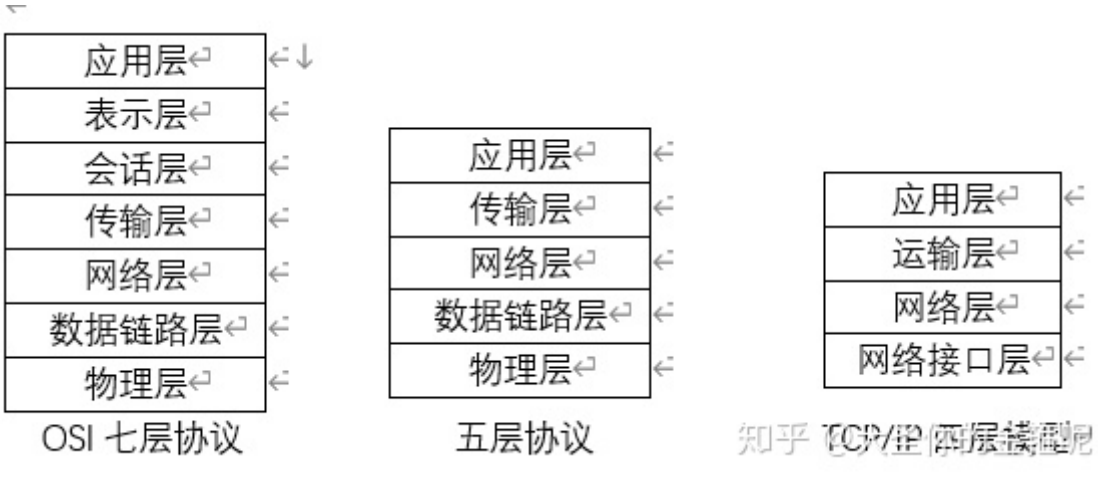
计算机网络

第一章概述

1.1 协议和服务之间的联系和区别

（1）首先，**协议**是对等层实体之间通信的**规则集。**在**协议的控制**下，两个对等实体间的通信使得本层能够**向上一层提供服务。**要**实现本层协议**，还需要使用**下面一层**所提供的服务。**本层**只能看见**本层的服务**，看不见**下层的协议**，即下层的协议对上层服务的实体而言是**透明**的。
（2）协议是“**水平的**”，即协议是控制对等实体之间通信的规则。但服务是“**垂直的**”，即服务是由下层向上层通过**层间接口**提供的。

1.2 计算机网络有哪些层?



五层协议各层的功能:

应用层（对应七层协议中的应用层、表示层、会话层）

通过**一台主机**内**进程间的交互**来完成**特定网络应用**，

包含的主要协议：FTP（文件传送协议）、Telnet（远程登录协议）、DNS（域名解析协议）、SMTP（邮件传送协议），POP3协议（邮局协议），HTTP协议（Hyper Text Transfer Protocol），**数据单位为报文。**

运输层
提供的是**两台主机**中**进程间**的**通用数据传输服务。**

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**；TCP 主要提供完整性服务。

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。**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**分组或包。**在TCP/IP体系中，网络层使用IP协议，因此分组也叫做**IP数据报。**

数据链路层

链路层协议就是为**同一链路**的**结点之间**提供服务。数据链路层把网络层传来的分组封装成帧，在两个相邻结点间的链路上传送**帧**。

物理层

考虑的是怎样在物理媒体上**传输数据比特流**，而不是指具体的物理媒体（传输媒体指双绞线、同轴电缆等）。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。

OSI各层的功能：

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关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。

TCP/IP各层的功能：

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。

1.3 面向连接的服务以及无连接的服务

（1）**面向连接的服务**：是指通信前需要建立连接，通信结束后需要进行连接释放。整个过程包括，连接建立、数据传送、连接释放。是**按序**传送、**可靠**传送的。
（2）**无连接的服务**：传送数据之前不许要建立连接，随时传就行，速度快，简单，但是无法避免数据的**丢失、重复**等只能“尽最大努力地交付”，是不可靠地传输。

1.4 对等层、实体、协议概念

对等层
两个相同层次的层之间，好像把数据通过水平虚线直接传递给对方，就叫做对等层。

实体
实体是任何可**发送或接收**信息的**硬件或软件进程。**

协议
协议是控制两个对等实体进行通信的规则的集合。

在协议的控制下，两个对等实体间的通信使得本层能够向上一层提供服务。要实现本协议，还需要使用下面一层所提供的服务。

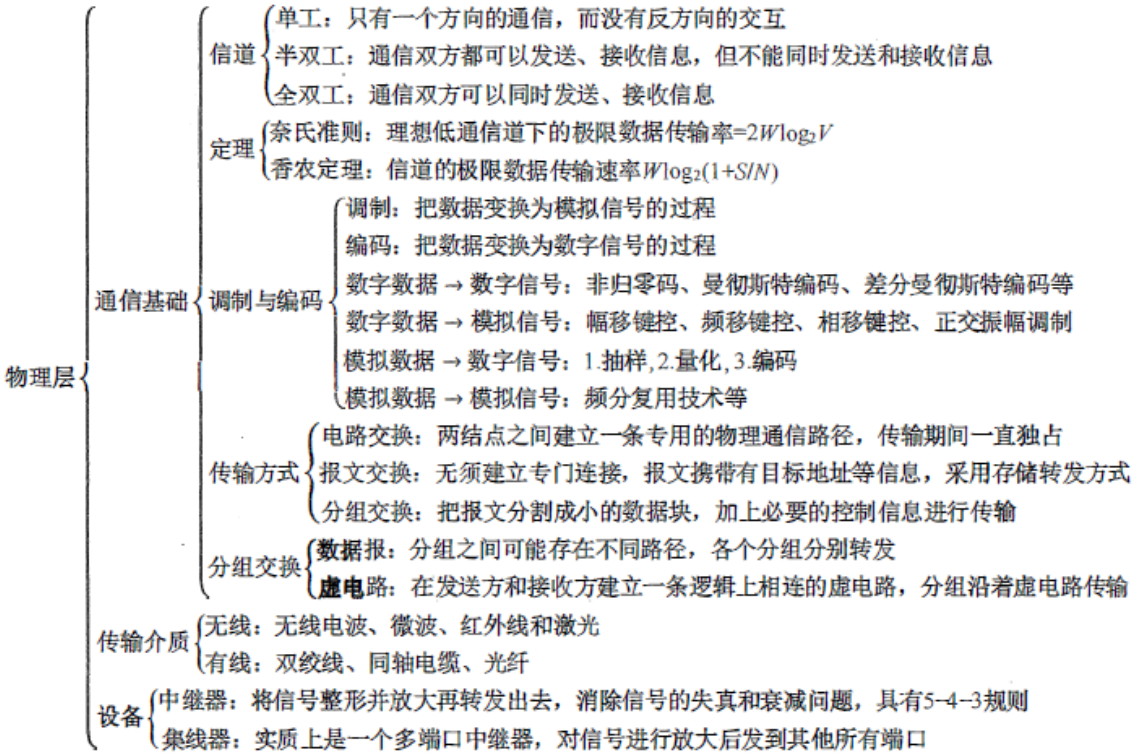
协议是“水平的”，即协议是控制对等实体之间通信的规则。但服务是“垂直的”，即服务是由下层向上层通过层间接口提供的。

