1.3 OSI模型

第一层:物理层(PhysicalLayer)

规定通信设备的机械的、电气的、功能的和规程的特性，用以建立、维护和拆除物理链路连接。具体地讲，机械特性定了网络连接时所需接插件的规格尺寸、引脚数量和排列情况等;电气特性规定了在物理连接上传输bit流时线路上信号电平的大小、阻抗匹配、传输速率距离限制等;功能特性是指对各个信号先分配确切的信号含义，即定义了DTE和DCE之间各个线路的功能;规程特性定义了利用信号线进行bit流传输的一组操作规程，是指在物理连接的建立、维护、交换信息时，DTE和DCE双方在各电路上的动作系列。

在这一层，数据的单位称为比特(bit)。物理层的主要设备:中继器、集线器、适配器。

第二层:数据链路层(DataLinkLayer)

物理层提供比特流服务的基础上，建立相邻结点之间的数据链路，通过差错控制提供数据帧(Frame)在信道上无差错的传输，并进行各电路上的动作系列。

数据链路层在不可靠的物理介质上提供可靠的传输。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)。

数据链路层主要设备:二层交换机、网桥。

第三层:网络层(Networklayer)

在计算机网络中进行通信的两个计算机之间可能会经过很多个数据链路，也可能还要经过很多通信子网。网络层的任务就是选择合适的网间路由和交换结点，确保数据及时传送。网络层将数据链路层提供的帧组成数据包，包中封装有网络层包头，其中含有逻辑地址信息- -源站点和目的站点地址的网络地址。

如果你在谈论一个IP地址，那么你是在处理第3层的问题，这是”数据包”问题，而不是第2层的”帧”。IP是第3层问题的一部分，此外还有一些路由协议和地址解析协议(ARP)。有关路由的一切事情都在第3层处理。地址解析和路由是3层的重要目的。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、ARP、RARP、OSPF等。

网络层主要设备:路由器

第四层:传输层(Transportlayer)

第4层的数据单元也称作处理信息的传输层(Transport layer)。但是，当你谈论TCP等具体的协议时又有特殊的叫法，TCP的数据单元称为段(segments)而UDP协议的数据单元称为”数据报(datagrams)”。这个层负责获取全部信息，因此，它必须跟踪数据单元碎片、乱序到达的数据包和其它在传输过程中可能发生的危险。第4层为上层提供端到端(最终用户到最终用户)的透明的、可靠的数据传输服务。所谓透明的传输是指在通信过程中传输层对上层屏蔽了通信传输系统的具体细节。

传输层协议的代表包括:TCP、UDP、SPX等。

第五层:会话层(Sessionlayer)

这一层也可以称为会晤层或对话层，在会话层及以上的高层次中，数据传送的单位不再另外命名，统称为报文。会话层不参与具体的传输，它提供包括访问验证和会话管理在内的建立和维护应用之间通信的机制。如服务器验证用户登录便是由会话层完成的。

第六层:表示层(Presentationlayer)

这一层主要解决用户信息的语法表示问题。它将欲交换的数据从适合于某一用户的抽象语法，转换为适合于OSI系统内部使用的传送语法。即提供格式化的表示和转换数据服务。数据的压缩和解压缩， 加密和解密等工作都由表示层负责。例如图像格式的显示，就是由位于表示层的协议来支持。

第七层:应用层(Applicationlayer)

应用层为操作系统或网络应用程序提供访问网络服务的接口。

应用层协议的代表包括:Telnet、FTP、HTTP、SNMP等。

2、TCP和UDP有什么区别

2.1 UDP （User Data Protocol，用户数据报协议）

首先UDP协议是无连接，不可靠，面向数据报的。

无连接：知道对端的IP和端⼝号就直接进⾏传输, 不需要建⽴连接;

不可靠： UDP没有确认机制来确保报文的到达，没有对报文进行排序的机制，也不提供反馈信息来控制端到端报文传输的速度，所以它的传输是不可靠的。

面向数据报：不能够灵活的控制读写数据的次数和数量。

UDP常用端口：

53：DNS域名服务

69：TFTP（简单文件传输协议）

123：NTP（网络时间协议）

161：SNMP（简单网络管理协议）

162：SNMP（简单网络管理协议：陷阱）

520：RIP（路由信息协议）

2.1 TCP（Transmission Control Protocol，传输控制协议）

TCP协议是面向连接的，可靠地，面向字节的。

连接管理机制：

正常情况下，TCP要经过三次握手建立连接，四次挥手断开连接。

确认应答（ACK）机制：

TCP主机A每次给主机B发送一次数据，主机B都要给主机A发送一个确认应答（每一个ACK都带有对应的确认序列号，告诉发送者我已经收到了那些数据，下一次该从哪里开始发）。

超时重传机制：

如果主机A发送给主机B的数据因为网络拥堵等原因丢包了，数据无法到达主机B。

在特定时间内，主机A没有接收到主机B发送回来的ACK，就会把数据进行重发。

滑动窗口：

因为加入了确认应答机制和超时重传机制，导致性能变差，为了提高性能，引入滑动窗口机制。

即两主机发送数据帧时，不是一帧帧的发送，而是每次发送指定窗口大小的数据帧（当然可以不满，窗口的大小取决于接收窗口和拥塞窗口中的较小值）。

TCP滑动窗口工作过程如下：

双方在建立连接时确定发送和接受的字节序号，确定最大段长度MSS的值为标准长度（确定发送和接受的窗口）。

发送方发送一个报文（其数据长度为MSS）后启动计时器，转到步骤4等待接收方应答。

接收方收到报文后给出应答，其中包含的窗口值即为可以接受的字节数，调整可接收的序列号（接受窗口）。

发送方等待接收方应答，如果收到对方的应答，则根据接收到的应答报文段中的窗口值更新自己的MSS值，转到步骤二。

发送方如果超时未接收到对方应答，则重传该报文段后启动计时器，转到步骤四。

流量控制：

接收端处理数据的速度是有限的. 如果发送端发的太快, 导致接收端的缓冲区被打满, 这个时候如果发送端继续发送, 就会造成丢包, 继⽽引起丢包重传等等⼀系列连锁反应。

