面试常见经典问题打卡

day1:TCP三次握手和四次握手：

1.TCP 协议的特点

TCP是在不可靠的IP层之上实现的可靠的数据传输协议，它主要解决传输的可靠、有序、无丢失和不重复问题。TCP 是TCP/IP 体系中非常复杂的一个协议，主要特点如下：
1) TCP 是面向连接的传输层协议。
2) 每条TCP 连接只能有两个端点，每条TCP 连接只能是点对点的（一对一）。
3) TCP 提供可靠的交付服务，保证传送的数据无差错、不丢失、不重复且有序。

> 这里是一个提问点：如何保证数据无差错、不丢失、不重复且有序的？有哪些机制来保证？
> 答：TCP 使用了校验、序号、确认和重传等机制来达到这一目的。

4) TCP 提供全双工通信，允许通信双方的应用进程在任何时候都能发送数据，为此TCP 连接的两端都设有发送缓存和接收缓存，用来临时存放双向通信的数据。

> 这里是一个提问点：为什么需要设置缓存，缓存的作用？
>
> 答：发送缓存用来暂时存放以下数据：1.发送应用程序传送给发送方TCP 准备发送的数据；2.TCP 已发送但尚未收到确认的数据。
>
> 接收缓存用来暂时存放以下数据：1.按序到达但尚未被接收应用程序读取的数据；2.不按序到达的数据。

5) TCP是面向字节流的，虽然应用程序和TCP的交互是一次一个数据块（大小不等），但TCP把应用程序交下来的数据仅视为一连串的无结构的字节流。

> 一个字节占一个序号，每个报文段用第一个字节的序号来标识,例如，一报文段的序号字段值是301, 而携带的数据共有l00B, 表明本报文段的数据的最后一个字节的序号是400, 因此下一个报文段的数据序号应从401开始，也就是期望的下一个序号（确认号）。