1.5主机间的通信方式

> **客户-服务器（C/S）**： 客户是服务的请求方，服务器是服务的提供方。

> **对等（P2P）**： 不区分客户和服务器。

第二章物理层



2.1 物理层作用

主要是规定了通信结点和通信链路之间的连接的接口的一些特性。包括机械特性、电气特性、功能特性和过程特性。规定了通信链路上传输的信号的意义以及电气特性。功能是在物理媒体上为数据端的设备**透明的传输原始的比特流。**

2.2 物理层主要设备

（1）**中继器**：起到的是**信号再生**的作用（数字信号），与放大器不同（模拟信号）。
（2）**集线器**：多端口的中继器。
（3）注意：物理层的设备均不支持存储转发功能，所以连接的两端必须使用同一个协议，且连接的两端是同一个局域网的不同的网段。
另外，**物理层的设备不阻隔冲突域，也不阻隔广播域**，它就是个最便宜的设备。

2.3 数据交换的方式

（1）**电路交换**：**整个报文**的比特流从源点**连续**的直达终点，像在一个**管道**中传输。包括建立连接、传输数据和断开连接**三个阶段**。最典型的电路交换网络是传统电话网络。

建立连接的**时间比较长。**且**独占**信道，适合传输**数据量大**的时候，信道**利用率低**且**一旦出差错**就会导致数据传输失败，无法控制。

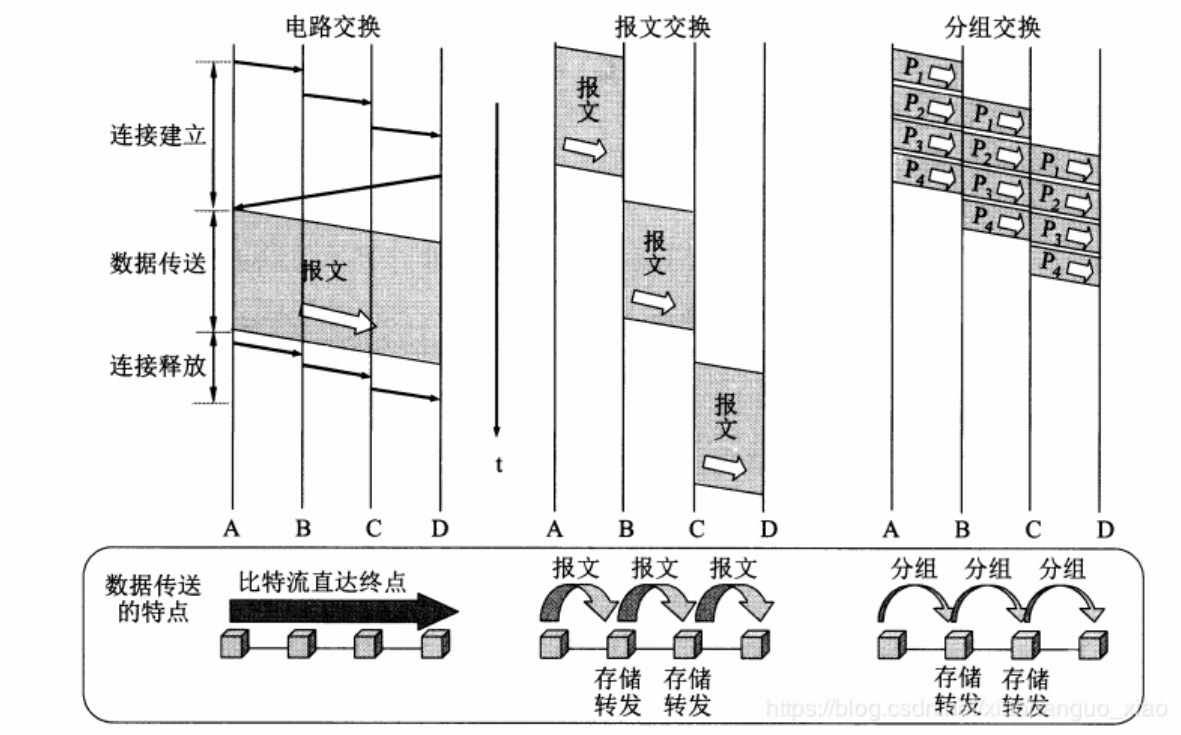
（2）**报文交换**：将**整个报文**转发到相邻节点，**全部存储**下来，查找**转发表**，转发到下一个节点。是**存储-转发**类型的网络。

每个结点接受**整个报文**之后进行检测，检测无误执行转发，故会存在延迟时间。缺点就是**报文大小**不确定，结点的**存储管理**不好控制，且**延迟时间**较长。

