面试真题

【美团】HTTP常见的方法有哪些？

1. **GET**
2. **POST**：用于向服务器提交数据，通常用于创建新资源或提交表单。参数包含在请求体中，可以处理较大的数据量。
3. **PUT**：用于更新资源，将请求体中的数据完全替换目标资源的内容。如果资源不存在，可能会创建新资源（取决于服务器配置）。
4. **DELETE**
5. **HEAD**：类似GET，但不返回响应体，只返回响应头，常用于检查缓存有效性。

HEAD 方法就好像你去商店问店员某个商品的情况，但是你只关心商品的一些基本信息，比如价格标签、生产日期等，而不想要把商品本身拿回家看。  
当你想检查一个网页是否更新了，但又不想把整个网页内容都下载下来看时，就可以用 HEAD 方法。浏览器或者一些工具发送一个 HEAD 请求到服务器，服务器就会只返回响应头信息。通过查看响应头中的 “Last-Modified” 字段，就能知道这个网页最后修改的时间，以此判断是否有更新。如果缓存里有这个网页，还可以根据响应头中的缓存相关字段，比如 “Expires” 或 “Cache-Control”，来确定缓存是否还有效。

1. **OPTIONS**：用于查询某个资源支持的HTTP方法，通常用于跨域请求的预检请求。  
   OPTIONS 方法就好比你要去一个新的游乐场玩，但是不确定这个游乐场里的每个项目你能不能玩，于是你先去问工作人员每个项目都有什么要求，有哪些项目可以玩。在网络中，当你的浏览器要向一个不同域名的服务器发送请求（比如从你的网站请求另一个公司网站的数据，这就是跨域请求）时，它会先发送一个 OPTIONS 请求，去问服务器这个资源支持哪些 HTTP 方法，比如支不支持 GET、POST 等。服务器收到这个 OPTIONS 请求后，就会告诉浏览器它支持哪些方法，这样浏览器就知道后续可以用什么方法来请求数据了。

在使用 JavaScript 进行跨域的 AJAX 请求时，浏览器会自动先发送 OPTIONS 请求进行预检。比如你在自己的网页上写了一段代码，想要从另一个域名的服务器获取数据，浏览器会先发送 OPTIONS 请求到目标服务器，询问 “我能不能用 GET 方法获取这个数据呀” 等信息。服务器会返回一个响应，告诉浏览器它支持的方法，可能会在响应头中通过 “Access-Control-Allow-Methods” 字段来列出支持的方法，像 “GET, POST, PUT, DELETE” 等，浏览器收到后，就知道后续该怎么正确地发送请求了。

1. **PATCH**：用于对指定资源进行部分修改，客户端只需要提交更改的部分数据，而不是替换整个资源。  
   通俗解释：PATCH 方法就好像你要修改一幅画，但是你只需要修改其中的一部分，比如把画里的一朵花颜色改一下，而不用把整幅画重新画一遍。在网络请求中，如果你要修改服务器上的某个资源，比如一个文档、一条数据库记录等，不需要把整个资源的所有内容都重新提交一遍，只需要把你要修改的那部分内容告诉服务器就行。

假设你在一个网站上有自己的个人资料，你只想修改其中的电话号码，就可以使用 PATCH 方法。你通过客户端（比如手机 APP 或者网页）发送一个 PATCH 请求到服务器，请求中只包含你要修改的电话号码这部分数据，服务器收到后，就会根据你提供的数据，只修改数据库中你个人资料里的电话号码字段，而其他信息，像姓名、地址等都保持不变。

【美团】get和post方法的区别？

* 用途：GET获取数据；POST提交数据
* 数据传输方式和安全性：GET通过URL传少量数据，较不安全；POST通过请求体传大量数据，相对更安全
* 幂等性：GET幂等，执行多次结果相同；POST非幂等，执行多次会改变服务器上的数据。

面向简历

TCP、UDP

TCP（Transmission Control Protocol）和UDP（User Datagram Protocol）是两种常用的传输层协议，它们在网络通信中有一些重要的区别：