2.TCP报文段格式

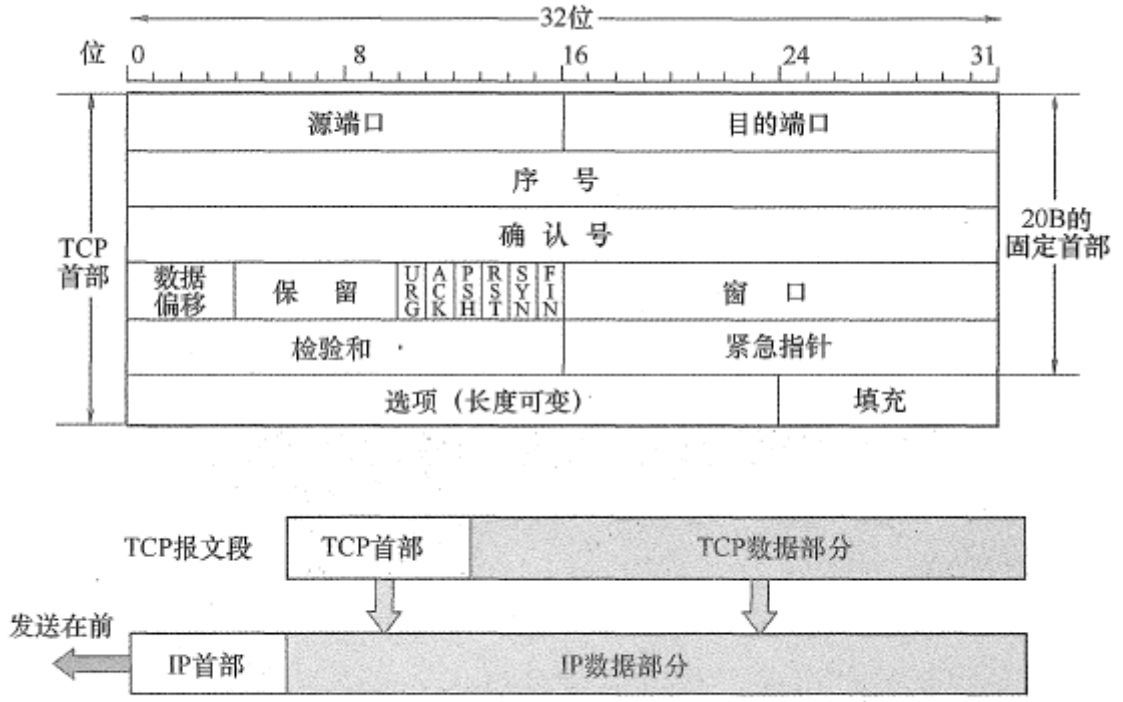


图 5.6 TCP 报文段

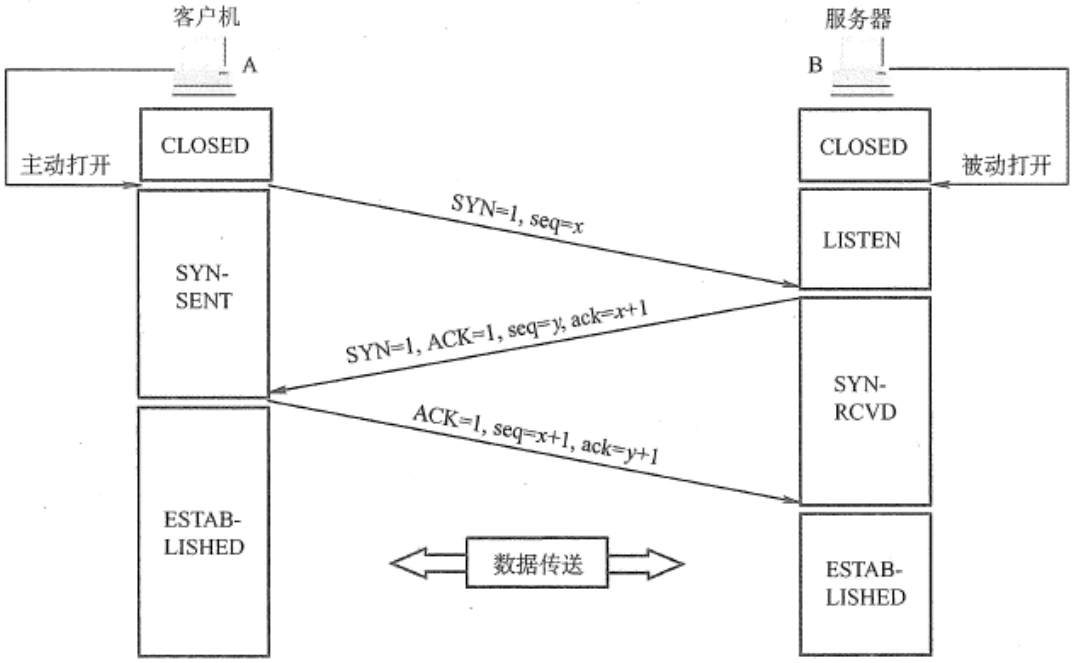
> 部分字段解释：
>
> 1) 序号字段（就是seq）：序号字段的值指的是本报文段所发送的数据的第一个字节的序号。
>
> 2) 确认号字段（就是ack）：是期望收到对方的下一个报文段的数据的第一个字节的序号。若确认号为N, 则表明到序号N-1为止的所有数据都已正确收到。（累积确认）
>
> 3) 确认位ACK：只有当ACK=1时确认号字段才有效。当ACK=0时，确认号无效。TCP 规定，在连接建立后所有传送的报文段都必须把ACK置1。
>
> 4) 同步位SYN。同步SYN=1表示这是一个连接请求或连接接收报文。当SYN= 1, ACK=0 时，表明这是一个连接请求报文，对方若同意建立连接，则在响应报文中使用SYN= 1, ACK=1。即SYN=1表示这是一个连接请求或连接接收报文。
>
> 5) 终止位FIN (Finish) 。用来释放一个连接。FIN=1表明此报文段的发送方的数据已发送完毕了并要求释放传输连接。

3.TCP连接管理

TCP 是面向连接的协议，因此每个TCP 连接都有三个阶段：连接建立、数据传送和连接释放。TCP 连接的管理就是使运输连接的建立和释放都能正常进行。
在TCP 连接建立的过程中，要解决以下三个问题：
1) 要使每一方都能够确知对方的存在。
2) 要允许双方协商一些参数（如最大窗口值、是否使用窗口扩大选项、时间戳选项及服务质量等）。
3) 能够对运输实体资源（如缓存大小、连接表中的项目等）进行分配。

每条TCP 连接唯一地被通信两端的两个端点（即两个套接字）确定。`端口拼接到IP地址即为套接字`

4.三次握手建立连接



1) 第一次握手：客户机的TCP首先向服务器的TCP发送一个连接请求报文段。这个特殊的报文段中不含应用层数据，其首部中的SYN标志位被置为1。另外，客户机会随机选择一个起始序号seq = x（连接请求报文不携带数据，但要消耗一个序号）。
2) 第二次握手：服务器的TCP 收到连接请求报文段后，如同意建立连接，就向客户机发回确认，并为该TCP连接分配TCP缓存和变量。在确认报文段中，SYN 和ACK 位都被置为1, 确认号字段的值为x+1,并且服务器随机产生起始序号seq= y(确认报文不携带数据，但也要消耗一个序号）。确认报文段同样不包含应用层数据。
3) 第三次握手：当客户机收到确认报文段后，还要向服务器给出确认，并且也要给该连接分配缓存和变量。这个报文段的ACK 标志位被置1, 序号字段为x+1, 确认号字段ack=y+1。该报文段可以携带数据，若不携带数据则不消耗序号 `http中的tcp连接的第三次握手的报文段中就捎带了客户对万维网文档的请求`。

成功进行以上三步后，就建立了TCP 连接，接下来就可以传送应用层数据。TCP 提供的是全双工通信，因此通信双方的应用进程在任何时候都能发送数据。

【总结】：
1) SYN = 1,ACK = 0,seq = x;
2) SYN = 1,ACK = 1,seq = y,ack = x+1;
3) SYN = 0,ACK = 1,seq = x+1,ack=y+1。

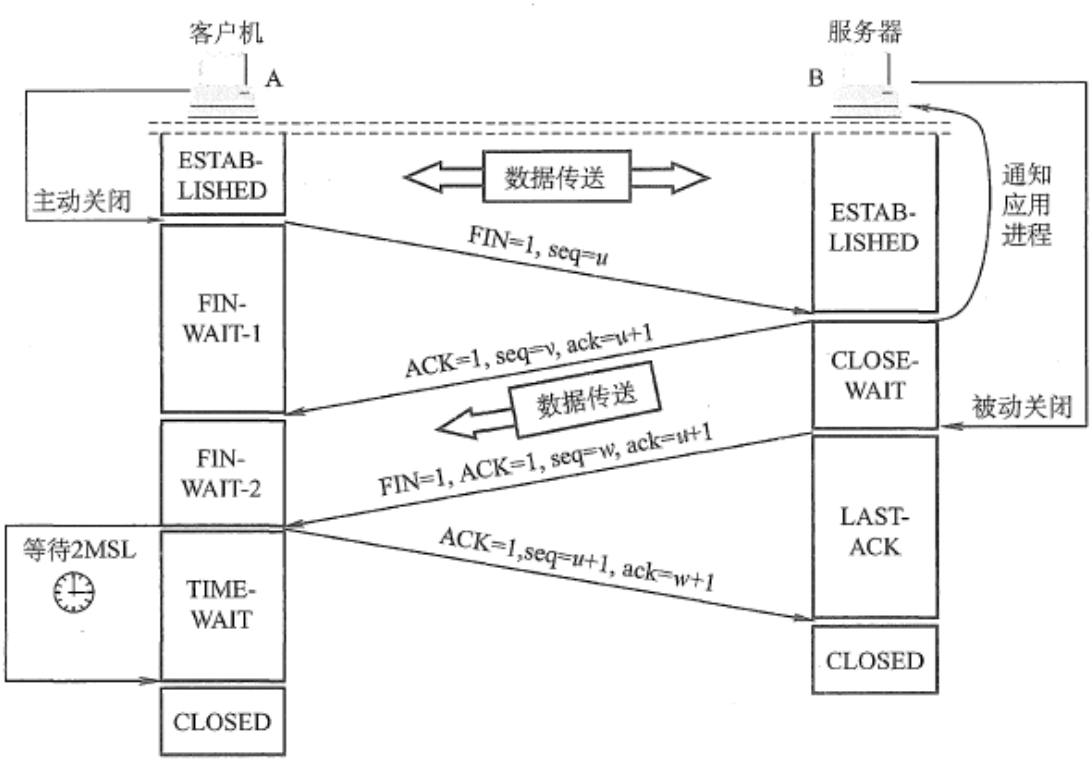
> 【拓展问题1】：什么是SYN洪泛攻击？（三次握手机制有什么问题？）答：由于服务器端的资源是在完成第二次握手时分配的，而客户端的资源是在完成第三次握手时分配的，攻击者发送TCP的SYN报文段，SYN是TCP三次握手中的第一个数据包，而当服务器返回ACK后，该攻击者就不对其进行再确认，那这个TCP连接就处于挂起状态，也就是所谓的半连接状态，服务器收不到再确认的话，还会重复发送ACK给攻击者。这样更加会浪费服务器的资源。攻击者就对服务器发送非常大量的这种TCP连接，由于每一个都没法完成三次握手，所以在服务器上，这些TCP连接会因为挂起状态而消耗CPU和内存，最后服务器可能死机，就无法为正常用户提供服务了。