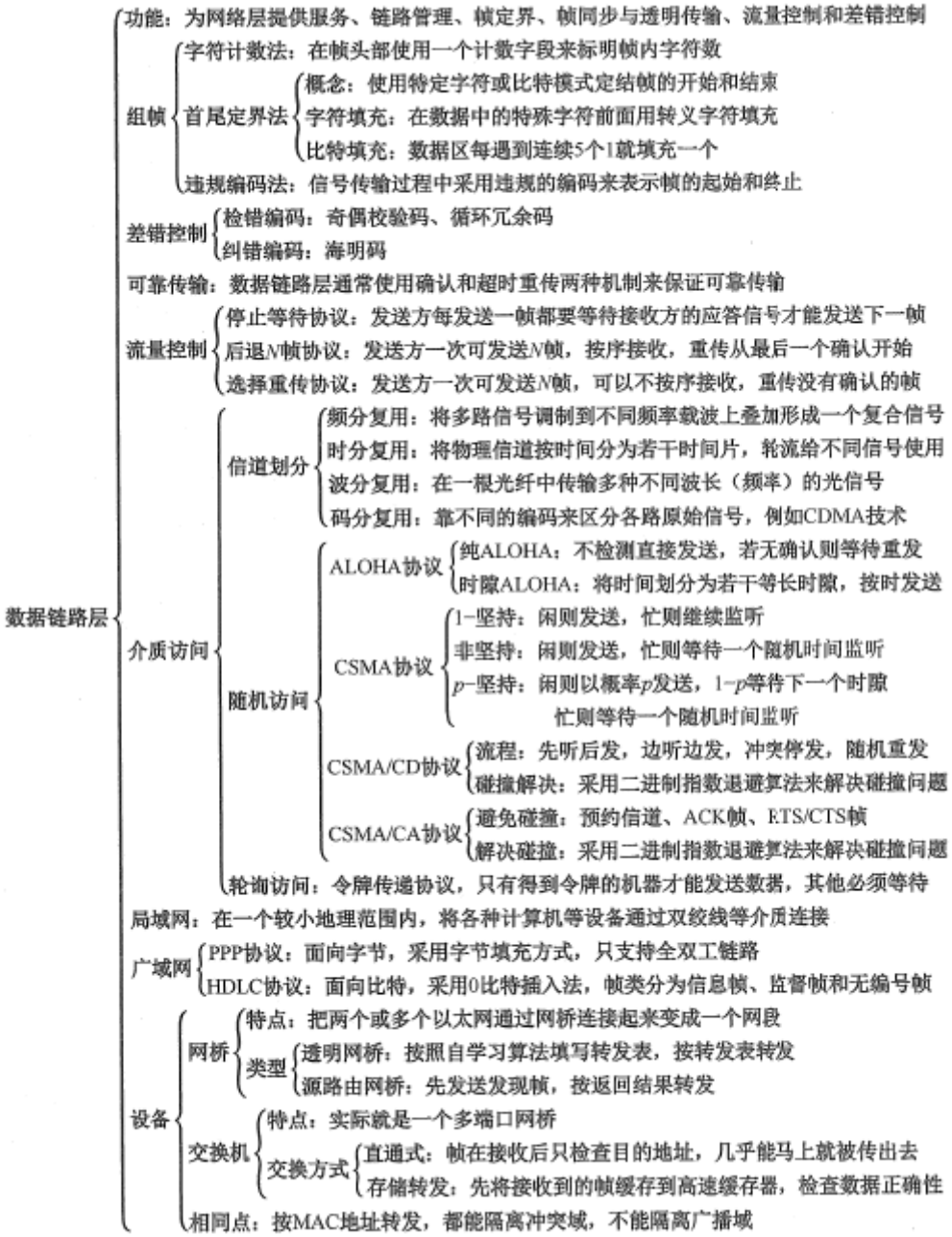
（3）**分组交换**：将**报文分组**转发到相邻节点，查找**转发表**，转发到下一个节点。也是存储-转发类型的网络。

将报文划分为**长度相同**的分组，每个分组都有相应的原地址+目的地址+序号等信息，进行存储转发操作。好处是，分组可以可以**并行进行转发**，延迟时间缩短且因为分组大小固定所以**易于存储的管理。**缺点就是，因为每个分组都要加地址信息以及编号之类的，所以**增加了传输的数据量。**

分组交换又分为：**数据报交换**与**虚电路交换**（虚电路之所以是“虚”的，是因为这条电路不是专用的，每个节点到其他节点之间的链路可能有若干虚电路通过，也可能同时与多个节点之间建立虚电路）



第三章数据链路层



3.1 数据链路层的功能

在物理层提供服务的基础上为网络层提供服务。主要是加强物理层传输原始比特流的能力，将物理层提供的可能出错的物理连接改造为逻辑上无差错的数据链路。

3.2 为什么要进行流量控制

由于接收发送双方各自的**工作速率**和**缓存空间**的差异，可能出现发送方的发送能力大于接收方的接收能力的现象，如若此时不适当限制发送方的发送速率（即链路上的信息流量），前面来不及接收的帧将会被后面不断发送的帧 " 淹没",造成帧的丢失而出错。

因此流量控制实际上就是限制发送方的数据流量，使其发送速率不超过接收方的接收能力。

3.3 数据链路层主要完成的功能

包括为网络层提供服务、链路管理、帧定界、帧同步、透明传输、差错控制、流量控制、随机访问介质访问控制（协议）

3.4 组帧的定义

组帧：包括**帧定界**、**帧同步**、**透明传输**

数据链路层将从网络层获得的IP数据包封装成帧，然后进行传输，封装成帧的时候要添加帧首部和帧尾部，用于提取出数据部分。

3.5 组帧的方式

（1）**字符计数法**：在帧首部添加计数字段，记录该帧数据的长度。
（2）**字符填充的首位定界法**：用特殊的字符去作为首位定界符。对于数据中可能出现的特殊字符，则在其前面加上一个转义字符，接收的时候自动去掉即可。
（3）**零比特填充的首位定界标志**：01111110作为开始和结束的表示，为了防止数据中出现这样的串而出现提前终止的情况，我们对数据中连续的11111后面加一个0，接收的时候相反处理即可，硬件处理。
（4）**违规编码法**：适用于冗余编码的时候，比如我们的差分曼彻斯特编码（由高到低表示0，则由低到高表示1，高高和低低就是违规编码），我们用违规编码作为定界符即可，但是只适用于冗余编码。