因此TCP支持根据接收端的处理能力，来决定发送端的发送速度，这个机制就叫做流量控制。

拥塞控制：

虽然TCP有了滑动窗⼝机制, 能够⾼效可靠的发送⼤量的数据. 但是如果在刚开始阶段就发送⼤量的数据, 仍然可能引发问题。

因为网络上有很多的计算机, 可能当前的网络状态就已经比较拥堵。在不清楚当前网络状态下，贸然发送大量的数据，很有可能造成严重的问题。

TCP引入慢启动机制，先发少量的数据，摸清当前的网络拥堵状态，再决定按照多大的速度传输数据。

拥塞窗口的增长速度，是指数级别的。“慢启动”只是初始时慢，但是增长速度非常快。为了不增长的那么快，因此不能使拥塞窗口单纯的加倍，此处引用一个叫做慢启动的阈值，当拥塞窗口超过这个阈值的时候，不再按照指数方式增长，而是按照线性方式增长。

延迟应答：

如果接收数据的主机⽴刻返回ACK应答, 这时候返回的窗⼝可能⽐较⼩。

但实际上可能接收端的处理端的速度很快，很快就会将发送来的数据从缓冲区消费掉，这种情况下，接收端处理远不能达到自己的极限，即使窗口再大一些也可以处理过来。

因此引入延时应答机制，接收端每次都略等一会在应答，这时候返回的窗口大小就又会变大，从而在保证了网络不拥塞的情况下尽量提高了传输效率。窗口越大，网络吞吐量就越大，传输效率就越高。

捎带应答：

在延时应答的基础上，发送端与接收端在应用层也是“一发一收”的，在这个基础上，捎带应答就是令ACK与接收端回送的消息一起回给客户端。

面向字节流：

TCP通过字节流传输，即TCP将应用程序看成是一连串的无结构的字节流。每个TCP套接口有一个发送缓冲区，如果字节流太长时，TCP会将其拆分进行发送。当字节流太短时，TCP会等待缓冲区中的字节流达到一定程度时再构成报文发送出去，TCP发给对方的数据，对方在收到数据时必须给矛确认，只有在收到对方的确认时，本方TCP才会把TCP发送缓冲区中的数据删除。

3.什么是网络套接字（Socket）？流套接字（SOCK\_STREAM）基于什么协议？

3.1 套接字（socket）

TCP用主机的IP地址加上主机上的端口号作为TCP连接的端点，这种端点就叫做套接字(socket)或插口。

套接字用(IP地址:端口号)表示。

它是网络通信过程中端点的抽象表示，包含进行网络通信必需的五种信息:连接使用的协议，本地主机的IP地址，本地进程的协议端口，远地主机的IP地址，远地进程的协议端口。

3.2 流套接字（SOCK\_STREAM）

流式套接字用于提供面向连接、可靠的数据传输服务。该服务将保证数据能够实现无差错、无重复发送，并按顺序接收。流式套接字之所以能够实现可靠的数据服务，原因在于其使用了传输控制协议，即TCP(The Transmission Control Protocol)协议。

流式套接字提供没有记录边界的数据流:可以是双向的字节流(应用程序是全双工:可以通过套接字同时传输和接收)。可依赖流传递有序的、不重复的数据。(“有序”指数据包按发送顺序送达。”不重复”指一个特定的数据包只能获取一次。)这能确保收到流消息，而流非常适合处理大量数据。

4、IP首部，TCP首部，UDP首部

4.1 IP首部

版本：占4位，用于标识该数据报的IP协议的版本信息，对于IPv4，该字段值为4，对于IPv6，该字段值为6。无论是主机还是中间路由器，在处理每个接收到的IP数据报时，根据版本值以选择相应版本的IP协议模块来进行处理。

首部长度：占4位，用于表示IP数据报首部的长度，其值以32位（4字节）为单位，因此，IP首部长度必须是32位的整数倍。当IP数据报首部长度不是32位的整数倍时，必须用填充字段加以填充来补齐32位。如果IP数据报长度为20个字节，该字段值为5。

区分服务：占8位，用来获得更好的服务。这个字段在旧标准中称为服务类型ToS（ Type of Service）。它包括优先权字段（3位，现在已不用）、ToS字段（4位，分别表示最小时延、最大吞吐量、最高可靠性和最小费用）、未用字段（1位）。

总长度：占16位，总长度是指IP首部和数据部分长度之和，单位为字节。IP数据

报的最大长度为215－1＝65535字节。但是，这并不意味着IP数据报必须都按最大长度来组织。因为数据链路层存在着最大传送单元MTU（ Maximum Transfer Unit）或网络中可能存在其他因素的限制。

标识：占16位。标识的作用是当IP数据报在传输过程中由于低层网络传输限制而需要分片时，同一个P数据报的每个分片所共有的标记，表示这些分片属于同一个IP数据报，以便在目的主机重组这些分片时参照使用。标识的产生：IP层维持着个计数器，每产生一个IP数据报，计数器就加1，并将该值赋给标识字段。但这个“标识”并不是序号，因为IP是无连接服务，数据报不存在按序接收的问题。

标志：占3位。第1位保留；第2位为不分片标志，记为DF（Don＇ t Fragment），当DF＝1时，表示此IP数据报不允许被分片；当DF＝0时，表示此IP数据报允许被分片；第3位为更多分片标志，记为MF（ More Fragment），当MF＝1时，表示此分片不是最后分片，其后还有更多分片；当MF＝0时，表示此分片是最后分片，其后没有其他分片。

片偏移：占13位。片偏移的作用是指明此分片在原IP数据报（未分片前）中的绝对位置，其值以8字节为单位。因为IP数据报的传输受物理网络的MTU限制，当

个IP数据报较大，无法承载在数据链路层的一个帧中传输时，IP协议需要将IP数据报划分成多个较小的分片，并为每个分片构造一个单独的IP数据报，以适应物理网络的传输。