> 【拓展问题2】：如果已经建立了连接，但是客户端突然出现故障了怎么办？
> TCP还设有一个保活计时器，显然，客户端如果出现故障，服务器不能一直等下去，白白浪费资源。服务器每收到一次客户端的请求后都会重新复位这个计时器，时间通常是设置为2小时，若两小时还没有收到客户端的任何数据，服务器就会发送一个探测报文段，以后每隔75秒钟发送一次。若一连发送10个探测报文仍然没反应，服务器就认为客户端出了故障，接着就关闭连接。

> 【拓展问题3】：为什么不采用“两次握手”建立连接呢？
> 答：这主要是为了防止两次握手情况下已失效的连接请求报文段突然又传送到服务器而产生错误。考虑下面这种情况。客户A 向服务器B 发出TCP 连接请求，第一个连接请求报文在网络的某个结点长时间滞留， A 超时后认为报文丢失，于是再重传一次连接请求， B 收到后建立连接。数

> 据传输完毕后双方断开连接。而此时，前一个滞留在网络中的连接请求到达服务器B, 而B 认为A 又发来连接请求，此时若使用“三次握手”，则B 向A 返回确认报文段，由于是一个失效的请求，因此A 不予理睬，建立连接失败。若采用的是“两次握手”，则这种情况下B 认为传输连接已经建立，并一直等待A 传输数据，而A 此时并无连接请求，因此不予理睬，这样就造成了B的资源白白浪费。

5.四次握手释放连接



1) 第一次握手：客户机打算关闭连接时，向其TCP发送一个连接释放报文段，并停止发送数据，主动关闭TCP 连接，该报文段的FIN 标志位被置1, seq= u, 它等于前面已传送过的数据的最后一个字节的序号加1 (FIN 报文段即使不携带数据，也要消耗一个序号）。TCP是全双工的，即可以想象为一条TCP 连接上有两条数据通路。发送FIN 报文时，发送FIN 的一端不能再发送数据，即关闭了其中一条数据通路，但对方还可以发送数据。
2) 第二次握手：服务器收到连接释放报文段后即发出确认，确认号是ack = u + 1, 而这个报文段自己的序号是v, 等千它前面已传送过的数据的最后一个字节的序号加1 。此时，从客户机到服务器这个方向的连接就释放了，TCP连接处千半关闭状态。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。
3) 第三次握手：若服务器已经没有要向客户机发送的数据，就通知TCP释放连接，此时其发出FIN=1的连接释放报文段。
4) 第四次握手：客户机收到连接释放报文段后，必须发出确认。在确认报文段中，ACK字段被置为1, 确认号ack= w + 1, 序号seq= u + 1 。此时TCP连接还未释放，必须经过时间等待计时器设置的时间2MSL（最长报文段寿命）后，A才进入连接关闭状态。

【总结】：
1) FIN = 1,seq = u;
2) ACK = 1,seq = v,ack = u+1;
3) FIN = 1,ACK = 1,seq = w,ack =u+1;(确认第一次的u)
4) ACK = 1,seq = u+1,ack = w+1。

> 【拓展问题1】为什么连接的时候是三次握手，关闭的时候却是四次握手？
> 答：因为当Server端收到Client端的SYN连接请求报文后，可以直接发送SYN+ACK报文。其中ACK报文是用来应答的，SYN报文是用来同步的。但是关闭连接时，当Server端收到FIN报文时，很可能并不会立即关闭SOCKET，所以只能先回复一个ACK报文，告诉Client端，"你发的FIN报文我收到了"。只有等到我Server端所有的报文都发送完了，我才能发送FIN报文，因此不能一起发送。故需要四次握手。

> 【拓展问题2】为什么TIME_WAIT状态需要经过2MSL(最大报文段生存时间)才能返回到CLOSE状态？
> 答：1)虽然按道理，四个报文都发送完毕，我们可以直接进入CLOSE状态了，但是网络是不可靠的，有可能最后一个ACK丢失。所以TIME_WAIT状态就是用来重发可能丢失的ACK报文。在Client发送出最后的ACK回复，但该ACK可能丢失。Server如果没有收到ACK，将不断重复发送FIN片段。所以Client不能立即关闭，它必须确认Server接收到了该ACK。Client会在发送出ACK之后进入到TIME_WAIT状态。Client会设置一个计时器，等待2MSL的时间。如果在该时间内再次收到FIN，那么Client会重发ACK并再次等待2MSL。所谓的2MSL是两倍的MSL(Maximum Segment Lifetime)。MSL指一个片段在网络中最大的存活时间，2MSL就是一个发送和一个回复所需的最大时间。如果直到2MSL，Client都没有再次收到FIN，那么Client推断ACK已经被成功接收，则结束TCP连接。2)防止出现“已失效的连接请求报文段”（和上面的为啥不用二次握手类似）。A 在发送最后一个确认报文段后，再经过2MSL可保证本连接持续的时间内所产生的所有报文段从网络中消失。