3.6 差错控制

检错编码（奇偶校验、CRC）和纠错编码（海明码）

3.7 流量控制

就是限制发送方的发送速度，使得接收方可以正确的接收数据。
（1）停止等待协议：发送窗口大小=1，接受窗口大小=1

> 发送方每发送一帧，**都要等待接收方的应答信号，之后才能发送下一帧**；**接收方每接收一帧，都要反馈一个应答信号**，表示可接收下一帧，如果接收方不反馈应答信号，那么**发送方必须一直等待**。每次只允许发送一帧，然后就陷入等待接收方确认信息的过程中，因而传输效率很低。

（2）滑动窗口流量控制方式基本原理

> 在任意时刻，发送方都维持一组**连续**的**允许发送**的**帧的序号**，称为**发送窗口**；同时接收方也维持一组连续的允许接收帧的序号，称为**接收窗口。发送窗口的大小**代表在还**未收到对方确认信息**的情况下发送方**最多还可以发送多少个数据帧**。同理，在接收端设置接收窗口是为了**控制可以接收哪些数据帧和不可以接收哪些帧**。在接收方，只有收到的数据帧的序号落入接收窗口内时，才允许将该数据帧收下，否则丢弃。

（4）后退N帧（GBN）：发送窗口大小>1，接受窗口大小=1

> 在后退N 帧式ARQ 中，发送方无须在收到上一个帧的ACK 后才能开始发送下一帧，而是可以连续发送帧。当接收方检测出失序的信息帧后，要求发送方**重发最后一个正确接收的信息帧之后的所有未被确认的帧**；或者当发送方发送了N 个帧后，若发现该N 个帧的前一个帧在计时器超时后仍未返回其确认信息，则该帧被判为出错或丢失，此时发送方就不得不重传该出错帧及随后的N 个帧。换句话说，**接收方只允许按顺序接收帧。**（接收窗口大小=1则按序接收）

（5）选择重发：发送窗口大小>1，接受窗口大小>1

> 设法只重传出现差错的数据帧或计时器超时的数据帧，但此时必须**加大接收窗口**，以便先收下**发送序号不连续但仍处在接收窗口中**的那些数据帧。等到所缺序号的数据帧收到后再一并送交主机。这就是**选择重传ARQ 协议。**
>
> 在选择重传协议中，每个发送缓冲区对应一个计时器，当计时器超时时，缓冲区的帧就会重传。另外，一旦接收方怀疑帧出错，就会发一个否定帧NAK给发送方，要求发送方对NAK中指定的帧进行重传。

（2）和（3）有一个共同点就是t+w<=2^n,n为编号的二进制数，否则容易产生若确认帧丢失接收方无法分清新帧旧帧的情况。

3.8 可靠传输机制

数据链路层的可靠传输通常使用**确认**和**超时重传**两种机制来完成。

确认是一种无数据的控制帧，这种控制帧使得**接收方可以让发送方**知道**哪些内容被正确接收。**有些情况下为了提高传输效率，将确认捎带在一个回复帧中，称为**捎带确认。**

超时重传是指发送方在发送某个数据帧后就**开启一个计时器**，在一定时间内如果没有得到发送的数据帧的确认帧，那么就重新发送该数据帧，直到发送成功为止。

> 自动重传请求(Auto Repeat reQuest， ARQ) 通过**接收方请求发送方重传出错的数据帧**来恢复出错的帧。
>
> 传统自动重传请求分为三种，即停止 - 等待(Stop-and-Wait) ARQ 、后退N 帧(Go-Back-N) ARQ 和选择性重传(Selective Repeat)ARQ,后两种协议是滑动窗口技术与请求重发技术的结合，由于窗口尺寸开到足够大时，帧在线路上可以连续地流动，因此又称其为**连续ARQ 协议。**

3.9 随机访问介质访控制机制

在局域网中是总线型的，广播式的发送，所以必须为了防止冲突采取一些措施。
（1）**随机接入系统ALOHA**：纯ALOHA、时域ALOHA协议

> 只要用户有数据要发送，就尽管让他们发送。
>
> 如果发送方检测到冲突，那么它可以等待一段随机长的时间后重发该帧。

（2）**CSMA（载波监听多路访问）**：1坚持、非坚持、P坚持。

> **非坚持式**：
>
> 经侦听，如果信道空闲，开始发送
> 如果信道忙，则等待一个随机分布的时间，然后重复步骤1
>
> **1-坚持式：**
>
> 经侦听，如信道空闲，开始发送
> 如信道忙，持续侦听，一旦空闲立即发送
> 如果发生冲突，等待一个随机分布的时间再重复步骤1
>
> **p-坚持式：**
>
> 经侦听，如信道空闲，那么以p的概率发送，以(1–p)的概率延迟一个时隙发送
>
> 如信道忙，持续侦听，一旦空闲重复步骤1
> 如果发送已推迟一个时间单元，再重复步骤1

（3）**CSMA/CD(碰撞检测)**：先听后发、边听边发、冲突停发、随机重发。适合**有线网（局域网）。**

> **存在争用期**，因为必须保证检测到冲突的时候数据还没发完，所以2*端到端的传播时间必须小于数据的发送延时。即可以确定最小帧长（以太网的最小帧长是64B）
>
> 随机重发的策略是**二进制指数退避算法**来防止再次冲突。

（4）**CSMA/CA（碰撞避免）**：适用于**无线网。**并不能完全避免，只能尽量的避免，在发送的过程中不检测冲突。

> CSMA/CA 采用二进制指数退避算法。
>
> 信道从忙变为空闲时，任何一个站要发送数据帧时，不仅都须等待一个时间间隔，而且还要进入争用窗口，并计算随机退避时间以便再次试图接入信道，因此降低了发生碰撞的概率。
>
> CSMA/CA 还使用预约信道、ACK 帧、RTS/CTS 帧等三种机制来实现碰撞避免：
>
> 预约信道。发送方在发送数据的同时向其他站点通知自己传输数据需要的时间长度，以便让其他站点在这段时间内不发送数据，从而避免碰撞。

> ACK 帧。所有站点在正确接收到发给自己的数据帧（除广播帧和组播帧）后，都需要向发送方发回一个ACK 帧，如果接收失败，那么不采取任何行动。发送方在发送完一个数据帧后，在规定的时间内如果未收到ACK 帧，那么认为发送失败，此时进行该数据帧的重发，直到收到ACK 帧或达到规定重发次数为止。
>
> RTS/CTS 帧。可选的碰撞避免机制，主要用于解决无线网中的"隐蔽站"问题。