1. 连接性：

TCP： 提供面向连接的通信。在数据传输之前，必须先建立连接，然后进行可靠的数据传输，最后再释放连接。

UDP： 是面向无连接的通信。每个数据包都是独立的，不需要先建立连接，直接发送数据。因此，UDP通信更为轻量，但不保证可靠性。

2. 可靠性：

TCP： 提供可靠的数据传输，通过序列号、确认和重传机制确保数据的完整性和顺序性。如果数据包丢失，TCP会进行重传。

UDP： 不提供可靠性保证，数据包可能会丢失，也不保证数据包的顺序性。对于某些实时性要求高的应用，可以容忍少量的丢包。

6. 连接和断开：

TCP： 建立连接时有三次握手的过程，断开连接有四次挥手的过程。

UDP： 无连接的特性，不需要建立和断开连接的过程。

5. 适用场景：

TCP： 适用于对数据完整性和顺序性有要求的应用，如文件传输、网页访问等。

UDP： 适用于实时性要求高、可以容忍少量数据丢失的应用，如语音通话、视频直播等。

3. 数据传输方式：

TCP： 数据以流的方式进行传输，被划分为小的数据段，然后通过序列号进行重新组装。数据传输是可靠而有序的。

UDP： 数据以数据包的形式传输，每个数据包都是独立的。UDP更适用于需要快速传输的场景，如实时音视频。

4. 头部开销：

TCP： TCP头部相对较大，包含序列号、确认号、窗口大小等字段，用于保证可靠性和流控制。

UDP： UDP头部较小，只包含源端口、目标端口、长度和校验和等基本信息，开销较小。

总体来说，选择TCP还是UDP取决于具体的应用需求。TCP适用于要求可靠性和有序性的场景，而UDP适用于实时性要求高、可以容忍少量数据丢失的场景。在某些应用中，也可以结合使用TCP和UDP，根据不同的数据传输需求选择合适的协议。

HTTP1.0和HTTP1.1的区别是什么？

**持久连接**、**管道化**、**缓存控制**、**Host头部**、**错误代码**

**1. 持久连接**

* **HTTP1.0**：每次请求都需要建立一个新的TCP连接，响应之后关闭连接。
* **HTTP1.1**：默认启用持久连接（即Connection: keep-alive），允许多次请求/响应复用同一连接，减少了建立和关闭连接的开销。

**2. 管道化**

* **HTTP1.0**：不支持请求管道化，每次请求必须等待前一个请求完成。
* **HTTP1.1**：支持请求管道化，客户端可以在等待响应的同时发送多个请求，但服务器端必须按顺序返回响应。

**3. 缓存控制**

* **HTTP1.0**：缓存控制较为简单，仅依赖于Expires头部。
* **HTTP1.1**：新增了Cache-Control头部，提供了更细粒度的缓存控制，例如no-cache、no-store、max-age等。

**4. Host头部**

* **HTTP1.0**：不包含Host头部，无法在多个域名共享同一IP时进行区分。
* **HTTP1.1**：强制要求在请求中包含Host头部，允许同一IP托管多个虚拟主机（即多个域名）。

**5. 错误代码**

* **HTTP1.0**：错误码较少，错误响应不够细致。
* **HTTP1.1**：新增了多个错误码，如：
  + 409 Conflict：表示请求与当前状态冲突。
  + 410 Gone：表示资源已永久删除等。

HTTP2.0与HTTP1.1相比有哪些主要改进？

**多路复用**、**头部压缩**、**流量控制**、**服务器推送、二进制协议**

**1. 多路复用**

* **HTTP2.0：使用一个连接复用多个请求和响应，消除了HTTP1.1中的队头阻塞问题，使得请求和响应可以并行传输。**
* **HTTP1.1：虽然支持管道化，但请求和响应仍然是串行化的，且如果其中一个请求被阻塞，其他请求也会受到影响。**

**2. 头部压缩**

* **HTTP2.0：采用HPACK算法对HTTP头进行压缩，减少了冗余数据的传输，尤其对于重复的请求头，如Cookies等，效果显著。**
* **HTTP1.1：头部信息未压缩，且请求头较为冗长，这增加了网络带宽的消耗。**

**3. 流量控制**

* **HTTP2.0：支持对每个流进行独立的流量控制，服务器和客户端可以通过调整流的大小来管理带宽使用。**
* **HTTP1.1：没有内建流量控制，主要依赖TCP的拥塞控制，可能导致带宽浪费。**

**4. 服务器推送**

* **HTTP2.0：服务器可以在客户端请求之前主动推送资源，这对于加载复杂页面时有显著的性能提升。例如，在请求HTML文件的同时，服务器可以主动推送该页面依赖的CSS、JavaScript文件等。**
* **HTTP1.1：只有客户端可以发起请求，服务器只能被动响应，无法主动推送资源。**

**5. 二进制协议**

* **HTTP2.0：采用二进制协议，而不是文本协议，数据以帧的形式传输，解析更高效。**
* **HTTP1.1：是基于文本的协议，易于人类阅读和调试，但由于解析过程繁琐，性能不如二进制协议。**