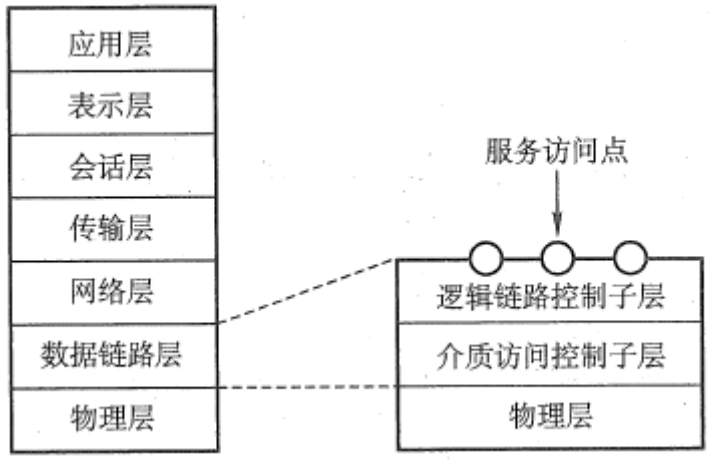
---

day2:介质访问控制

「正文开始：」

1.介质访问控制综述

局域网的数据链路层分为逻辑链路层LLC和介质访问控制MAC两个子层。逻辑链路控制（Logical Link Control或简称LLC）是局域网中数据链路层的上层部分，IEEE 802.2中定义了逻辑链路控制协议。用户的数据链路服务通过LLC子层为网络层提供统一的接口。在LLC子层下面是MAC子层。MAC(medium access control)属于LLC（Logical Link Control）下的一个子层，提供介质访问控控制的功能。模型图如下：



1. 为什么需要介质访问控制？ 因为局域网是一种广播式的网络（广域网是一种点对点的网络），所有联网计算机都共享一个公共信道，所以，需要一种方法能有效地分配传输介质的使用权，使得两对结点之间的通信不会发生相互干扰的情况，这种功能就叫介质访问控制。
2. 介质访问控制的分类？ 常见的介质访问控制方法有信道划分介质访问控制、随机访问介质访问控制和轮询访问介质访问控制。其中前者是静态划分信道的方法，而后两者是动态分配信道的方法。

2.信道划分介质访问控制

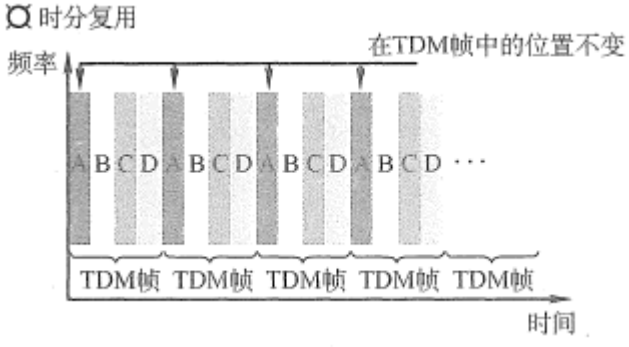
信道划分介质访问控制将使用介质的每个设备与来自同一通信信道上的其他设备的通信隔离开来，把时域和频域资源合理地分配给网络上的设备。信道划分的实质就是通过分时、分频、分码等方法把原来的一条广播信道，逻辑上分为几条用于两个结点之间通信的互不干扰的子信道，实际上就是把广播信道转变为点对点信道。 信道划分介质访问控制分为以下4 种：

频分多路复用（Frequency division multiplexing FDM）

- 频分多路复用是一种将多路基带信号调制到不同频率载波上，再叠加形成一个复合信号的多路复用技术。
- 每个子信道分配的带宽可不相同，但它们的总和必须不超过信道的总带宽。在实际应用中，为了防止子信道之间的干扰，相邻信道之间需要加入“保护频带”。
- 频分多路复用的优点在于充分利用了传输介质的带宽，系统效率较高；由于技术比较成熟，实现也较容易。缺点在于无法灵活地适应站点数及其通信量的变化。

时分多路复用（Time division multiplexing TDM）

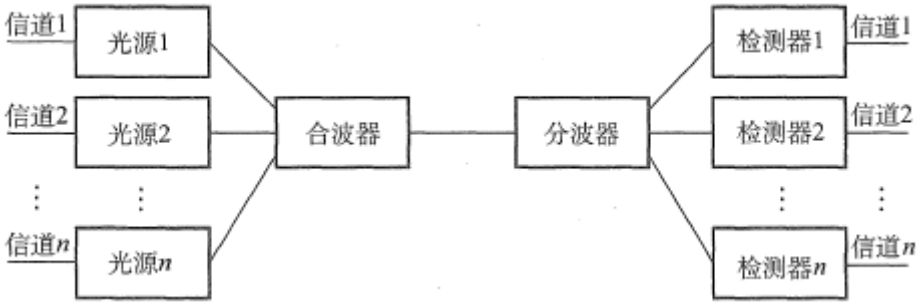
- 时分多路复用是将一条物理信道按时间分成若干时间片，轮流地分配给多个信号使用。每个时间片由复用的一个信号占用。
- 就某个时刻来看，时分多路复用信道上传送的仅是某一对设备之间的信号；就某段时间而言，传送的是按时间分割的多路复用信号。
- 但由于计算机数据的突发性，一个用户对已经分配到的子信道的利用率一般不高。所以对TDM进行改进，有了统计时分多路复用（STDM），它采用STDM 帧， STDM 帧并不固定分配时隙，而按需动态地分配时隙，当终端有数据要传送时，才会分配到时间片，因此可以提高线路的利用率。



波分多路复用（Wavelength division multiplexing　WDM）

- 波分多路复用即光的频分多路复用，它在一根光纤中传输多种不同波长（频率）的光信号，由于波长（频率）不同，各路光信号互不干扰，最后再用波长分解复用器将各路波长分解出来。由于光波

处于频谱的高频段，有很高的带宽，因而可以实现多路的波分复用.