寿命TTL（ Time to Live）：也称为生存时间，占8位。TTL以秒为单位设置了该IP数据报在互联网中允许存在的时间。当IP数据报经过主机或路由器时，它们将对数据报进行处理，根据时间的消逝，递减TTL的值。一旦TTL递减至0，路由器就丢弃该数据报，并向源主机发送一个ICMP超时差错报告报文。但是由于计算IP

数据报延迟比较复杂，而且不准确。因此现在的路由器大多采用IP数据报所经过

的路由器数量来计算TTL值，每经过一个路由器，TTL减1，直到减为0为止，这种计算方法又称为跳数（Hop）衡量法。

协议：占8位，用于指出IP数据报携带的数据属于哪一种高层协议，该字段的值是高层协议对应的编号。

首部校验和：占16位。用用于保证IP数据报首部数据的完整性。其计算方法是将首部看作一个16位的整数序列，对每个整数分别进行二进制反码相加，然后再对计算的结果求反。

源IP地址和目的IP地址：各占32位，分别为发送主机的IP地址和接收主机的IP地址。IP数据报在传输过程中可能经过多个中间路由器，但这两个字段值在传送期间始终不变。

IP选项：可选项，长度可变。常用于网络测试和调试。目前常用的IP选项有记录路由选项、源路由选项和时间戳选项等。

记录路由选项：用来监视和控制互联网路由器如何路由数据报。

源路由选项：分为严格源路由选项和宽松源路由选项。严格源路由选项规定IP数据报必须严格按照指定的路径到达目的主机，否则报告错误。宽松源路由选项要求IP数据报必须沿着IP地址表中的地址序列传输，但允许表中相继两个地址之间经过其他的IP地址。

时间戳选项：用于测量IP数据报路由过程中所经过的每个路由器的时间，便于对路由的性能进行分析。

填充：可可选项，长度可变，取决于IP选项的长度，其目的是保证数据报首部为32位的整数倍。填充时，填充内容全为“0”。

4.2 UDP协议

源端口和目的端口：源端口和目的端口字段各为16位（2字节），它是UDP协议的端口号，其中源端口是可选的（因为UDP不需要反馈信息，因此源端口基本不起作用），目的端口必须填写。若源端口不选，则取值为0。

长度：长度字段为16位（2字节），它是指UDP报文的总长度，包括UDP首部和用户数据两部分，长度以字节为单位。

校验和：校校验和字段为16位（2字节），UDP的校验和字段是保证UDP数据正确的惟一手段。计算UDP校校验和时必须包括UDP伪首部、UDP首部和用户数据3个部分。

4.3 TCP协议

源端口号和目的端口号：各占2字节，分别表示发送方和接收方的应用程序端口号。

序列号：占4字节，指派给该报文段第一个数据字节的一个号，表示该数据在发送方的数据流中的位置。初始序列号一般并不是从1开始的，而是根据采用某些算法计算出的一个数值作为起始序列号（5．3．3节讲述）。这样做的目的是为避免出现重复序列号。

确认号：占4字节，确认号是接收方期望收到对方的下一个报文段数据的第一个字节的序号，也就是期望收到对方的下一个TCP报文段首部序列号字段的值。

首部长度：占4位，表示TCP报文首部信息的长度。由于首部可能含有选项内容，因此TCP首部的长度是不确定的。首部长度的单位不是字节而是32位（以4字节为计算单位），其范围是5～15，对应首部的长度是20～～60字节。首部长度也指示了数据区在报文段中的起始偏移值。

保留：占6位，保留为今后使用。目前为全0。

紧急标志URG（U）：占1位，当URG＝1时，表示紧急指针字段有效。通知发送方本数据报文段中含有紧急数据，需要马上传输，这时发送方不会等到缓冲区满再发送，而是直接优先将该报文段发送出去。

确认标志ACK（A）：占1位，当ACK＝1时，表示确认号字段有效。

推送标志PSH（P）：占1位，PSH＝1时，表示当前报文段需要请求推送（Push）操作，即接收方TCP收到推送标志为1的报文时，就立即提交给接收的应用进程，而不必等到整个缓存都填满后再向上提交。

复位标志RST（R）：占1位，当RST＝1时，表表示TCP连接中出现严重差错（如主机崩溃或其他原因），必须先释放连接，然后再重新建立传输连接。

同步标志SYN（S）：占1位，当SYN＝1时，表示这是一个连接请求或连接接收报文，用于建立传输连接。

终止标志FIN（F）：占1位，当FIN＝1时，表示此报文段的发送方的数据已发送完毕，并要求释放传输连接。

窗口大小：占2字节，它是窗口通告值。该值由接收方设置，发送方根据接收到的窗口通告值来来调整发送窗口的大小。通过窗口机制，可以控制发送方发送的数据量，实现流量控制。

校验和：占2字节，TCP的校验和是必选项，它的计算方法与UDP校验和的计算方法相同，同样需要包含TCP伪首部。TCP伪首部中的协议类型值为6（UDP伪首部中的协议类型为17）

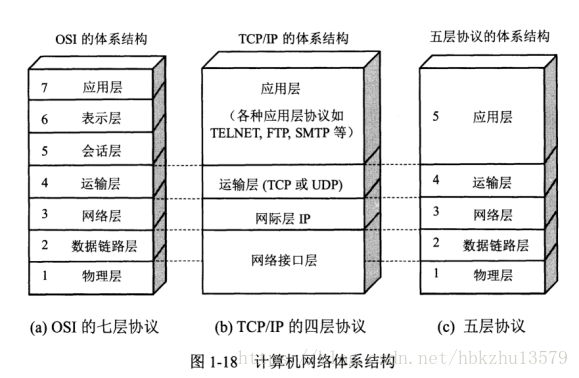
紧急指针：占2字节，与紧急标志URG配合使用。在TCP通信中，如果一方有紧急的数据（例如中断或退出命令等）需要尽快发送给另一方，并且让接收方的TCP协议尽快通知相应的应用程序时，可以将URG位置1，并通过紧急指针指示紧急数在报文段中的结束位置。