3.9 局域网

局域网是指在一个**较小的地理范围**内，将各种**计算机、外部设备**和**数据库系统**等通过双绞线、同轴电缆等**连接介质**互相连接起来，组成**资源和信息**共享的**计算机互联网络。**

3.10 数据链路层在广域网中的协议

（1）**PPP协议**：按**字节**传输的，点到点的协议，不可靠，只检错丢弃，不纠错，不适用序号和确认机制。支持全双工。

（2）**HDLC（高级数据链路控制协议）**：面向**比特**，提供可靠传输（使用了编号和确认机制），用的是**0比特插入法**实现透明传输。

3.11 链路层设备（局域网中）

隔离冲突域，不隔离广播域

（1）**网桥**：一次只能转发一个帧。可以互联不同的物理层、不同的MAC 子层及不同速率的以太网，具有过滤帧及存储转发帧的功能。

（2）**交换机**：多端口的网桥，解决网桥一次只能转发一帧的不足。实现多个结点之间的并发传输。有直通式、存储转发式。

第四章网络层

4.1 IP地址与MAC地址的区别

（1）举个栗子，就像是你接收快递时所填的家庭地址和你的个人信息一样。IP地址就是你的住址，这个是可以变的，但是你的个人信息，也就是你的身份证号码是不会变的。
（2）**链路层**传输需要的是物理地址**MAC**，**网络**之间通过路由器转发分组需要的地址是**IP地址。**
（3）IP地址是**逻辑**的，MAC地址基于**物理设备。**

4.2 IPV4和IPV6的区别

（1）IPV4地址32位，IPV6地址128位。IPV6的出现是为了解决IPV4地址不够用等问题。

（2）IPV6提高安全性。身份认证和隐私权是IPV6的关键特性。

4.3 如何实现IPV4和IPV6的互联

（1）**双栈技术**（IPV4协议栈和IPV6协议栈）：数据链路层根据收到的IP数据报进行解析其头部第一个字段，即版本，版本位4就用IPV4协议栈去处理，版本位6就用IPV6协议栈去处理。
（2）**隧道技术**：实现IPV6的较完整运行。隧道技术是在IPv6网络与IPv4网络间的隧道入口处，由路由器将IPv6的数据分组封装到IPv4分组中。IPv4分组的源地址和目的地址分别是隧道入口和出口的IPv4地址。在隧道的出口处拆封IPv4分组并剥离出IPv6数据包。

4.4 虚电路与数据报比较

（1）**虚电路**需要建立连接，是有序的，能够保证数据传输的质量，但是灵活性不够强，结点出现问题所有经过此节点的虚电路都不能用了。

（2）**数据报交换**：无连接不可靠，但是灵活。

4.5 路由器的功能

1. 路由选择。指按照复杂的分布式算法，根据从各相邻路由器所得到的关于整个网络拓扑的变化情况，动态地改变所选择的路由。
2. 分组转发。指路由器根据转发表将用户的IP 数据报从合适的端口转发出去。

4.6 动态路由算法

1、距离-向量路由算法（例如RIP算法）

> 在距离 - 向量路由算法中，所有结点都**定期**地将它们的**整个路由选择表**传送给所有与之**直接相邻**的结点。
>
> 路由选择表包含：1.每条路径的目的地 2.路径的代价（距离）。
>
> **更新情况：**
>
> 1.新路由不存在则在本节点加入该路由。
>
> 2.发来的路由信息中有一条路由，该路由与当前使用的路由相比，有较短的距离（较小的代价）。此时新路由替换旧路由。

2、链路状态路由算法（例如OSPF算法）

> 链路状态路由算法要求**每个**参与该算法的**结点**都具有**完全的网络拓扑信息**。
>
> 它们执行下述两项任务。第一，**主动**测试所有**邻接结点的状态**。第二，**定期**地将链路状态**传播给所有其他结点**（或称路由结点）

4.7 因特网中的两大类路由选择协议

（1）域内路由选择协议IGP：一个自治系统内部所使用的路由选择协议

路由信息协议RIP（UDP，用的路由选择算法是距离向量，最多15跳）

开放最短路径优先OSPF（IP，用的路由选择算法是链路状态，其中用到了迪杰斯特拉最短路径求解算法）

（2）域间路由选择协议EGP：自治系统之间所使用的路由选择协议

BGP（TCP）

每个自治系统一个BGP发言人，与其他自治系统的BGP发言人建立TCP连接，交换BGP报文建立BGP会话，交换路由信息，找出较好的路由。

4.8 网络层转发分组的流程

> 1. 从数据报的首部提取**目的IP 地址**D, 得出**目的网络地址**N。
> 2. 若网络N 与此**路由器直接相连**，则把数据报**直接交付**给目的主机D, 这称为路由器的直接交付；否则是间接交付，执行步骤3) 。
> 3. 若路由表中有目的地址为D 的**特定主机路由**，则把数据报传送给路由表中所指明的下一跳路由器；否则执行步骤4)
> 4. 若路由表中有**到达网络N 的路由**，则把数据报传送给路由表指明的下一跳路由器；否则，执行步骤5) 。
> 5. 若路由表中有一个**默认路由**，则把数据报传送给路由表中所指明的默认路由器；否则，执行步骤6) 。
> 6. 报告转发**分组出错**。

4.9 有人认为：“ARP协议向网络层提供转换地址的服务，因此ARP应当属于数据链路层。”正确么？

不正确。 因为ARP本身是网络层的一部分，ARP协议为IP协议提供了转换地址的服务；数据链路层使用硬件地址而不使用IP地址，无需ARP协议数据链路层本身即可正常运行。因此ARP不属于数据链路层。

4.10 ARP协议

ARP 是解决同一个局域网上的主机或路由器的 **IP 地址和硬件地址**的映射问题。

4.11 在经过路由器转发的过程中IP数据报中的IP源地址、目的地址以及MAC源地址和目的地址会发生变化吗?

