，封装在一个帧里的并不仅仅是数据报，还可能附加了一些填充。比如，以太网协议对帧的数据有最大值（1500字节）和最小值（46字节）的限制，当数据小于46字节时，数据将含有填充数据。

- 标识（**identification**）：　这个16位字段标志了从源主机发出的一个数据报，这样就确定了数据报的唯一性。这样使得数据报被分片后，在到达终点时终点能根据标识号将同一个数据报的分片重新组装成一个数据报。

- 标志（**flag**）：第一位保留（未用），第二位为“不分片（do not fragment）”，第三位位“还有分片（more fragment）”。

- 分片偏移：这个13位字段表示的是分片在整个数据报中的相对位置。这是数据在原始数据报中的偏移量，以8字节位单位。

- 生存时间：这个8位字段用来控制数据报所经过的最大跳数（路由器），每经过一个路由器，这个字段数值都减1，减1后变位0时，路由器就丢弃这个数据报。
- 协议：这个8位字段定义了使用IPv4服务的高层协议，如TCP，UDP，ICMP，IGMP，OSPF等的数据都将被封装到IP数据报中。这个字段指明数据报必须交付给哪个最终目的协议。

- 检验和：检验IP数据报首部。
- 源地址：定义了源点的IP地址，这个字段始终保持不变。
- 目的地址：定义了终点的IP地址，这个字段始终保持不变。

IPV4地址格式：

IPv4中规定IP地址长度为**32**（按TCP/IP参考模型划分），即有2^32-1个地址。 ipv4所存在的问题 一般的书写法为4个用小数点分开的十进制数。也有人把4位数字化成一个十进制长整数，但这种标示法并不常见。另一方面，IPv6使用的128位地址所采用的位址记数法，在IPv4也有人用，但使用范围更少。 过去IANAIP地址分为A,B,C,D 4类，把32位的地址分为两个部分：前面的部分代表网络地址，由IANA分配，后面部分代表局域网地址。如在C类网络中，前24位为网络地址，后8位为局域网地址，可提供254个设备地址(因为有两个地址不能为网络设备使用: 255为广播地址，0代表此网络本身) 。网络掩码(Netmask) 限制了网络的范围，1代表网络部分，0代表设备地址部分，例如C类地址常用的网络掩码为255.255.255.0。

IPV6：

由于IPv4最大的问题在于网络地址资源有限，严重制约了互联网的应用和发展。IPv6的使用，不仅能解决网络地址资源数量的问题，而且也解决了多种接入设备连入互联网的障碍。IPV6号称可以为全世界的每一粒沙子编上一个网址。

发展历史：2003年1月22日，IETF发布了IPv6测试性网络。最初开始于虚拟网络，它使用IPv6-over-IPv4隧道过渡技术。因此，它是一个基于IPv4互联网且支持IPv6传输的网络，后来逐渐建立了纯IPv6链接。从2011年开始，主要用在个人计算机和服务器系统上的操作系统基本上都支持高质量IPv6配置产品。例如，Microsoft Windows从Windows 2000起就开始支持IPv6，到Windows XP时已经进入了产品完备阶段。一些应用基于IPv6实现．如BitTorrent点到点文件传输协议等，避免了使用NAT的IPv4私有网络无法正常使用的普遍问题。2012年6月

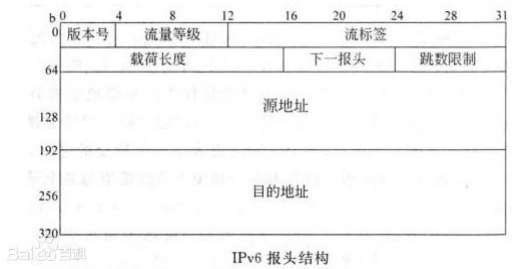
6日，国际互联网协会举行了世界IPv6启动纪念日，这一天，全球IPv6网络正式启动。多家知名网站，如Google、Facebook和Yahoo等，于当天全球标准时间0点（北京时间8点整）开始永久性支持IPv6访问。