HTTP3.0有了解过吗？它与之前的版本有哪些主要不同？

**传输协议、连接建立**、**多路复用**、**加密**、**头部压缩**：

**1. 基于QUIC协议**

* **HTTP3.0**：HTTP3不再依赖TCP，而是基于QUIC协议（Quick UDP Internet Connections）。QUIC是Google提出的一个新协议，建立在UDP之上，具有低延迟和更好的多路复用性能。
* **HTTP1.x 和 HTTP2**：两者都依赖于TCP协议，TCP的连接建立和管理较为复杂且容易受到延迟影响。

**2. 零RTT连接建立**

* **HTTP3.0**：QUIC协议支持零RTT连接建立（0-RTT），即客户端在首次发送请求时，可以立即开始数据传输，而无需等到完整的三次握手过程结束。通过保存会话信息，可以减少延迟。
* **HTTP1.x 和 HTTP2**：这两个版本都使用基于TCP的三次握手过程，每次建立新连接都需要等待完整的握手。

**3. 避免队头阻塞**

* **HTTP3.0**：QUIC协议为每个HTTP请求分配独立的流，所有流之间互不阻塞。即使某个流发生丢包或延迟，其他流依然能够继续传输数据，从而避免队头阻塞。
* **HTTP2**：虽然支持多路复用，但仍然受TCP队头阻塞的影响。当一个流的数据包丢失时，整个连接的延迟会增加，影响其他流的传输。

**4. 内建加密**

* **HTTP3.0**：QUIC将TLS加密内建于协议中，每个QUIC数据包都会自动加密，不需要额外的TLS握手过程。
* **HTTP2**：虽然支持TLS加密，但并没有内建，使用时需要显式配置TLS层。

**5. 头部压缩**

* **HTTP3.0**：与HTTP2类似，HTTP3继续使用HPACK进行头部压缩，提高了带宽效率。
* **HTTP1.1**：HTTP1.1没有内建头部压缩功能，导致大量重复头部信息的传输，浪费带宽。

HTTPS和HTTP有哪些区别？

**协议类型**、**端口号**、**加密方式**、**性能**、**安全性**

**1. 协议类型**

* **HTTP（超文本传输协议）**：是一种用于从Web服务器传输超文本到本地浏览器的协议。HTTP协议是无状态的，不提供数据加密和认证，所有信息以明文形式传输。
* **HTTPS（安全超文本传输协议）**：是在HTTP协议基础上加入SSL/TLS加密层，保证数据的保密性和完整性，防止中间人攻击。

**2. 端口号**

* **HTTP**：默认为80端口。
* **HTTPS**：默认为443端口，通信是加密的，增加了一个安全层。

**3. 加密和传输层**

* **HTTP**：数据在传输过程中是未加密的，容易被拦截、篡改。
* **HTTPS**：通过SSL/TLS协议进行加密，使得数据在传输过程中即使被截获，也无法解密。

**4. 性能**

* **HTTP**：由于没有加密过程，性能较好，传输速度较快。
* **HTTPS**：虽然由于加密和解密过程稍有性能损失，但现如今的硬件和加密算法优化已经让其性能差距不再显著。

**5. 安全性**

* **HTTP**：数据传输过程中不进行加密，容易被窃听和篡改，存在中间人攻击风险。
* **HTTPS**：通过SSL/TLS协议提供数据的加密、完整性验证及认证机制，有效防止中间人攻击、数据篡改及窃取。

HTTPS工作原理是什么？它是如何实现数据加密的？

**1. SSL/TLS协议**

* **SSL/TLS**协议是实现HTTPS加密通信的核心技术，TLS是SSL的继任版本。SSL/TLS用于加密数据传输、验证身份并确保数据完整性。TLS提供了对称加密、非对称加密和哈希算法的结合，确保数据在传输中的机密性、完整性和认证。

**2. 加密过程**

* **对称加密**：HTTPS使用对称加密算法（如AES）对传输的数据进行加密。对称加密的优点是加密和解密速度快，但问题是密钥的安全性。因为双方使用相同的密钥加密和解密数据，必须确保密钥的安全传输。
* **非对称加密**：用于密钥交换和身份认证。非对称加密使用公钥和私钥对数据进行加解密，公钥加密的数据只能用对应的私钥解密。服务器通过数字证书将公钥发送给客户端，客户端用该公钥加密会话密钥（对称密钥），然后发送给服务器。服务器用自己的私钥解密该密钥。

**3. 握手过程**

* **SSL/TLS握手**：HTTPS通信开始时，客户端与服务器会进行SSL/TLS握手，以下是握手的基本步骤：
  1. **客户端Hello**：客户端向服务器发送支持的加密算法、生成的随机数等信息。
  2. **服务器Hello**：服务器从客户端提供的算法中选择一个并返回自己的证书（包含公钥），以及用于生成会话密钥的随机数。
  3. **密钥交换**：客户端生成一个会话密钥并使用服务器的公钥加密该密钥，然后发送给服务器。服务器使用自己的私钥解密，获得会话密钥。
  4. **验证身份**：客户端验证服务器的数字证书，以确保与预期的服务器通信，防止中间人攻击。
  5. **完成握手**：客户端和服务器使用协商的会话密钥进行加密通信。