选项：长度可变，可以是一个或多个字节，规定相应的功能。每个选项由类型、长度、数据3部分组成。

MSS：用于TCP连接双方在建立连接时相互告知对方期望的最大报文段长度MSS值。

填充：为了使选项字段对齐32位，可以采用若干0作为填充数据。

**1. 计算机网络体系结构\***



**1. 五层协议**

应用层 ：为特定应用程序提供数据传输服务，例如 HTTP、DNS 等。数据单位为**报文**。

运输层 ：提供的是进程间的通用数据传输服务。由于应用层协议很多，定义通用的运输层协议就可以支持不断增多的应用层协议。  
运输层包括两种协议：  
传输控制协议 TCP，提供面向连接、可靠的数据传输服务，数据单位为报文段；  
用户数据报协议 UDP，提供无连接、尽最大努力的数据传输服务，数据单位为用户数据报。  
TCP 主要提供完整性服务，UDP 主要提供及时性服务。

网络层 ：为主机间提供数据传输服务，而运输层协议是为主机中的进程提供服务。网络层把运输层传递下来的**报文段**或者**用户数据报**封装成**分组**。

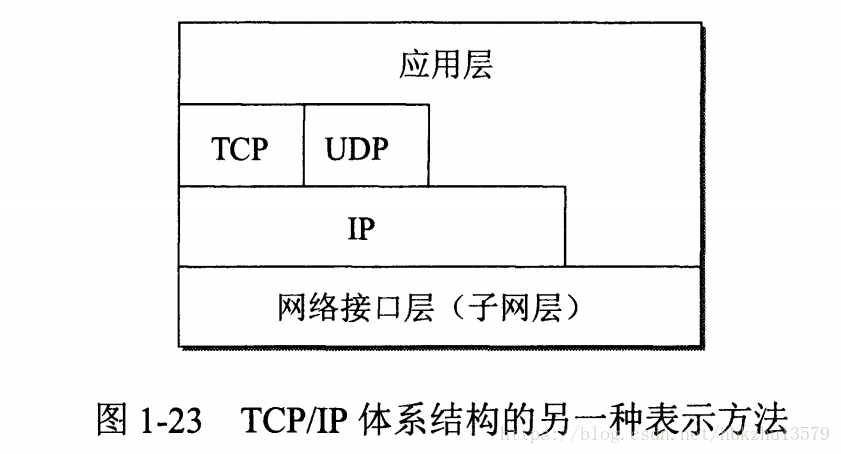
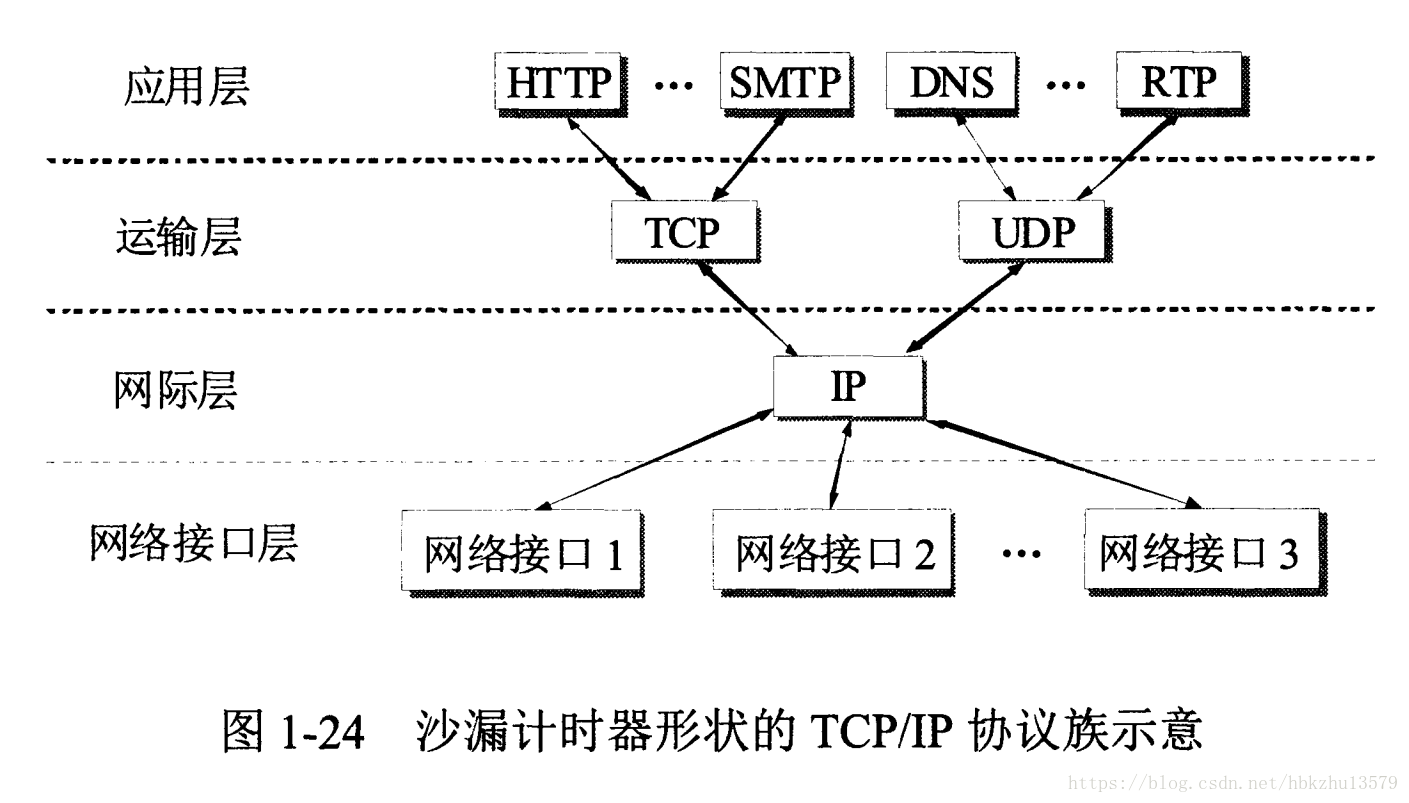
数据链路层 ：网络层针对的还是**主机之间**的数据传输服务，而主机之间可以有很多链路，链路层协议就是为同一链路的主机提供服务。数据链路层把网络层传下来的**分组**封装成**帧**。