IPV6地址格式：

IPv6的地址长度为128b，是IPv4地址长度的4倍。于是IPv4点分十进制格式不再适用，采用十六进制表示。IPv6有3种表示方法。

- 冒分十六进制表示法：
  格式为X:X:X:X:X:X:X:X，其中每个X表示地址中的16b，以十六进制表示，例如：
  ABCD:EF01:2345:6789:ABCD:EF01:2345:6789，这种表示法中，每个X的前导0是可以省略的，例如：2001:0DB8:0000:0023:0008:0800:200C:417A→
  2001:DB8:0:23:8:800:200C:417A。
- **0**位压缩表示法**:**
  在某些情况下，一个IPv6地址中间可能包含很长的一段0，可以把连续的一段0压缩为“::”。但为保证地址解析的唯一性，地址中”::”只能出现一次，例如：
  FF01:0:0:0:0:0:0:1101 → FF01::1101
  0:0:0:0:0:0:0:1 → ::1
  0:0:0:0:0:0:0:0 → ::
- 内嵌**IPv4**地址表示法**:**
  为了实现IPv4-IPv6互通，IPv4地址会嵌入IPv6地址中，此时地址常表示为：
  X:X:X:X:X:X:d.d.d.d，前96b采用冒分十六进制表示，而最后32b地址则使用IPv4的点分十进制表示，例如::192.168.0.1与::FFFF:192.168.0.1就是两个典型的例子，注意在前96b中，压缩0位的方法依旧适用。

IPV6报文格式：



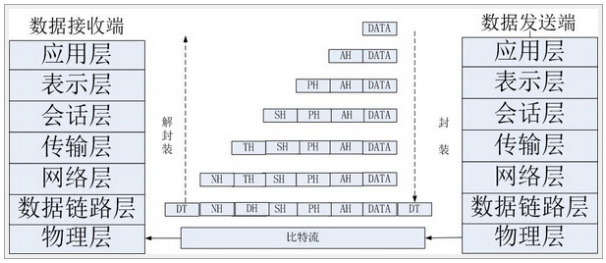
IPv6 报头结构

版本号	表示协议版本，值为6
流量等级	主要用于QoS
流标签	用来标识同一个流里面的报文
载荷长度	表明该IPv6包头部后包含的字节数，包含扩展头部
下一报头	该字段用来指明报头后接的报文头部的类型，若存在扩展头，表示第一个扩展头的类型，否则表示其上层协议的类型，它是IPv6各种功能的核心实现方法
跳数限制	该字段类似于IPv4中的TTL，每次转发跳数减一，该字段达到0时包将会被丢弃
源地址	标识该报文的来源地址
目的地址	标识该报文的目的地址

五、TCP协议

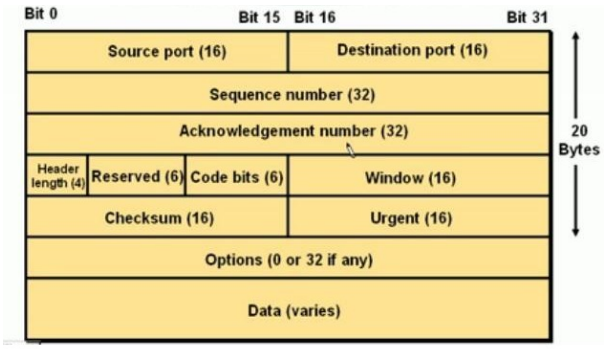
1、介绍

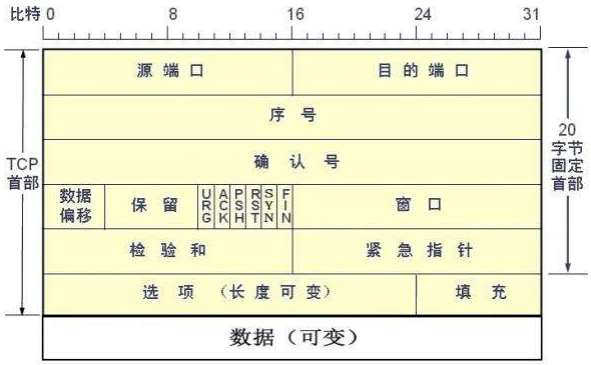
TCP(Transmission Control Protocol 传输控制协议)是一种面向连接的、可靠的，基于IP的传输层协议。TCP在IP报文的协议号是6。TCP是一个超级麻烦的协议，而它又是互联网的基础。下面是OSI七层模型图：



TCP工作在网络OSI的七层模型中的第四层——Transport层（传输层），IP在第三层——Network层，ARP 在第二层——Data Link层。在第二层的数据，我们把它叫Frame（数据帧），在第三层的数据叫Packet（数据包），第四层的数据叫Segment（数据段）。 同时，我们需要简单的知道，数据从应用层发下来，会在每一层都会加上头部信息，进行封装，然后再发送到数据接收端。所以数据的发送和接收其实就是数据的封装和解封装的过程。