如果路由器不提供NAT，则IP地址是不会变化的。但是如果提供NAT的话，是会改变的。但是MAC源、目的地址是会变化的。

（1）如果**源IP地址**是**私有IP地址**，需要经过路由器的NAT的转换，源地址转换成**公有的IP地址**，但是**目的地址是不变的。**

（2）**MAC源、目的地址会一直变化**，根据所处的位置以及下一跳进行变化。

4.12 DHCP动态主机配置协议（应用层协议，UDP ）

用于给主机动态地分配IP 地址

> DHCP 的工作原理如下：使用**客户 / 服务器方式**。
>
> 需要IP 地址的**主机**在启动时就向DHCP 服务器**广播发送发现报文**，这时该主机就成为**DHCP 客户**。本地网络上所有主机都能收到此广播报文，但只有**DHCP 服务器才回答**此广播报文。DHCP 服务器先在其**数据库中查找**该计算机的**配置信息**。若**找到**，则**返回**找到的信息。若**找不到**，则从服务器的**IP 地址池**中取一个地址分配给该计算机。

4.13 .ICMP网际控制报文协议（IP层）

让主机或路由器报告差错和异常情况。

ICMP 差错报告报文和ICMP 询问报文。

ICMP 差错报告报文用于报告差错和异常情况。

（终点不可达、源点抑制、时间超时、参数问题、改变路由）

第五章传输层（提供端到端的服务）

5.1 传输层的两个协议

无连接不可靠但是很快速的**UDP**（面向报文）

有连接可靠但是管理麻烦的**TCP**（面向字节流）。

5.2 TCP和UDP首部格式

UDP：首部只有8B，4个字段（每个占2B），分别为源端口号、目的端口号、长度（是UDP整个的长度）、校验位（校验UDP中的首部+数据）

TCP：20B的首部，包含很多东西。

5.3 如果IP可以可靠传输是不是就不需要UDP了，直接跟应用层联系?

不可以。IP只能确定主机，但是真正通信的是两台主机中的进程，所以我们需要确定具体的进程也就是需要源端口号和目的端口号，这需要UDP提供。

5.4 TCP的特点

传输控制协议 TCP（Transmission Control Protocol）

1. **有连接**的
2. 每一条 TCP连接只能是**点对点**的（一对一）
3. 提供**可靠**交付
4. **全双工**通信

5. 有**流量**控制，**拥塞**控制，面向**字节流**

5.5 UDP特点

用户数据报协议 UDP（User Datagram Protocol）

1. **无连接**的
2. **尽最大可能交付**
3. 面向**报文**，对于应用程序传下来的报文不合并也不拆分
4. **没有拥塞控制**
5. 支持**一对一、一对多、多对一和多对多**的交互通信
6. 只是添加 UDP 首部，且**首部开销小**

5.6 TCP和UDP的区别

1. TCP提供**面向连接**的传输；UDP提供**无连接**的传输

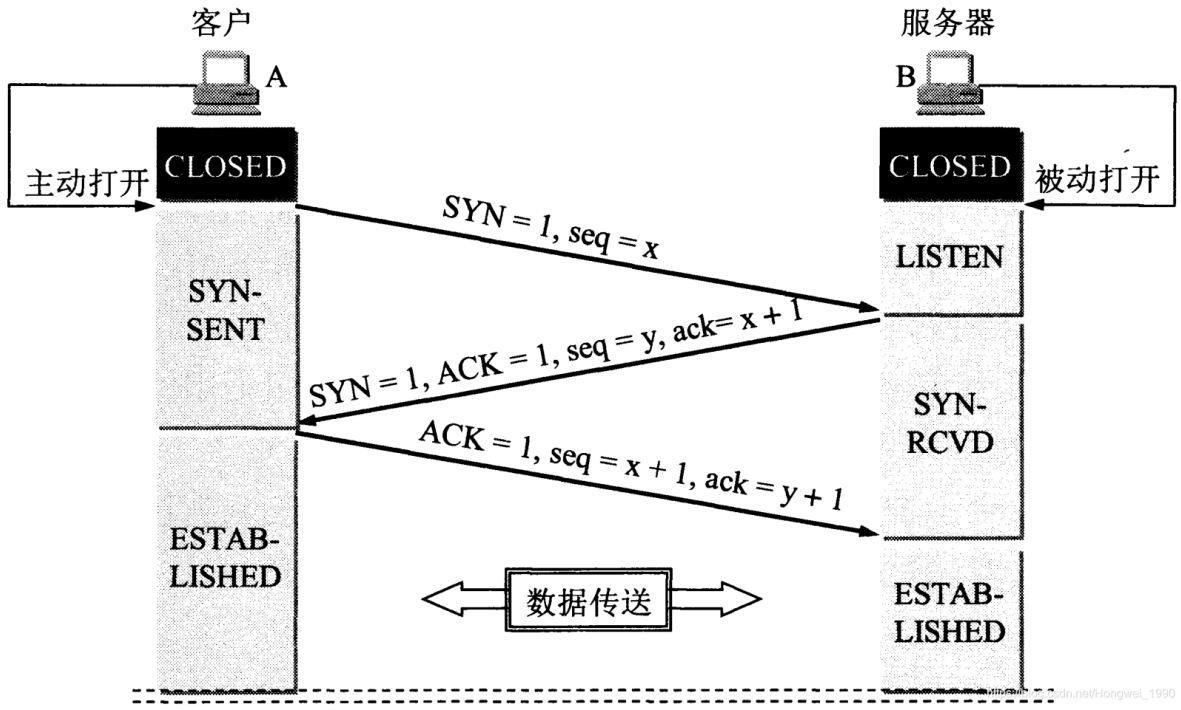
2. TCP提供**可靠的传输**（有序，无差错，不丢失，不重复）；UDP提供**不可靠的传输。**

3. TCP面向**字节流**的传输，因此它能将信息分割成组，并在接收端将其重组；UDP是面向**数据报**的传输，没有分组开销。

4. TCP提供**拥塞控制和流量控制**机制；UDP**不提供拥塞控制和流量控制**机制。

> 5. TCP只能是**点对点**的（一对一）。UDP支持**一对一、一对多、多对一和多对多**的交互通信。

5.7 TCP建立连接（三次握手）



连接建立前，服务器进程处于LISTEN（收听）状态，等待客户的连接请求。

第1步，客户机的TCP首先向服务器的TCP发出 连接请求报文段，这时首部中的 同步位SYN=1，同时选择一个初始 序列seq=x。TCP规定， SYN报文段 （即 SYN=1 的报文段） 不能携带数据，但要 消耗掉一个序列。这时，TCP客户进程进入 SYN-SENT （同步已发送）状态。