码分多路复用(Code division multiplexing CDM)

- 码分多路复用是采用不同的编码来区分各路原始信号的一种复用方式。与FDM 和TDM 不同，它既共享信道的频率，又共享时间。
- 码分多址(Code Division Multiple Access, CDMA) 是码分复用的一种方式，其原理是每比特时间被分成m个更短的时间槽，称为码片(Chip), 通常情况下每比特有64 或128 个码片。每个站点被指定一个唯一的m 位代码或码片序列。发送1 时，站点发送码片序列；发送0 时，站点发送码片序列的反码。当两个或多个站点同时发送时，各路数据在信道中线性相加。为从信道中分离出各路信号，要求各个站点的码片序列相互正交。简单理解就是， A 站向C 站发出的信号用一个向量来表示， B 站向C 站发出的信号用另一个向量来表示，两个向量要求相互正交。向量中的分量，就是所谓的码片。(相关计算细节感兴趣的可自行百度)
- 码分多路复用技术具有频谱利用率高、抗干扰能力强、保密性强、语音质量好等优点，还可以减少投资和降低运行成本，主要用千无线通信系统，特别是移动通信系统。

> “
>
> 【一个生动形象的例子帮忙理解和总结】假设A 站要向C 站运输黄豆， B 站要向C 站运输绿豆，A 与C 、B 与C 之间有一条公共的道路，可以类比为广播信道。 在频分复用方式下，公共道路被划分为两个车道，分别提供给A 到C 的车和B 到C 的车行走，两类车可以同时行走，但只分到了公共车道的一半，因此频分复用（波分复用也一样）共享时间而不共享空间。 在时分复用方式下，先让A 到C 的车走一趟，再让B 到C 的车走一趟，两类车交替地占用公共车道。公共车道没有划分，因此两车共享了空间，但不共享时间。 码分复用与另外两种信道划分方式大为不同，在码分复用情况下，黄豆与绿豆放在同一辆车上运送，到达C 后，由C 站负责把车上的黄豆和绿豆分开。因此，黄豆和绿豆的运送，在码分复用的情况下，既共享了空间，也共享了时间。
>
> ”

3.随机访问介质访问控制

在随机访问协议中，不采用集中控制方式解决发送信息的次序问题，所有用户能根据自己的意愿随机地发送信息，占用信道全部速率。在总线形网络中，当有两个或多个用户同时发送信息时，就会产生帧的冲突（碰撞），导致所有冲突用户的发送均以失败告终。为了解决随机接入发生的碰撞，每个用户需要按照一定的规则反复地重传它的帧，直到该帧无碰撞地通过。这些规则就是随机访问介质访问控制协议，常用的协议有ALOHA 协议、CSMA协议、CSMA/CD 协议和CSMA/CA 协议等，它们的核心思想都是：胜利者通过争用获得信道，从而获得信息的发送权。因此，随机访问介质访问控制协议又称争用型协议。随机介质访问控制实质上是一种将广播信道转化为点到点信道的行为。

ALOHA协议

- 纯ALOHA 基本思想: 1.用户有数据要发送时，就让它们发送 2.然后监听信道是否产生冲突，若产生冲突，则等待一段随机的时间重发
- 分槽(时隙)ALOHA 基本思想: 1.把时间分成离散的间隔，每个间隔对应于发送一帧所需时间 2.每个站点只能等到下一个时槽开始时才允许发送 3.其他过程与纯ALOHA相同

- 分槽(时隙)ALOHA网络的吞吐量比纯ALOHA协议大了一倍。

CSMA协议（Carrier sense multiple access）

- 时隙ALOHA 系统的效率虽然是纯ALOHA 系统的两倍，但每个站点都是随心所欲地发送数据的，即使其他站点正在发送也照发不误，因此发送碰撞的概率很大。若每个站点在发送前都先侦听一下共用信道，发现信道空闲后再发送，则就会大大降低冲突的可能，从而提高信道的利用率，载波侦听多路访问协议依据的正是这一思想。CSMA 协议是在ALOHA 协议基础上提出的一种改进协议，它与ALOHA协议的主要区别是多了一个载波侦听装置。（先听后发）

表 3.1　三种不同类型的 CSMA 协议比较

信道状态	1-坚持	非坚持	p-坚持
空闲	立即发送数据	立即发送数据	以概率 p 发送数据，以概率 $1-p$ 推迟到下一个时隙
忙	继续坚持侦听	放弃侦听，等待一个随机的时间后再侦听	持续侦听，直至信道空闲

CSMA/CD 协议

载波侦听多路访问 / 碰撞检测(Carrier Sense Multiple Access with Collision Detection, CSMA/CD)(这些单词大家背一背没有坏处)协议是CSMA 协议的改进方案，适用于总线形网络或半双工网络环境。”载波帧听”就是发送前先侦听，即每个站在发送数据之前先要检测一下总线上是否有其他站点正在发送数据，若有则暂时不发送数据，等待信道变为空闲时再发送。”碰撞检测”就是边发送边侦听，即适配器边发送数据边检测信道上信号电压的变化情况，以便判断自己在发送数据时其他站点是否也在发送数据。引入原因:当两个帧发生冲突时，两个被损坏帧继续传送毫无意义，而且信道无法被其他站点使用，对于有限的信道来讲，造成很大的浪费。如果站点边发送边监听，并在监听到冲突之后立即停止发送，可以提高信道的利用率。

> “
>
> CSMA/CD 的【工作流程】可简单概括为“先听后发，边听边发（区别于CSMA协议），冲突停发，随机重发”。1) 适配器从其父结点获得一个网络层数据报，准备一个以太网帧，并把该帧放到适配器缓 冲区中。2) 如果适配器侦听到信道空闲，那么它开始传输该帧。如果适配器侦听到信道忙，那么它 将等待直至侦听到没有信号能量，然后开始传输该帧。3) 在传输过程中，适配器检测来自其他适配器的信号能量。如果这个适配器传输了整个帧，而没有检测到来自其他适配器的信号能量，那么这个适配器完成该帧的传输。否则，适配器就须停止传输它的帧，取而代之传输一个48 比特的拥塞信号。4) 在中止（即传输拥塞信号）后，适配器采用截断二进制指数退避算法等待一段随机时间 后返回到步骤2)。
>
> ”