2、TCP报文格式



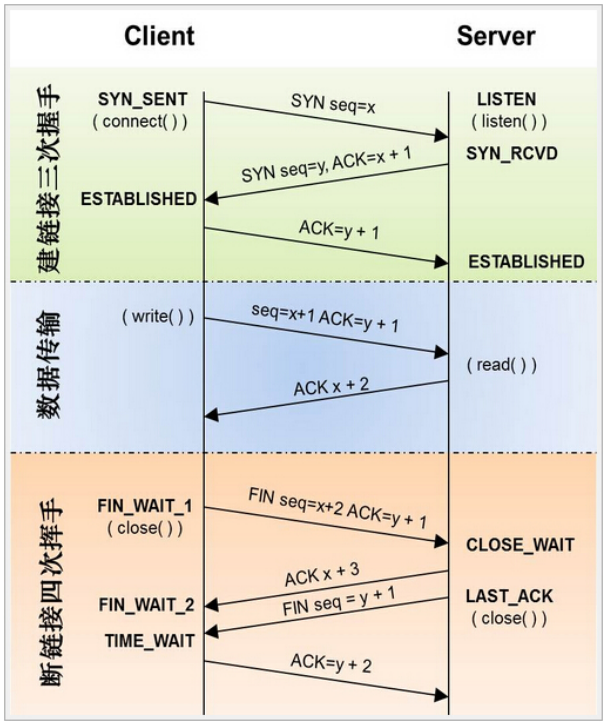


如上图所示，TCP数据格式是由若干具有特殊含义字段组成的。其中，

- **Source Port**和**Destination Port**：分别占用16位，表示源端口号和目的端口号;用于区别主机中的不同进程， 而IP地址是用来区分不同的主机的，源端口号和目的端口号配合上IP首部中的源IP地址和目的IP地址就能唯一 的确定一个TCP连接；
- **Sequence Number**：用来标识从TCP发端向TCP收端发送的数据字节流，它表示在这个报文段中的的第一个数据 字节在数据流中的序号;主要用来解决网络报乱序的问题；
- **Acknowledgment Number**：32位确认序列号包含发送确认的一端所期望收到的下一个序号，因此，确认序号应 当是上次已成功收到数据字节序号加1。不过，只有当标志位中的ACK标志(下面介绍)为1时该确认序列号的字 段才有效。主要用来解决不丢包的问题；
- **Offset**：给出首部中32 bit字的数目，需要这个值是因为任选字段的长度是可变的。这个字段占4bit(最多能 表示15个32bit的的字，即4*15=60个字节的首部长度)，因此TCP最多有60字节的首部。然而，没有任选字段， 正常的长度是20字节；
- **TCP Flags:**TCP首部中有6个标志比特，它们中的多个可同时被设置为1，主要是用于操控TCP的状态机的，依次 为URG，ACK，PSH，RST，SYN，FIN。
  URG：此标志表示TCP包的紧急指针域(后面马上就要说到)有效，用来保证TCP连接不被中断，并且督促 中间层设备要尽快处理这些数据；
  ACK：此标志表示应答域有效，就是说前面所说的TCP应答号将会包含在TCP数据包中;有两个取值：0和1， 为1的时候表示应答域有效，反之为0；
  PSH：这个标志位表示Push操作。所谓Push操作就是指在数据包到达接收端以后，立即传送给应用程序， 而不是在缓冲区中排队；
  RST：这个标志表示连接复位请求。用来复位那些产生错误的连接，也被用来拒绝错误和非法的数据包；
  SYN：表示同步序号，用来建立连接。SYN标志位和ACK标志位搭配使用，当连接请求的时候，SYN=1， ACK=0；连接被响应的时候，SYN=1，ACK=1；这个标志的数据包经常被用来进行端口扫描。扫描者发送 一个只有SYN的数据包，如果对方主机响应了一个数据包回来 ，就表明这台主机存在这个端口;但是由于这 种扫描方式只是进行TCP三次握手的第一次握手，因此这种扫描的成功表示被扫描的机器不很安全，一台安全 的主机将会强制要求一个连接严格的进行TCP的三次握手；
- **Window**：窗口大小，也就是有名的滑动窗口，用来进行流量控制。这是一个复杂的问题，本文不再论述。