物理层 ：考虑的是怎样在传输媒体上传输数据**比特流**，而不是指具体的传输媒体。物理层的作用是尽可能屏蔽传输媒体和通信手段的差异，使数据链路层感觉不到这些差异。

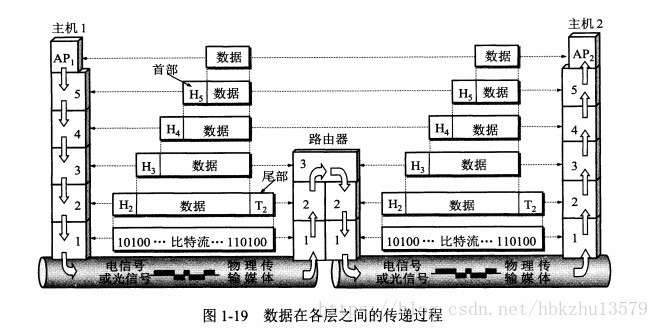
**2. OSI（七层协议 多了表示层和会话层）**

其中表示层和会话层用途如下：  
表示层 ：数据压缩、加密以及数据描述，这使得应用程序不必担心在各台主机中数据内部格式不同的问题。  
会话层 ：建立及管理会话。  
五层协议没有表示层和会话层，而是将这些功能留给应用程序开发者处理。

**3. TCP/IP**

它只有四层，相当于五层协议中数据链路层和物理层合并为网络接口层。  
TCP/IP 体系结构不严格遵循 OSI 分层概念，应用层可能会直接使用 IP 层或者网络接口层。  
  
TCP/IP 协议族是一种沙漏形状，中间小两边大，IP 协议在其中占用举足轻重的地位。  


**4. 数据在各层之间的传递过程**

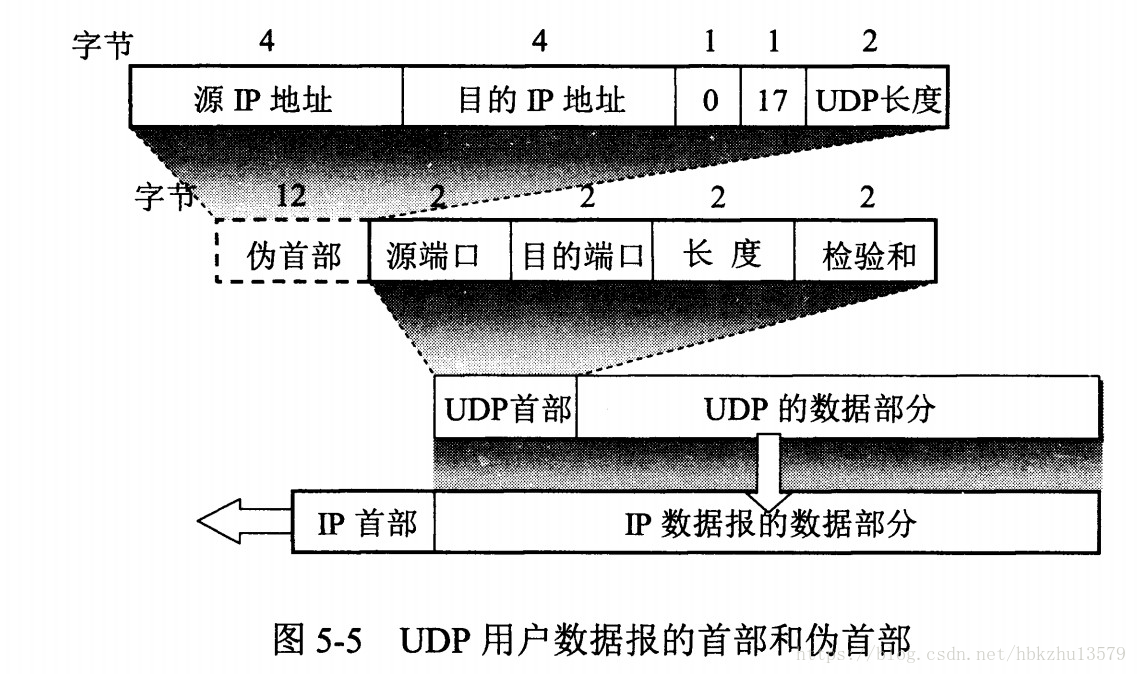
在向下的过程中，需要添加下层协议所需要的首部或者尾部，而在向上的过程中不断拆开首部和尾部。  
路由器只有下面三层协议，因为路由器位于网络核心中，不需要为进程或者应用程序提供服务，因此也就不需要运输层和应用层。  