第2步，服务器的TCP收到 连接请求报文段 **后，如果同意建立连接，则向客户机发送确认。**在 确认报文段 中应把 SYN位 和 ACK位 都置 1，确认号是 ack=x+1，同时也为自己选择一个初始 序列seq=y。请注意，这个报文段也 不能携带数据，但同样 要消耗掉一个序号。这时TCP服务器进程进入 SYN-RCVD （同步收到）状态。

第3步，当客户机收到确认报文段后，还要向服务器给出确认，并为该TCP连接分配缓存和变量。 确认报文段的ACK=1，确认号ack=y+1，而自己的序列seq=x+1。TCP的标准规定，ACK报文段可以携带数据。但如果不携带数据则不消耗序号，在这种情况下，下一个数据报文段的序列仍是seq=x+1。这时，TCP连接已经建立，A进入ESTABLISHED（已建立连接）状态。

当服务器收到客户机的确认后，也进入ESTABLISHED状态。

5.7.1 TCP三次握手建立连接，两次不行吗?

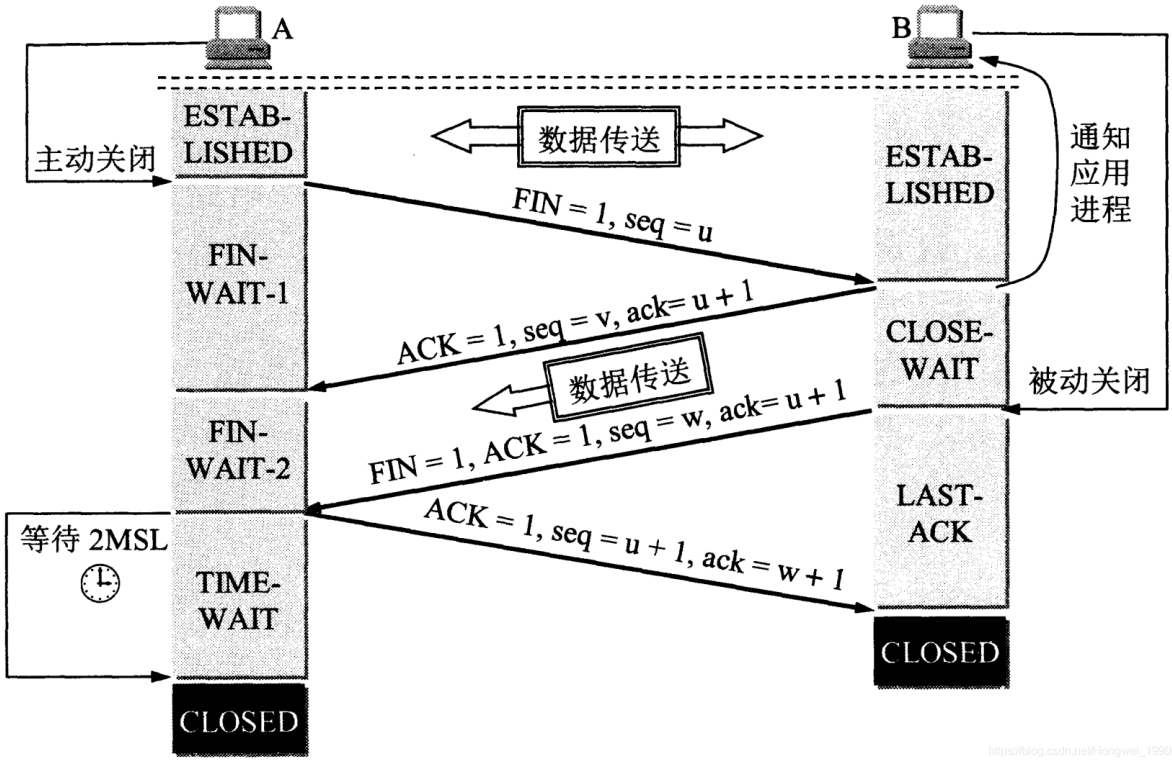
不可以。 如果存在已失效的连接请求报文重新传到了服务器端，则服务器会响应然后向客户端发送ACK，此时认为已经建立连接，之后服务器便开始等待客户机传送数据。但是客户机并未响应这个ACK，所以服务器会一直等待下去，造成资源的浪费。

如果有最后一次确认，则在服务器收到滞留的请求并回复之后，客户机不会对服务器的回复做出回复，进而不会建立连接。

5.7.2 四次呢?

四次太多了，三次就可以双方都确认对方已经做好了接收数据以及发送数据的准备。

5.8 TCP连接释放（四次挥手）



数据传输结束后，通信的双方都可释放连接。

第1步，客户机打算关闭时，先向其TCP发出连接释放报文段，并停止再发送数据，主动关闭TCP连接。 该报文段的终止位FIN置1，其序列seq=u，它等于前面已传送过的数据的最后一个字节的序列加1。这时客户机进入FIN-WAIT-1（终止等待1）状态。请注意，TCP规定，FIN报文段即使不携带数据，它也要消耗一个序列。

第2步，服务器收到连接释放报文，发出确认报文，ACK=1，ack=u+1，并且带上自己的序列号seq=v，此时，服务端就进入了CLOSE-WAIT（关闭等待）状态。此时，从客户机到服务器这个方向的连接就释放了，TCP处于半关闭状态。但服务器若发送数据，客户机仍要接收，即从服务器到客户机这个方向的连接并未关闭。

客户端收到服务器的确认请求后，此时，客户端就进入FIN-WAIT-2（终止等待2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最后的数据）

第3步，服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，`FIN=1，重复上次已发送的确认号ack=u+1`，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的`序列号为seq=w`，此时，服务器就进入了`LAST-ACK（最后确认）状态`，等待客户端的确认。

第4步，客户端收到服务器的连接释放报文后，必须发出确认，`ACK=1，ack=w+1`，而自己的`序列号是seq=u+1`，此时，TCP连接还没有释放，必须经过`2*MSL（最长报文段寿命）`的时间后，客户机才进入`CLOSED（连接关闭）状态`。