2、TCP协议的三次握手

TCP是面向连接的，无论哪一方向另一方发送数据之前，都必须先在双方之间建立一条连接。在TCP/IP协议中，TCP 协议提供可靠的连接服务，连接是通过三次握手进行初始化的。三次握手的目的是同步连接双方的序列号和确认号并交换 TCP窗口大小信息。如下图TCP的通信过程所示：



三次握手具体过程（状态）如下（其实可以类比打电话的过程：甲打电话，并等待接听→乙收到来电显示，“并表示可以接听”→“甲收到乙可以接听的信息”，甲接听电话。注：引号部分是打电话过程中没有的，但在TCP三次握手中存在）：

1. 第一次握手：建立连接。客户端发送连接请求报文段，将SYN位置为1，Sequence Number为x;然后，客户端进入SYN_SEND状态，等待服务器的确认。（客户的建立连接并等待确认）

2. 第二次握手：服务器收到SYN报文段。服务器收到客户端的SYN报文段，需要对这个SYN报文段进行确认，设置Acknowledgment Number为x+1(Sequence Number+1);同时，自己自己还要发送SYN请求信息，将SYN位置为1，Sequence Number为y;服务器端将上述所有信息放到一个报文段(即SYN+ACK报文段)中，一并发送给客户端，此时服务器进入SYN_RECV状态。（服务器端发送相关报文段信息并等待连接）

3. 第三次握手：客户端收到服务器的SYN+ACK报文段。然后将Acknowledgment Number设置为y+1，向服务器发送ACK报文段，这个报文段发送完毕以后，客户端和服务器端都进入ESTABLISHED状态，完成TCP三次握手。（客户的接收到服务端信息并实现连接）

然后，客户端和服务端就能实现正常的数据传输啦！

3、TCP协议的四次分手

既然握手都需要频繁确认，那么“分手”又怎能马虎呢？具体过程（状态）如下（同样也可以看做挂电话的过程：我说完了，挂？→我也说完了，挂吧？→好，拜拜→bye。简言之就是确认通信双方都交流完毕再确认断开连接）：

1. 第一次分手：主机1(可以使客户端，也可以是服务器端)，设置Sequence Number和Acknowledgment Number，向主机2发送一个FIN报文段;此时，主机1进入FIN_WAIT_1状态;这表示主机1没有数据要发送给主机2了。（一方数据发送完成）
2. 第二次分手：主机2收到了主机1发送的FIN报文段，向主机1回一个ACK报文段，Acknowledgment Number为Sequence Number加1;主机1进入FIN_WAIT_2状态;主机2告诉主机1，我也没有数据要发送了，可以进行关闭连接了。（另一方数据发送完成）
3. 第三次分手：主机2向主机1发送FIN报文段，请求关闭连接，同时主机2进入CLOSE_WAIT状态。（请求关闭连接并等待）
4. 第四次分手：主机1收到主机2发送的FIN报文段，向主机2发送ACK报文段，然后主机1进入TIME_WAIT状态;主机2收到主机1的ACK报文段以后，就关闭连接;此时，主机1等待2MSL后依然没有收到回复，则证明Server端已正常关闭，那好，主机1也可以关闭连接了。（关闭连接）

现在，我们也应该理解为什么TCP协议是面向连接的、可靠的、基于IP协议的“通信控制协议”了。TCP的三次握手保证了数据的可靠性，保证资源不被浪费，而四次分手保证连接的可靠性而不至于随意断开连接，但TCP协议也由其可靠性，数据传输效率变得较低，而不像UDP那样进行实时快速传输，下面我们就来学习什么是UDP协议。

六、**UDP**协议

1、介绍

UDP 是User Datagram Protocol的简称， 中文名是用户数据报协议，是OSI（Open System Interconnection，开放式系统互联） 参考模型中一种无连接的传输层协议，提供面向事务的简单不可靠信息传送服务，IETF RFC 768是UDP的正式规范。UDP在IP报文的协议号是17。

与所熟知的TCP（传输控制协议）协议一样，UDP协议直接位于IP（网际协议）协议的顶层。根据OSI（开放系统互连）参考模型，UDP和TCP都属于传输层协议。UDP协议的主要作用是将网络数据流量压缩成数据包的形式。一个典型的数据包就是一个二进制数据的传输单位。每一个数据包的前8个字节（16*4位）用来包含报头信息，剩余字节则用来包含具体的传输数据。