> “
>
> 【何为截断二进制指数退避算法】1) 确定基本退避时间,一般取两倍的总线端到端传播时延(即争用期）。2) 定义参数k, 它等于重传次数，但k不超过10, 即k=min【重传次数， 10】。当重传次数不超过10 时，k等于重传次数；当重传次数大于10时，k就不再增大而一直等于10(这个条件往往容易忽略，请注意）。3) 从离散的整数集合【0, 1, …, 2^k-1】 中随机取出一个数r, 重传所需要退避的时间就是r倍的基本退避时间。4) 当重传达16 次仍不能成功时，说明网络太拥挤，认为此帧永远无法正确发出，抛弃此帧并向高层报告出错（这个条件也容易忽略，请注意）。使用二进制指数退避算法可使重传需要推迟的平均时间随重传次数的增大而增大（这也称动态退避），因而能降低发生碰撞的概率，有利于整个系统的稳定。
>
> ”

总线的传播时延对 CSMA/CD 的影响很大。如图 3.20 所示，设τ 为单程传播时延。在$t=0$ 时，A 发送数据，B 检测到信道空闲。在$t=\tau-\delta$ 时，A 发送的数据还未到达 B，由于 B 检测到信道空闲而发送数据。经过时间$\delta/2$ 后，即在$t=\tau-\delta/2$ 时，A 发送的数据和 B 发送的数据发生碰撞，但这时 A 和 B 都不知道。在$t=\tau$ 时，B 检测到碰撞，于是停止发送数据。在$t=2\tau-\delta$ 时，A 检测到碰撞，也停止发送数据。显然，CSMA/CD 中的站不可能同时进行发送和接收，因此采用 CSMA/CD 协议的以太网不可能进行全双工通信，而只能进行半双工通信。

由图 3.22 可知，站 A 在发送帧后至多经过时间2τ 就能知道所发送的帧是否发生碰撞（$\delta\to0$ 时）。因此把以太网端到端往返时间 2τ 称为争用期（又称冲突窗口或碰撞窗口）。每个站在自己发送数据之后的一小段时间内，存在发生冲突的可能性，只有经过争用期这段时间还未检测到冲突时，才能确定这次发送不会发生冲突。

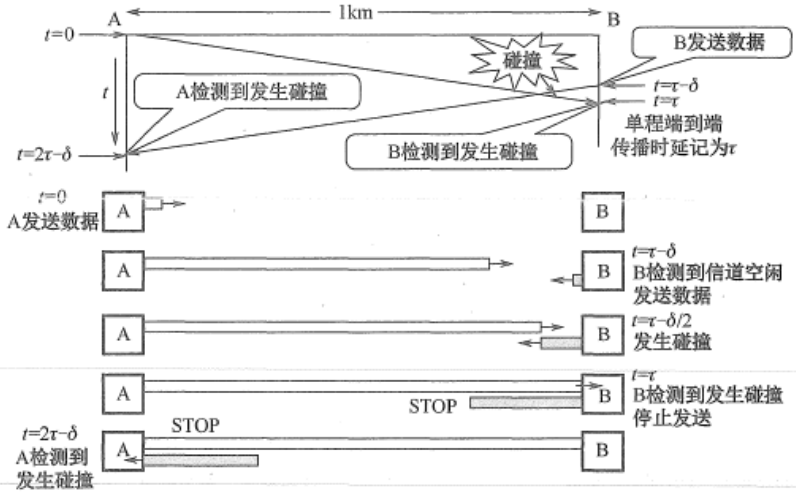


图 3.22　传播时延对载波帧听的影响

CSMA/CA协议(Collision Avoidance, 碰撞避免CA)

CSMA/CD 协议已成功应用于使用有线连接的局域网，但在无线局域网环境下，却不能简单地搬用 CSMA/CD 协议，特别是碰撞检测部分。主要有两个原因：

1. 接收信号的强度往往会远小于发送信号的强度，且在无线介质上信号强度的动态变化范围很大，因此若要实现碰撞检测，则硬件上的花费就会过大。
2. 在无线通信中，并非所有的站点都能够听见对方，即存在“隐蔽站”问题。

CSMA/CA使用预约信道、ACK 帧、RTS/CTS 帧等三种机制来实现碰撞避免：

1. 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。
2. 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。
3. 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。

> “
>
> 【CSMA/CD和CSMA/CA的区别】： 1) CSMA/CD 可以检测冲突，但无法避免； CSMA/CA 发送包的同时不能检测到信道上有无冲突，本结点处没有冲突并不意味着在接收结点处就没有冲突，只能尽量避免。2) 传输介质不同。CSMA/CD 用于总线形以太网， CSMA/CA 用于无线局域网 802.11a/b/g/n 等。3) 检测方式不同。CSMA/CD 通过电缆中的电压变化来检测；而CSMA/CA 采用能量检测、载波检测和能量载波混合检测三种检测信道空闲的方式。
>
> ”

> “
>
> ”

> 【总结】：总结： CSMA/CA 协议的基本思想是在发送数据时先广播告知其他结点，让其他结点在某段时间内不要发送数据，以免出现碰撞。CSMA/CD 协议的基本思想是发送前侦听，边发送边侦听，一旦出现碰撞马上停止发送。
>
> ”