**2. 传输层\*  
对于开发人员来说，重点掌握上层的协议就差不多了，传输层是核心**

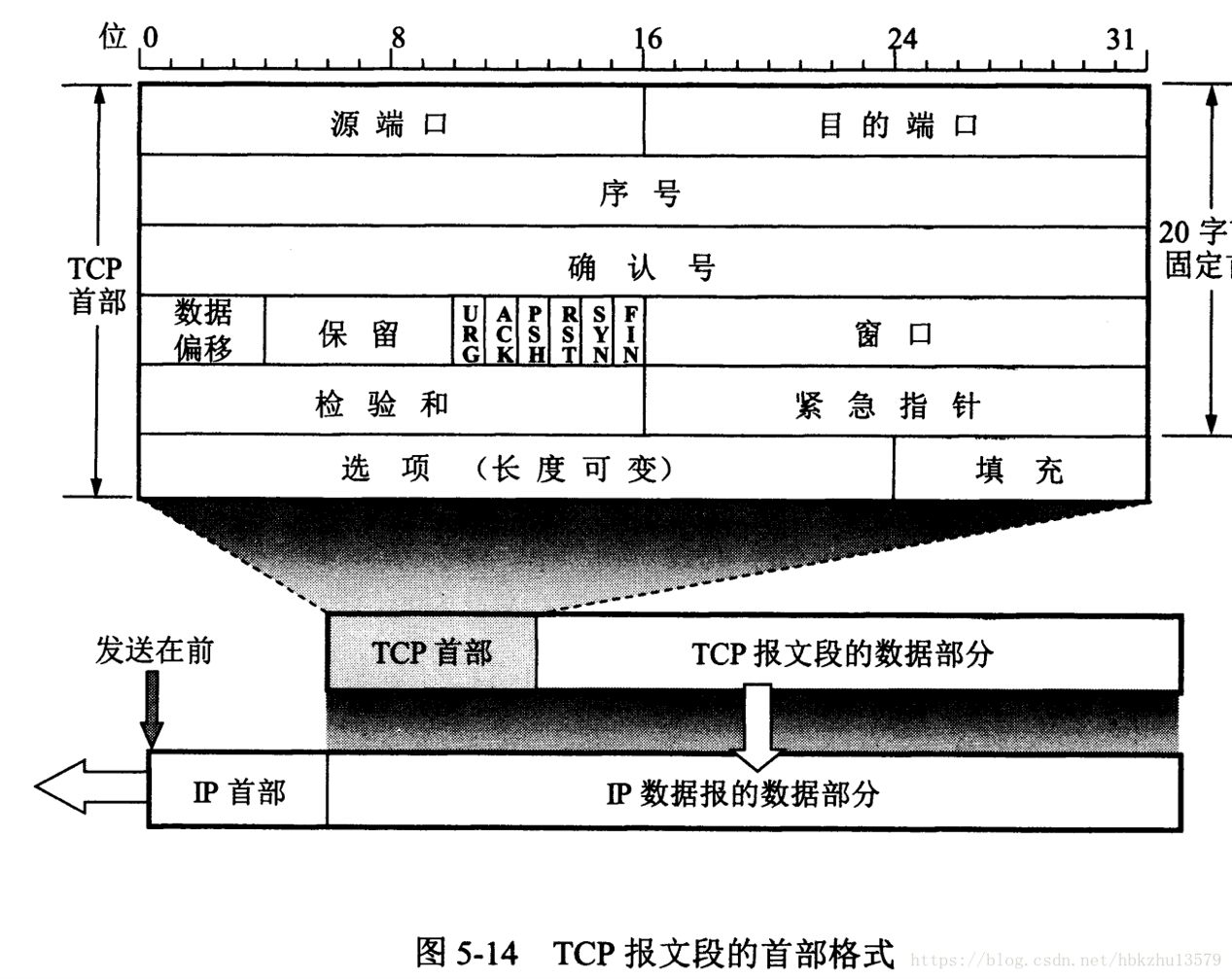
**1.UDP 和 TCP 的特点**

用户数据报协议 UDP（User Datagram Protocol）是无连接的，尽最大可能交付，没有拥塞控制，面向报文（对于应用程序传下来的报文不合并也不拆分，只是添加 UDP 首部），支持一对一、一对多、多对一和多对多的交互通信。  
传输控制协议 TCP（Transmission Control Protocol）是面向连接的，提供可靠交付，有流量控制，拥塞控制，提供全双工通信，面向字节流（把应用层传下来的报文看成字节流，把字节流组织成大小不等的数据块），每一条 TCP 连接只能是点对点的（一对一）。