2、UDP报文格式

与TCP协议不同，UDP协议是非面向连接的不可靠协议，因此没有了SYN等处理两端等待或连接的报文段，相比之下，UDP的报文格式更为简单，主要由报文头（由均16位的源端口号、目的端口号、UDP长度和UDP校验和组成）和具体传输数据组成。如图所示：

16 位源端口号	16 位目的端口号
16 位 UDP 长度	16 位 UDP 检验和
数据（如果有）	

- **UDP**长度：UDP报文的整个大小，最小为8个字节（16*4位）（仅为首部）。
- **UDP**检验和：在进行检验和计算时，会添加一个伪首部一起进行运算。伪首部（占用12个字节）为：4个字节的源IP地址、4个字节的目的IP地址、1个字节的0、一个字节的数字17、以及占用2个字节UDP长度。这个伪首部不是报文的真正首部，只是引入为了计算校验和。相对于IP协议的只计算首部，UDP检验和会把首部和数据一起进行校验。接收端进行的校验和与UDP报文中的校验和相与，如果无差错应该全为1。如果有误，则将报文丢弃或者发给应用层、并附上差错警告。

3、UDP特性

- （1） UDP是一个无连接协议，传输数据之前源端和终端不建立连接，当 UDP想传送时就简单地去抓取来自应用程序的数据，并尽可能快地把它扔到网络上。在发送端，UDP传送数据的速度仅仅是受应用程序生成数据的速度、计算机的能力和传输带宽的限制；在接收端，UDP把每个消息段放在队列中，应用程序每次从队列中读一个消息段。
- （2） 由于传输数据不建立连接，因此也就不需要维护连接状态，包括收发状态等，因此一台服务机可同时向多个客户机传输相同的消息。
- （3） UDP信息包的标题很短，只有8个字节，相对于TCP的20个字节信息包的额外开销很小。
- （4） 吞吐量不受拥挤控制算法的调节，只受应用软件生成数据的速率、传输带宽、源端和终端主机性能的限制。
- （5）UDP使用尽最大努力交付，即不保证可靠交付，因此主机不需要维持复杂的链接状态表（这里面有许多参数）。
- （6）**UDP**是面向报文的。发送方的UDP对应用程序交下来的报文，在添加首部后就向下交付给IP层。既不拆分，也不合并，而是保留这些报文的边界，因此，应用程序需要选择合适的报文大小。 虽然UDP是一个不可靠的协议，但它是分发信息的一个理想协议。例如，在屏幕上报告股票市场、在屏幕上显示航空信息等等。UDP也用在路由信息协议RIP（Routing Information Protocol）中修改路由表。在这些应用场合下，如果有一个消息丢失，在几秒之后另一个新的消息就会替换它。UDP广泛用在多媒体应用中，例如，Progressive Networks公司开发的RealAudio软件，它是在因特网上把预先录制的或者现场音乐实时传送给客户机的一种软件，该软件使用的RealAudio audio-on-demand

protocol协议就是运行在UDP之上的协议，大多数因特网电话软件产品也都运行在UDP之上。

在选择使用协议的时候，选择**UDP**必须要谨慎。在网络质量令人十分不满意的环境下，UDP协议数据包丢失会比较严重。但是由于UDP的特性：它不属于连接型协议，因而具有资源消耗小，处理速度快的优点，所以通常音频、视频和普通数据在传送时使用**UDP**较多，因为它们即使偶尔丢失一两个数据包，也不会对接收结果产生太大影响。比如我们聊天用的ICQ和QQ就是使用的UDP协议。

七、**TCP**协议和**UDP**协议的区别

1、一般区别

- TCP是面向连接的，传输数据保证可靠性和安全性；TCP协议是非面向连接的，是不可靠但高效率的协议。
- TCP占用资源多而UDP占用少。
- TCP是流模式而TCP是数据报模式。（可以这样理解：TCP是面向连接的，用打电话的过程来类比，就是通信双方是互相明确的，所以进行的是“你一句我一句”的交流，TCP整个通信过程间有一个缓存区，由于通信主体明确，因此可以断断续续地进行交流，数据好比水流，知道源头和目的地，因此称为流模式。反过来，UDP是非面向连接的，好比写信的过程，假设我们只要知道佩奇的地址，我们就能写信给佩奇，而佩奇却不认识我们。这样发起通信方的身份是不明确的，每个发送端的信息都不能和别的发送端混淆，不然会造成数据失效，所以UDP要对数据进行“打包”发送，是面向报文的，就像写信需要用信封套起来，不然只发送数据甚至数据混合会变得毫无意义，就像qq群的匿名聊天，这不扯皮吗？）
- TCP和UDP的应用场景和编程方式也有很大差别，此处不再赘述。