4.轮询访问介质访问控制：令牌传递协议

在轮询访问中，用户不能随机地发送信息，而要通过一个集中控制的监控站，以循环方式轮询每个结点，再决定信道的分配。当某结点使用信道时，其他结点都不能使用信道。典型的轮询访问介质访问控制协议是令牌传递协议，它主要用在令牌环局域网中。在令牌传递协议中，一个令牌在各结点间以某个固定次序交换。令牌是由一组特殊的比特组合而成的帧。当环上的一个站希望传送帧时，必须等待令牌。一旦收到令牌，站点便可启动发送帧。帧中包括目的站的地址，以标识哪个站应接收此帧。帧在环上传送时，不管该帧是否是发给本站点的，所有站点都进行转发，直到该帧回到它的始发站，并由该始发站撤销该帧。帧的目的站除转发帧外，应针对该帧维持一个副本，并通过在帧的尾部设置“响应比特”来指示已收到此副本。站点在发送完一帧后，应释放令牌，以便让其他站使用。当计算机都不需要发送数据时，令牌就在环形网上游荡，而需要发送数据的计算机只有在拿到该令牌后才能发送数据帧，因此不会发送冲突（因为令牌只有一个）。 在令牌传递网络中，传输介质的物理拓扑不必是一个环，但是为了把对介质访问的许可从一个设备传递到另一个设备，令牌在设备间的传递通路逻辑上必须是一个环。轮询介质访问控制既不共享时同，也不共享空间，它实际上是在随机介质访问控制的基础仁限定了有权力发送数据的结点只能有一个。

---

day3:路由算法

1.路由算法综述

路由器转发分组是通过路由表转发的，而路由表是通过各种算法得到的。主机通常直接与一台路由器相连接，该路由器即为该主机的默认路由器(defaultrouter)，又称该主机的第一跳路由器(first-hop router)每当主机发送一个分组时，该分组被传送给它的默认路由器。源主机的默认路由器称作源路由器(sourcerouter),目的主机的默认路由器称作目的路由器(destination router)。
一个分组从源主机到目的主机的路由选择问题显然可归结为从源路由器到目的路由器的路由选择问题。

> 【路由选择算法的分类】1)静态路由算法（又称非自适应路由算法）2)动态路由算法（又称自适应路由算法）;常用的动态路由算法可分为两类：距离 - 向量路由算法和链路状态路由算法。

2.静态路由算法

- 由网络管理员手工配置路由信息。当网络的拓扑结构或链路的状态发生变化时，网络管理员需要手工去修改路由表中相关的静态路由信息。大型和复杂的网络环境通常不宜采用静态路由。一方面，网络管理员难以全面了解整个网络的拓扑结构；另一方面，当网络的拓扑结构和链路状态发生变化时，路由器中的静态路由信息需要大范围地调整，这一工作的难度和复杂程度非常高。
- 静态路由算法的优点是简便、可靠，在负荷稳定、拓扑变化不大的网络中运行效果很好，因此仍广泛用于高度安全的军事系统和较小的商业网络。
- 动态路由算法能改善网络的性能并有助于流量控制；但算法复杂，会增加网络的负担，有时因对动态变化的反应太快而引起振荡，或反应太慢而影响网络路由的一致性，因此要仔细设计动态路由算法，以发挥其优势。

3.距离-向量路由算法（RIP）

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。这种路由选择表包含：
1) 每条路径的目的地（另一结点）。
2) 路径的代价（也称距离）。

> 【注意】这里的距离是一个抽象的概念，如RIP 就将距离定义为“跳数 " 。跳数指从源端口到达目的端口所经过的路由个数，每经过一个路由器，跳数加1 。

在这种算法中，所有结点都必须参与距离向量交换，以保证路由的有效性和一致性，也就是说，所有的结点都监听从其他结点传来的路由选择更新信息，并在下列情况下更新它们的路由选择表：

1) 被通告一条新的路由，该路由在本结点的路由表中不存在，此时本地系统加入这条新的路由。
2) 发来的路由信息中有一条到达某个目的地的路由，该路由与当前使用的路由相比，有较短的距离（较小的代价）。此种情况下，就用经过发送路由信息的结点的新路由替换路由表中到达那个目的地的现有路由。

> 【距离向量路由算法的实质】是，迭代计算一条路由中的站段数或延迟时间，从而得到到达一个目标的最短（最小代价）通路。它要求每个结点在每次更新时都将它的全部路由表发送给所有相邻的结点。显然，更新报文的大小与通信子网的结点个数成正比，大的通信子网将导致很大的更新报文。由于更新报文发给直接邻接的结点，所以所有结点都将参加路由选择信息交换。基于这些原因，在通信子网上传送的路由选择信息的数量很容易变得非常大。

> RIP(Routing Information Protocol,路由信息协议）,采用距离-向量算法，在实际使用中已经较少适用。在默认情况下，RIP使用一种非常简单的度量制度：距离就是通往目的站点所需经过的链路数，取值为0~16，数值16表示路径无限长。RIP进程使用UDP的520端口来发送和接收RIP分组。RIP分组每隔30s以广播的形式发送一次，为了防止出现“广播风暴”，其后续的分组将做随机延时后发送。在RIP中，如果一个路由在180s内未被刷新，则相应的距离就被设定成无穷大，并从路由表中删除该表项。RIP分组分为两种：请求分组和响应分组。

4.链路状态路由算法(OSPF)

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:
1) 第一，主动测试所有邻接结点的状态。两个共享一条链接的结点是相邻结点，它们连接到同一条链路，或者连接到同一广播型物理网络。
2) 第二，定期地将链路状态传播给所有其他结点。`典型的链路状态算法是OSPF算法。`

在一个链路状态路由选择中，一个结点检查所有直接链路的状态，并将所得的状态信息发送给网上的所有其他结点，而不是仅送给那些直接相连的结点。每个结点都用这种方式从网上所有其他的结点接收包含直接链路状态的路由选择信息。

每当链路状态报文到达时，路由结点便使用这些状态信息去更新自己的网络拓扑和状态“视野图”，一旦链路状态发生变化，结点就对更新的网络图利用Dijsktra最短路径算法重新计算路由，从单一的源出发计算到达所有目的结点的最短路径。

> 【注意】这是Dijsktra算法的一个实际应用，别忘了

链路状态路由算法主要有三个特征：
1) 向本自治系统中所有路由器发送信息，这里使用的方法是泛洪法，即路由器通过所有端口向所有相邻的路由器发送信息。而每个相邻路由器又将此信息发往其所有相邻路由器（但不再发送给刚刚发来信息的那个路由器）。