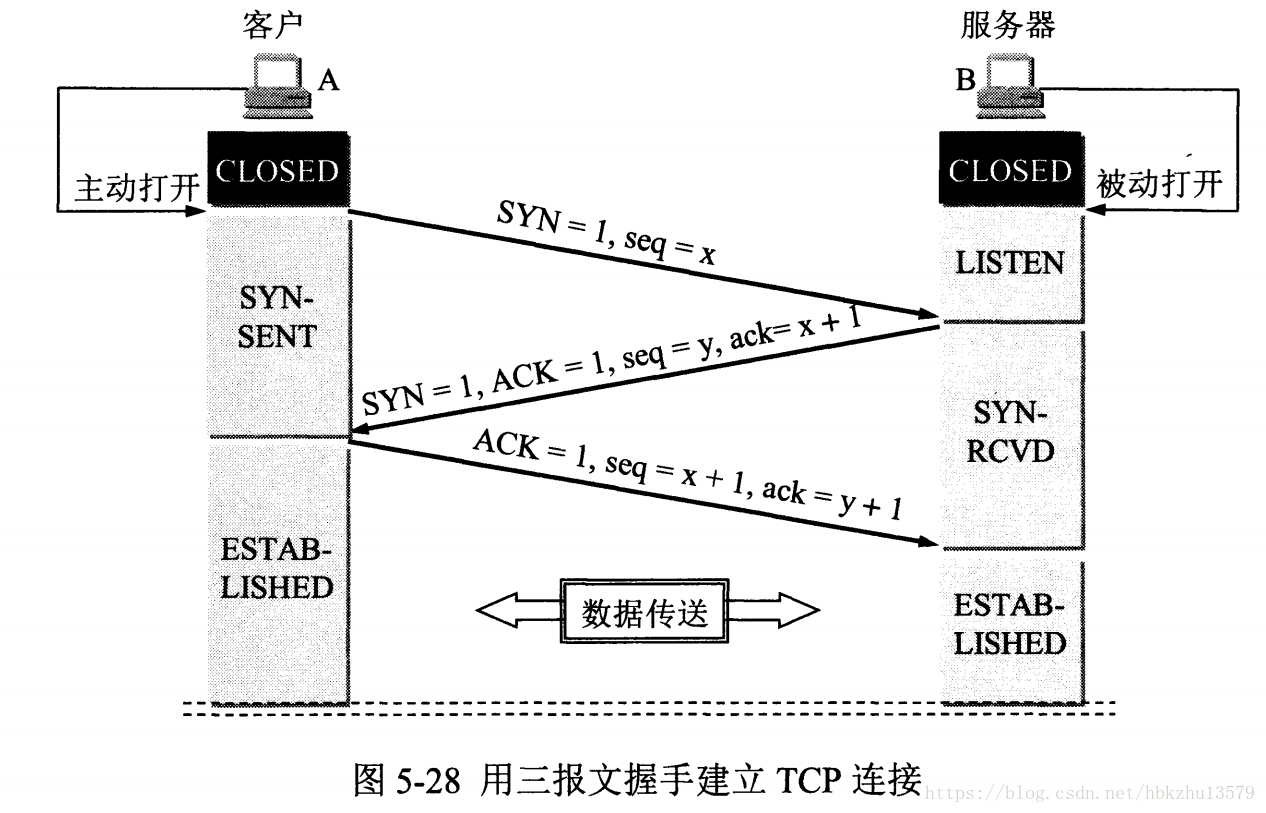
**2.UDP 首部格式**

  
首部字段只有 8 个字节，包括源端口、目的端口、长度、检验和。12 字节的伪首部是为了计算检验和临时添加的。

**3.TCP 首部格式**

  
**序号** ：用于对字节流进行编号，例如序号为 301，表示第一个字节的编号为 301，如果携带的数据长度为 100 字节，那么下一个报文段的序号应为 401。  
**确认号** ：期望收到的下一个报文段的序号。例如 B 正确收到 A 发送来的一个报文段，序号为 501，携带的数据长度为 200 字节，因此 B 期望下一个报文段的序号为 701，B 发送给 A 的确认报文段中确认号就为 701。  
**数据偏移** ：指的是数据部分距离报文段起始处的偏移量，实际上指的是首部的长度。  
确认 ACK ：当 ACK=1 时确认号字段有效，否则无效。TCP 规定，在连接建立后所有传送的报文段都必须把 ACK 置 1。  
**同步 SYN** ：在连接建立时用来同步序号。当 SYN=1，ACK=0 时表示这是一个连接请求报文段。若对方同意建立连接，则响应报文中 SYN=1，ACK=1。  
**终止 FIN** ：用来释放一个连接，当 FIN=1 时，表示此报文段的发送方的数据已发送完毕，并要求释放连接。  
**窗口** ：窗口值作为接收方让发送方设置其发送窗口的依据。之所以要有这个限制，是因为接收方的数据缓存空间是有限的。

**3.TCP 的三次握手**

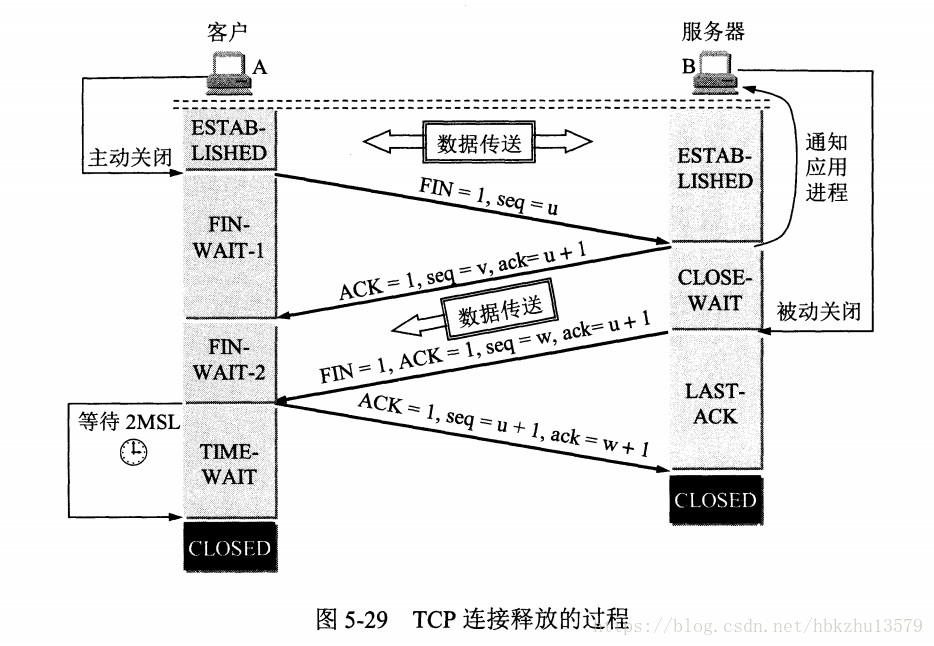
  
假设 A 为客户端，B 为服务器端。  
首先 B 处于 LISTEN（监听）状态，等待客户的连接请求。

1. A 向 B 发送连接请求报文，SYN=1，ACK=0，选择一个初始的序号 x。
2. B 收到连接请求报文，如果同意建立连接，则向 A 发送连接确认报文，SYN=1，ACK=1，确认号为 x+1，同时也选择一个初始的序号 y。
3. A 收到 B 的连接确认报文后，还要向 B 发出确认，确认号为 y+1，序号为 x+1。
4. B 收到 A 的确认后，连接建立。

**三次握手的原因**

第三次握手是为了防止失效的连接请求到达服务器，让服务器错误打开连接。  
客户端发送的连接请求如果在网络中滞留，那么就会隔很长一段时间才能收到服务器端发回的连接确认。客户端等待一个超时重传时间之后，就会重新请求连接。但是这个滞留的连接请求最后还是会到达服务器，如果不进行三次握手，那么服务器就会打开两个连接。如果有第三次握手，客户端会忽略服务器之后发送的对滞留连接请求的连接确认，不进行第三次握手，因此就不会再次打开连接。

**4.TCP 的四次挥手**

  
以下描述不讨论序号和确认号，因为序号和确认号的规则比较简单。并且不讨论 ACK，因为 ACK 在连接建立之后都为 1。

1. A 发送连接释放报文，FIN=1。
2. B 收到之后发出确认，此时 TCP 属于半关闭状态，B 能向 A 发送数据但是 A 不能向 B 发送数据。
3. 当 B 不再需要连接时，发送连接释放报文，FIN=1。
4. A 收到后发出确认，进入 TIME-WAIT 状态，等待 2 MSL（最大报文存活时间）后释放连接。
5. B 收到 A 的确认后释放连接。

**四次挥手的原因**

客户端发送了 FIN 连接释放报文之后，服务器收到了这个报文，就进入了 CLOSE-WAIT 状态。这个状态是为了让服务器端发送还未传送完毕的数据，传送完毕之后，服务器会发送 FIN 连接释放报文。