2、TCP的粘包和**UDP**的丢包

TCP粘包现象：**TCP**粘包是指发送方发送的若干包数据到接收方接收时粘成一包，从接收缓冲区看，后一包数据的头紧接着前一包数据的尾。

粘包原因：

- 发送端：TCP默认会使用Nagle算法。而**Nagle**算法主要做两件事：1）只有上一个分组得到确认，才会发送下一个分组；2）收集多个小分组，在一个确认到来时一起发送。所以，正是Nagle算法造成了发送方有可能造成粘包现象。
- 接收端：TCP接收到分组时，并不会立刻送至应用层处理，或者说，应用层并不一定会立即处理；实际上，TCP将收到的分组保存至接收缓存里，然后应用程序主动从缓存里读收到的分组。这样一来，如果TCP接收分组的速度大于应用程序读分组的速度，多个包就会被存至缓存，应用程序读时，就会读到多个首尾相接粘到一起的包。

粘包处理：如果黏在一起的包是同一个整体，即同意部分数据分割而来的，那么就不用进行处理。如果是不同部分的数据粘到一起，就需要进行粘包解决：

- 发送端导致：使用TCP_NODELAY选项来关闭Nagle算法。
- 接收端导致：暂无。
- 统一解决（应用层）：可以解决接收方造成的粘包问题，还能解决发送方造成的粘包问题。
  解决方法就是循环处理：应用程序在处理从缓存读来的分组时，读完一条数据时，就应该循环读下一条数据，直到所有的数据都被处理；但是如何判断每条数据的长度呢？
  两种途径：
  1）格式化数据：每条数据有固定的格式（开始符、结束符），这种方法简单易行，但选择开始符和结束符的时候一定要注意每条数据的内部一定不能出现开始符或结束符；
  2）发送长度（推荐）：发送每条数据的时候，将数据的长度一并发送，比如可以选择每条数据的前4位是数据的长度，应用层处理时可以根据长度来判断每条数据的开始和结束。

UDP丢包现象：丢包现象即使用UDP发送时，由于不可靠连接方式，收到各种因素影响，数据包可能会在接受过程中丢失一部分，从而导致数据不完整。

UDP丢包原因：

- 发送端：发送的包太大导致send方法无法正常切割为小包导致丢包、发送的包太大超过缓存设置也会出现对包、发送频率太快导致接收端未接受或溢出缓冲区而丢包。
- 接收端：处理时间过长导致丢包。
- 其他：网络等问题。

UDP丢包处理：

- UDP的缺陷在于丢包和乱序问题，一般视情况进行处理，而发送的时候也需要注意上述导致丢包的问题。

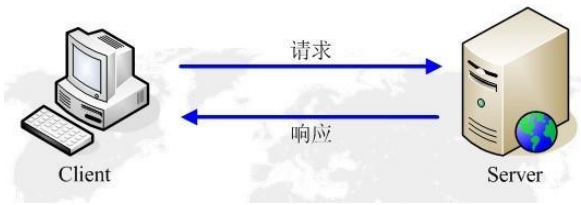
八、**HTTP**协议

1、介绍

超文本传输协议（HTTP，HyperText Transfer Protocol)是互联网上应用最为广泛的一种网络协议。所有的万维网WWW（World Wide Web）文件都必须遵守这个标准。设计HTTP最初的目的是为了提供一种发布和接收**HTML**页面的方法。1960年美国人Ted Nelson构思了一种通过计算机处理文本信息的方法，并称之为超文本（hypertext）,这成为了HTTP超文本传输协议标准架构的发展根基。

HTTP是一个基于TCP/IP通信协议来传递数据（HTML 文件, 图片文件, 查询结果等）。HTTP是一个属于应用层的面向对象的协议，由于其简捷、快速的方式，适用于分布式超媒体信息系统。它于1990年提出，经过几年的使用与发展，得到不断地完善和扩展。目前在WWW中使用的是HTTP/1.0的第六版，HTTP/1.1的规范化工作正在进行之中，而且HTTP-NG(Next Generation of HTTP)的建议已经提出。