> 【洪泛法小知识】洪泛法（Flooding）是一种简单的路由算法，将收到的封包，往所有的可能连结路径上递送，直到封包到达为止。洪泛法被使用在桥接器上，Usenet以及点对点档案分享等。部份的路由协定也以洪泛法为基础，例如开放式最短路径优先（OSPF）、距离向量群体广播路由协定(DistanceVectorMulticastRoutingProtocol，DVMRP)。无线随意网络也使用洪泛法来进行路由。

2) 发送的信息是与路由器相邻的所有路由器的链路状态，但这只是路由器所知道的部分信息。所谓“链路状态”，是指说明本路由器与哪些路由器相邻及该链路的“度量”。对于OSPF 算法，链路状态的 " 度量” 主要用来表示费用、距离、时延、带宽等。
3) 只有当链路状态发生变化时，路由器才向所有路由器发送此消息。由于一个路由器的链路状态只涉及相邻路由器的连通状态，而与整个互联网的规模并无直接关系，因此链路状态路由算法可以用于大型的或路由信息变化聚敛的互联网环境。

链路状态路由算法的主要优点是，每个路由结点都使用同样的原始状态数据独立地计算路径，而不依赖中间结点的计算；链路状态报文不加改变地传播，因此采用该算法易于查找故障。当一个结点从所有其他结点接收到报文时，它可以在本地立即计算正确的通路，保证一步汇聚。最后，由于链路状态报文仅运载来自单个结点关于直接链路的信息，其大小与网络中的路径结点数目无关，因此链路状态算法比距离-向量算法有更好的规模可伸展性。

> 【距离-向量路由算法与链路状态路由算法的比较】：在距离-向量路由算法中，每个结点仅与它的直接邻居交谈，它为它的邻居提供从自已到网络中所有其他结点的最低费用估计。在链路状态路由算法中，每个结点通过广播的方式与所有其他结点交谈，但它仅告诉它们与它直接相连的链路的费用。相较之下，距离-向量路由算法有可能遇到路由环路等问题。

> 【路由环路问题】：在维护路由表信息的时候，如果在拓扑发生改变后，网络收敛缓慢产生了不协调或者矛盾的路由选择条目，就会发生路由环路的问题，这种条件下，路由器对无法到达的网络路由不予理睬，导致用户的数据包不停在网络上循环发送，最终造成网络资源的严重浪费。（例子和解决方案由于篇幅过大，感兴趣的请自行百度百科（狗头保命））

> OSPF(Open Shortest Path First开放式最短路径优先）是对链路状态路由协议的一种实现，著名的迪克斯加算法被用来计算最短路径树。OSPF支持负载均衡和基于服务类型的选路，也支持多种路由形式，如特定主机路由和子网路由等。OSPF的简单说就是两个相邻的路由器通过发报文的形式成为邻居关系，邻居再相互发送链路状态信息形成邻接关系，之后各自根据最短路径算法算出路由，放在OSPF路由表，OSPF路由与其他路由比较后优的加入全局路由表。

5.层次路由

当网络规模扩大时，路由器的路由表成比例地增大。这不仅会消耗越来越多的路由器缓冲区空间，而且需要用更多CPU 时间来扫描路由表，用更多的带宽来交换路由状态信息。因此路由选择必须按照层次的方式进行。

因特网将整个互联网划分为许多较小的自治系统（注意一个自治系统中包含很多局域网），每个自治系统有权自主地决定本系统内应采用何种路由选择协议。如果两个自治系统需要通信，那么就需要一种在两个自治系统之间的协议来屏蔽这些差异。据此，因特网把路由选择协议划分为两大类
1) 一个自治系统内部所使用的路由选择协议称为内部网关协议(IGP), 也称域内路由选择，具体的协议有RIP 和OSPF 等。
2) 自治系统之间所使用的路由选择协议称为外部网关协议(EGP), 也称域间路由选择，用在不同自治系统的路由器之间交换路由信息，并负责为分组在不同自治系统之间选择最优的路径。具体的协议有BGP 。

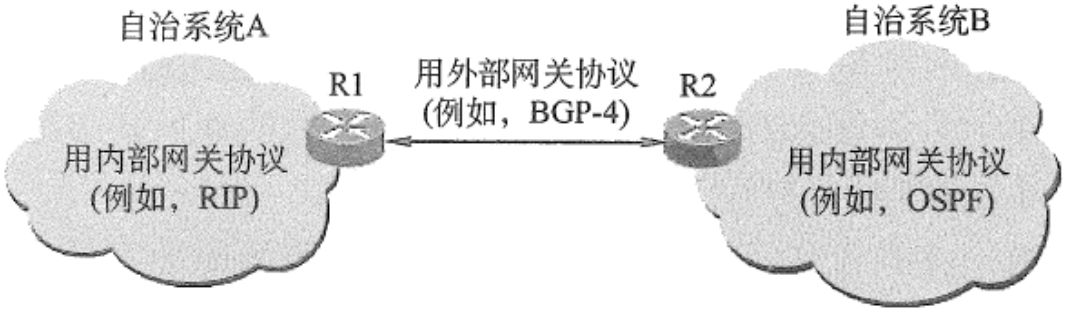


图 4.7　自治系统和内部网关协议、外部网关协议

使用层次路由时， OSPF 将一个自治系统再划分为若干区域(Area), 每个路由器都知道在本区域内如何把分组路由到目的地的细节，但不用知道其他区域的内部结构。采用分层次划分区域的方法虽然会使交换信息的种类增多，但也会使OSPF 协议更加复杂。但这样做却能使每个区域内部交换路由信息的通信量大大减小，因而使OSPF协议能够用于规模很大的自治系统中。

表 4.3　三种路由协议的比较

协　议	RIP	OSPF	BGP	
类型	内部	内部	外部	
路由算法	距离-向量	链路状态	路径-向量	
传递协议	UDP	IP	TCP	
路径选择	跳数最少	代价最低	较好，非最佳	
交换结点	和本结点相邻的路由器	网络中的所有路由器	和本结点相邻的路由器	
交换内容	当前本路由器知道的全部信息，即自己的路由表	与本路由器相邻的所有路由器的链路状态	首次	整个路由表
			非首次	有变化的部分