服务器只要收到了客户端发出的确认，立即进入CLOSED状态。同样，撤销TCB后，就结束了这次的TCP连接。

5.8.1 TCP四次挥手（握手）释放连接，为啥需要四次而不是三次?

（1）是因为可能有**部分数据需要处理。**
（2）**建立连接**时，被动方服务器端结束CLOSED阶段进入"握手"阶段并不需要任何准备，可以**直接返回SYN和ACK报文**，开始建立连接。
（3）**释放连接**时，被动方服务器，突然收到主动方客户端释放连接的请求时并不能立即释放连接，因为还有必要的数据需要处理，所以服务器先返回ACK确认收到报文，经过CLOSE-WAIT阶段准备好释放连接之后，才能返回FIN释放连接报文。但是连接释放请求的时候，**第二次发送的只是ACK，并没有FIN**，是因为服务器可能还需要传送一些数据，所以要等待**传送完毕之后**才能再次给客户端**发送FIN=1**的连接释放报文。

5.8.2 TCP四次挥手为啥第四次客户机发送ACK之后要等待2MSL之后才能真的关闭?

1）**MSL是最大报文段生存周期。**客户机发送ACK后如果MSL后服务器端没有接受到，就会再次发送一个请求释放报文，这一过程总的到达客户端的时间最多就是2MSL，所以一旦ACK出现丢失，那么2MSL内客户机一定可以得到消息，然后重新发送ACK，再次重新计时。
2）如果客户机发送ACK之后就关闭了，那么如果出现**ACK丢失的现象**，即使服务器端再次重新发送请求释放报文，它也不会理会，就会导致服务器端无法结束。
3）还可以防止**已失效的连接请求报文。**

5.9 重传的两种方法

1）**超时重传**：根据RTT均值设置一个超时重传的时间。但是这种方法不好的点在于慢。但是这种情况预示着此时拥塞很严重了。
2）**冗余ACK**：利用TCP接受数据的特点，当前未接受的数据k而言，如果k后的数据到达，每次都会产生一个ACK的确认序号为k，3个这样的就表示需要为k的报文段丢失需要重发。这样的好处是快，而且此时没有那么的拥塞。

5.10 TCP实现可靠传输的方式

5.10.1 停止等待协议

每发送一个分组就停止发送，等待对方的确认。在收到确认后再发送下一个分组。

5.10.2 流量控制

让发送方的发送速率不要太快，要让接收方来得及接收。

5.10.3 拥塞控制

防止过多的数据注入到网络中，这样可以使网络中的路由器或链路不致过载。

四种算法：

1、**慢开始**：由小到大逐渐增大拥塞窗口，每经过一个传输轮次，拥塞窗口加倍
2、**拥塞避免**（网络阻塞）：每经过一个传输轮次，拥塞窗口加1
3、**快重传**：当发送方连续收到三个重复的ACK 报文时，直接重传对方尚未收到的报文段，而不必等待那个报文段设置的重传计时器超时。

4、**快恢复**（三个冗余ACK）：发送端收到连续三个冗余ACK时可以执行快恢复，调整限制到原拥塞窗口的1/2，然后开始执行拥塞避免算法。

第六章

6.1 应用层传输的方式

C/S方式、P2P点对点方式（平等的，每个主机既可以作为Client也可以作为Service）

6.2 域名解析协议DNS

域名解析是指把**域名映射成为IP 地址**或把**IP 地址映射成域名**的过程。前者称为**正向解析**，后者称为**反向解析。**当客户端需要域名解析时，通过本机的**DNS 客户端**构造一个**DNS 请求报文**，以**UDP 数据报**方式发往**本地域名服务器。**域名解析有两种方式：**递归查询**和**递归与迭代相结合**的查询。

6.3 FTP（文件传输协议）

FTP提供**交互式**的访问，允许客户指明**文件的类型与格式**，并允许文件具有**存取权限。**它**屏蔽**了各计算机系统的**细节**，因而**适合**于在**异构网络**中的任意计算机之间**传送文件。**

FTP 采用**客户 / 服务器**的工作方式，它使用**TCP 可靠**的传输服务。**FTP 的服务器进程**由两大部分组成：一个**主进程**，负责**接收新的请求**；另外有若干**从属进程**，负责**处理单个请求。**

其工作步骤如下：（1）打开**熟知端口21** (控制端口），使客户进程能够连接上。（2）等待客户进程发连接请求。（3）启动从属进程来处理客户进程发来的请求。主进程与从属进程并发执行，从属进程对客户进程的请求处理完毕后即终止。（4）回到等待状态，继续接收其他客户进程的请求。

6.4 SMTP以及POP3：（基于TCP）

（1）SMTP：简单邮件传输协议，使用**客户 / 服务器**方式，SMTP 用的是**TCP 连接**，端口号为**25** 。在用户代理--->邮件服务器，以及邮件服务器-->邮件服务器之间的邮件的传输。（Push推的方式）

（2）POP3：邮局协议，客户 / 服务器的工作方式，在传输层使用TCP, 端口号为110 。是传输从邮件服务器到用户代理。（Pull拉的方式）POP 有两种工作方式："下载并保留"和"下载并删除 " 。

6.5 HTTP(基于TCP有连接)

超文本传输协议HTTP（HyperText Transfer Protocol），HTTP是**面向事务**的**应用层协议。**定义了浏览器怎样向万维网服务器请求万维网文档，以及服务器怎样把文档传送给浏览器。HTTP是万维网上能够可靠地交换文件的重要基础。

6.6 HTTP中GET和POST的区别

1、从**原理性**看：

GET用于信息获取，而且应该是安全的；POST请求表示可能修改服务器上资源的请求。

2、从**表面**上看：

GET请求的数据会附在URL后面，POST的数据放在HTTP包体，POST安全性比GET安全性高。

常见题

1. 说下浏览器请求一个网址的过程?

（1）首先是**解析域名**，**DNS**进行，获得**IP地址。**然后端口号发送端随机分配，服务器是固定的80端口。
（2）知道**IP和端口号**之后就可以建立3次握手**建立TCP连接。**
（3）发送**HTTP请求报文。**
（4）服务器处理浏览器发来的请求，**返回**要显示的页面**html信息。**

(5) 传输结束，4次握手**释放连接。**

(6) **显示页面。**