HTTP协议工作于客户端-服务端架构为上。浏览器作为HTTP客户端通过URL向HTTP服务端即WEB服务器发送所有请求。Web服务器根据接收到的请求后，向客户端发送响应信息。



2、HTTP特点

HTTP是一个客户端和服务器端请求和应答的标准，通常，由HTTP客户端发起一个请求，建立一个到服务器指定端口（默认是80端口）的TCP连接。HTTP服务器则在那个端口监听客户端发送过来的请求。一旦收到请求，服务器（向客户端）发回一个状态行。 HTTP协议的网页 HTTP协议的网页 **HTTP**使用**TCP**而不是**UDP**的原因在于（打开）一个网页必须传送很多数据，而TCP协议提供传输控制，按顺序组织数据，和错误纠正。

通过HTTP或者HTTPS协议（HTTP协议+SSL协议）请求的资源由统一资源标示符（Uniform Resource Identifiers）（或者，更准确一些，URLs）来标识。HTTP有以下特点：

- 简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。

- 灵活：HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type加以标记。
- 无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。
- 无状态：HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。
- 支持B/S及C/S模式。

3、HTTP的**URL**地址

URL是一种特殊类型的URI，包含了用于查找某个资源的足够的信息，URL,全称是UniformResourceLocator, 中文叫统一资源定位符,是互联网上用来标识某一处资源的地址。以下面这个URL为例，介绍下普通URL的各部分组成：

http://www.cnblogs.com：8080/fzz9/index.jsp？id=30303&page=2#name

- 协议部分：一般为HTTP或Https，后接//作为分隔符。
- 域名部分：www.cnblogs.com为网站域名。
- 端口号部分：此网址为8080。跟在域名后面的是端口号，域名和端口之间使用“:”作为分隔符。端口不是一个URL必须的部分，如果省略端口部分，将采用默认端口。
- 虚拟目录部分：从域名后的第一个“/”开始到最后一个“/”为止，是虚拟目录部分。虚拟目录也不是一个URL必须的部分。
- 参数部分：从“？”开始到“#”为止之间的部分为参数部分。本例中的参数部分为“id=30303&page=2”。不是必要部分。
- 文件名部分：从域名后的最后一个“/”开始到后面一个“？”为止，是文件名部分，如果没有“?”,则是从域名后的最后一个“/”开始到“#”为止，是文件部分，如果没有“？”和“#”，那么从域名后的最后一个“/”开始到结束，都是文件名部分。本例中的文件名是“index.jsp”。文件名部分也不是一个URL必须的部分，如果省略该部分，则使用默认的文件名。
- 锚部分：从“#”开始到最后，都是锚部分。本例中的锚部分是“name”。锚部分也不是一个URL必须的部分。

4、HTTP请求之request

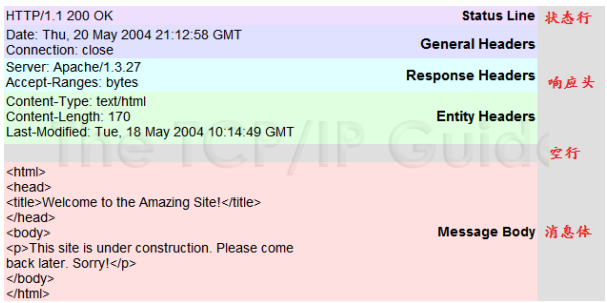
客户端通过HTTP协议进行请求时遵循一定的格式，请看下面的请求报文格式（由请求行、请求头、空行、请求体组成）：

而各部分组成如下所示：

而请求主要分为post提交方法和get提交方法，每种选择各有优缺点，此处不再赘述。目前大多数网站多采用post提交。

5、Http响应之response

在客户端发送请求后服务端进行响应，将信息发送给客户端，以实现功能服务，报文格式如下（包含状态行、响应头、空行、消息体）：



响应组成此处也不再赘述，值得注意的是状态码，它以清晰明确的数字告诉客户端本次请求的处理结果。 常见的状态码有：

2xx:成功

消息	描述
200 OK	请求成功（其后是对GET和POST请求的应答文档。）
201 Created	请求被创建完成，同时新的资源被创建。
202 Accepted	供处理的请求已被接受，但是处理未完成。
203 Non-authoritative Information	文档已经正常地返回，但一些应答头可能不正确，因为使用的是文档的拷贝。
204 No Content	没有新文档。浏览器应该继续显示原来的文档。如果用户定期地刷新页面，而Servlet可以确定用户文档足够新，这个状态代码是很有用的。
205 Reset Content	没有新文档。但浏览器应该重置它所显示的内容。用来强制浏览器清除表单输入内容。
206 Partial Content	客户发送了一个带有Range头的GET请求，服务器完成了它。