**TIME\_WAIT**

客户端接收到服务器端的 FIN 报文后进入此状态，此时并不是直接进入 CLOSED 状态，还需要等待一个时间计时器设置的时间 2MSL。这么做有两个理由：  
确保最后一个确认报文能够到达。如果 B 没收到 A 发送来的确认报文，那么就会重新发送连接释放请求报文，A 等待一段时间就是为了处理这种情况的发生。  
等待一段时间是为了让本连接持续时间内所产生的所有报文都从网络中消失，使得下一个新的连接不会出现旧的连接请求报文。

**5.TCP 可靠传输**

TCP可靠传输方面，主要采用以下4中机制：

1. 字节编号机制。TCP 数据段以字节为单位对数据段中的“数据”部分进行一一编号，确保每个字节的数据都可以有序传送和接受。
2. 数据段确认机制。TCP 要求每接受一个数据段都必须由接收端向发送端返回一个确认数据段（可以用一个去人数据段一次确认全面多个数据段），其中的“确认号”表明接收端已正确接受的数据段序号（“确认号”前面的所有数据段，确认号表示将要接收的下一个数据段编号）。
3. 超时重传机制。在 TCP 中有一个重传定时器（Retransmission Timer， RTT），在发送一个数据段的同时也启动了该定时器。如果在定时器过期之前该数据段还没有被对方确认的话，且定时器停止，然后重传对应序号的数据段。
4. 选择性确认（Selective ACK， SACK）机制。在 SACK 支持下，仅可以重传缺少部分的数据，而不会重传那些已经正确接受的数据段。

**3.应用层**

**1.常用端口**



**4. Web 页面请求过程**

这里采用了《计算机网络——自顶向下》这本书中的说法，比较全面，面试可以简化。

**1. DHCP 配置主机信息**

假设主机最开始没有 IP 地址以及其它信息，那么就需要先使用 DHCP 来获取。  
主机生成一个 DHCP 请求报文，并将这个报文放入具有目的端口 67 和源端口 68 的 UDP 报文段中。  
该报文段则被放入在一个具有广播 IP 目的地址(255.255.255.255) 和源 IP 地址（0.0.0.0）的 IP 数据报中。  
该数据报则被放置在 MAC 帧中，该帧具有目的地址 FF:FF:FF:FF:FF:FF，将广播到与交换机连接的所有设备。  
连接在交换机的 DHCP 服务器收到广播帧之后，不断地向上分解得到 IP 数据报、UDP 报文段、DHCP 请求报文，之后生成 DHCP ACK 报文，该报文包含以下信息：IP 地址、DNS 服务器的 IP 地址、默认网关路由器的 IP 地址和子网掩码。该报文被放入 UDP 报文段中，UDP 报文段有被放入 IP 数据报中，最后放入 MAC 帧中。  
该帧的目的地址是请求主机的 MAC 地址，因为交换机具有自学习能力，之前主机发送了广播帧之后就记录了 MAC 地址到其转发接口的交换表项，因此现在交换机就可以直接知道应该向哪个接口发送该帧。  
主机收到该帧后，不断分解得到 DHCP 报文。之后就配置它的 IP 地址、子网掩码和 DNS 服务器的 IP 地址，并在其 IP 转发表中安装默认网关。

**2.ARP 解析 MAC 地址**

主机通过浏览器生成一个 TCP 套接字，套接字向 HTTP 服务器发送 HTTP 请求。为了生成该套接字，主机需要知道网站的域名对应的 IP 地址。  
主机生成一个 DNS 查询报文，该报文具有 53 号端口，因为 DNS 服务器的端口号是 53。  
该 DNS 查询报文被放入目的地址为 DNS 服务器 IP 地址的 IP 数据报中。  
该 IP 数据报被放入一个以太网帧中，该帧将发送到网关路由器。  
DHCP 过程只知道网关路由器的 IP 地址，为了获取网关路由器的 MAC 地址，需要使用 ARP 协议。  
主机生成一个包含目的地址为网关路由器 IP 地址的 ARP 查询报文，将该 ARP 查询报文放入一个具有广播目的地址（FF:FF:FF:FF:FF:FF）的以太网帧中，并向交换机发送该以太网帧，交换机将该帧转发给所有的连接设备，包括网关路由器。  
网关路由器接收到该帧后，不断向上分解得到 ARP 报文，发现其中的 IP 地址与其接口的 IP 地址匹配，因此就发送一个 ARP 回答报文，包含了它的 MAC 地址，发回给主机。

**3.DNS 解析域名**

知道了网关路由器的 MAC 地址之后，就可以继续 DNS 的解析过程了。  
网关路由器接收到包含 DNS 查询报文的以太网帧后，抽取出 IP 数据报，并根据转发表决定该 IP 数据报应该转发的路由器。  
因为路由器具有内部网关协议（RIP、OSPF）和外部网关协议（BGP）这两种路由选择协议，因此路由表中已经配置了网关路由器到达 DNS 服务器的路由表项。  
到达 DNS 服务器之后，DNS 服务器抽取出 DNS 查询报文，并在 DNS 数据库中查找待解析的域名。  
找到 DNS 记录之后，发送 DNS 回答报文，将该回答报文放入 UDP 报文段中，然后放入 IP 数据报中，通过路由器反向转发回网关路由器，并经过以太网交换机到达主机。

**4. HTTP 请求页面**

有了 HTTP 服务器的 IP 地址之后，主机就能够生成 TCP 套接字，该套接字将用于向 Web 服务器发送 HTTP GET 报文。  
在生成 TCP 套接字之前，必须先与 HTTP 服务器进行三次握手来建立连接。生成一个具有目的端口 80 的 TCP SYN 报文段，并向 HTTP 服务器发送该报文段。  
HTTP 服务器收到该报文段之后，生成 TCP SYN ACK 报文段，发回给主机。  
连接建立之后，浏览器生成 HTTP GET 报文，并交付给 HTTP 服务器。  
HTTP 服务器从 TCP 套接字读取 HTTP GET 报文，生成一个 HTTP 响应报文，将 Web 页面内容放入报文主体中，发回给主机。  
浏览器收到 HTTP 响应报文后，抽取出 Web 页面内容，之后进行渲染，显示 Web 页面。