3xx:重定向

消息	描述
300 Multiple Choices	多重选择。链接列表。用户可以选择某链接到达目的地。最多允许五个地址。
301 Moved Permanently	所请求的页面已经转移至新的url。
302 Found	所请求的页面已经临时转移至新的url。
303 See Other	所请求的页面可在别的url下被找到。
304 Not Modified	未按预期修改文档。客户端有缓冲的文档并发出了一个条件性的请求（一般是提供If-Modified-Since头表示客户只想比指定日期更新的文档）。服务器告诉客户，原来缓冲的文档还可以继续使用。
305 Use Proxy	客户请求的文档应该通过Location头所指明的代理服务器提取。
306 *Unused*	此代码被用于前一版本。目前已不再使用，但是代码依然被保留。
307 Temporary Redirect	被请求的页面已经临时移至新的url。

4xx:客户端错误

消息	描述
400 Bad Request	服务器未能理解请求。
401 Unauthorized	被请求的页面需要用户名和密码。
401.1	登录失败。
401.2	服务器配置导致登录失败。
401.3	由于 ACL 对资源的限制而未获得授权。
401.4	筛选器授权失败。
401.5	ISAPI/CGI 应用程序授权失败。
401.7	访问被 Web 服务器上的 URL 授权策略拒绝。这个错误代码为 IIS 6.0 所专用。
402 Payment Required	此代码尚无法使用。
403 Forbidden	对被请求页面的访问被禁止。
403.1	执行访问被禁止。
403.2	读访问被禁止。
403.3	写访问被禁止。

5xx:服务器错误

消息	描述
500 Internal Server Error	请求未完成。服务器遇到不可预知的情况。
500.12	应用程序正忙于在 Web 服务器上重新启动。
500.13	Web 服务器太忙。
500.15	不允许直接请求 Global.asa。
500.16	UNC 授权凭据不正确。这个错误代码为 IIS 6.0 所专用。
500.18	URL 授权存储不能打开。这个错误代码为 IIS 6.0 所专用。
500.100	内部 ASP 错误。
501 Not Implemented	请求未完成。服务器不支持所请求的功能。
502 Bad Gateway	请求未完成。服务器从上游服务器收到一个无效的响应。
502.1	CGI 应用程序超时。
502.2	CGI 应用程序出错。
503 Service Unavailable	请求未完成。服务器临时过载或当机。
504 Gateway Timeout	网关超时。
505 HTTP Version Not Supported	服务器不支持请求中指明的HTTP协议版本。

九、**Socket**编程简述

Socket的英文原义是“孔”或“插座”。网络上的两个程序通过一个双向的通信连接实现数据的交换，这个连接的一端称为一个socket。

建立网络通信连接至少要一对端口号(socket)。socket本质是编程接口**(API)**，对**TCP/IP**的封装，TCP/IP也要提供可供程序员做网络开发所用的接口，这就是Socket编程接口。

HTTP是轿车，提供了封装或者显示数据的具体形式；Socket是发动机，提供了网络通信的能力。

网络层的**ip**地址可以唯一标识网络中的主机，而传输层的“协议**+**端口”可以唯一标识主机中的应用程序（进程）。这样利用三元组（**ip**地址，协议，端口）就可以标识网络的进程了，网络中的进程通信就可以利用这个标志与其它进程进行交互。 使用TCP/IP协议的应用程序通常采用应用编程接口，即**Socket**（UNIX BSD的套接字（socket）和UNIX System V的TLI，已经被淘汰），来实现网络进程之间的通信。

这里只是简单了解，有兴趣的可自行搜索，或参考这篇博文：https://blog.csdn.net/alpha_love/article/details/62043077。

十、碎碎念

经过本文的学习，我们已经大概了解了网络到底如何改变世界，网络协议又是多么重要。但网络协议的发展不止于此，且本文仅是浅显的总结，以后会继续补充完善，除此之外还有诸多内容需要我们深入学习，要想加深对网络协议的认知，还需要我们在运用中进行实践总结。