**4. 数据传输**

* **加密数据传输**：一旦握手完成，客户端和服务器就开始使用对称加密算法（如AES）加密通信数据。此时，通信双方使用的是会话密钥（由握手过程中的密钥交换生成），确保数据在传输过程中既不会被窃取，也不会被篡改。
* **数据完整性**：在数据传输过程中，SSL/TLS协议使用哈希算法（如SHA-256）对数据进行完整性验证，确保数据在传输过程中没有被篡改。

知识框架

1.OSI/RM

**① 应用层：**在用户应用程序与网络间作为接口，使能交互

**② 表示层：**对数据编码、加密、压缩、转换，使应程理解数据报文的真实含义

**③ 会话层：**建立、保持、结束会话

**④ 传输层**（通信子网与资源子网的分界层）

提供端到端的可靠透明数据传输

传输连接管理（连接建立、维护和拆除）

传输差错处理（差错控制、流量控制）

**⑤ 网络层**（通信子网的最高层）

逻辑地址寻址（IP地址翻译成物理地址）

路由选择（为分组选择最适当的路径）

**⑥ 数据链路层**

提供可靠数据传输链路，分为：

MAC解决信道竞争问题，完成介质访问控制。

LLC负责建立和维护网络连接，执行差错控制、流量控制。

**⑦ 物理层**

提供正确透明数据传输介质。（机械、电气、功能、过程特性）

实现相邻计算机节点之间比特流的透明传送（数模转换与模数转换）

屏蔽具体设备的差异，使链路层不必关心网络的具体传输介质。

2.以太网工作原理

（1）无连接无确认

（2）采用曼彻斯特编码

（3）基于CSMA/CD

先听后说 | 边听边说；

一旦冲突，立即停说；

等待时机，然后再说.

**① 载波侦听 CS**

发送前用电子技术侦听总线上是否有数据传输

若无数据传输，**立即**发送准备好的数据

若有数据传输，则不发送数据

**② 多路访问 MA**

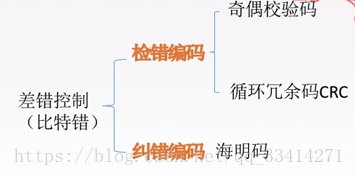
所有的工作站**共享**数据信道，以**广播**方式发送

**③ 冲突检测 CD**

发送中不断检测冲突

检测到**冲突**、发送完**阻塞**信号后，等待**随机**时间再监听二进制指数退避算法

3.数据链路层如何进行差错控制



（1）检错编码——奇偶校验码: 只能检查出奇数个比特错误。

（2）检错编码——CRC循环冗余码

接收端把数据帧**除以**一个除数，然后检查得到的**余数**

1.余数为0，判定没有差错接受。

2.余数为不为0，判定有差错丢弃。

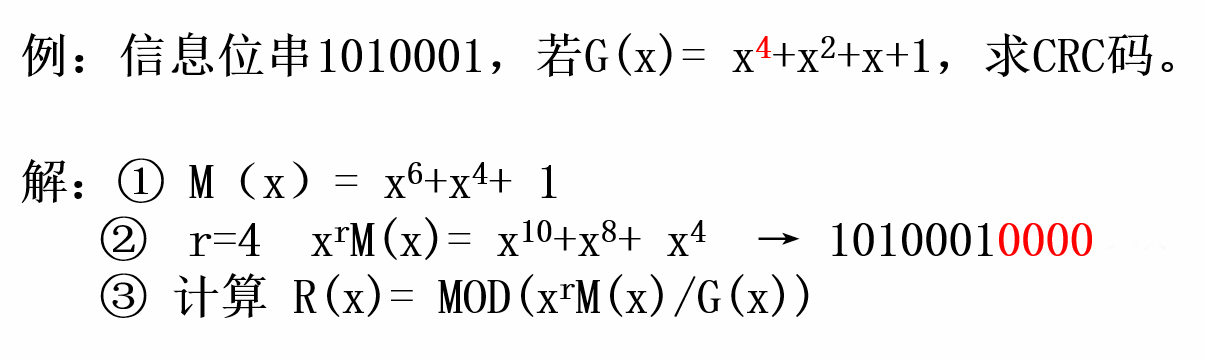
∵ 仅使用循环冗余检验CRC差错检测技术，只能做到对帧的无差错接收

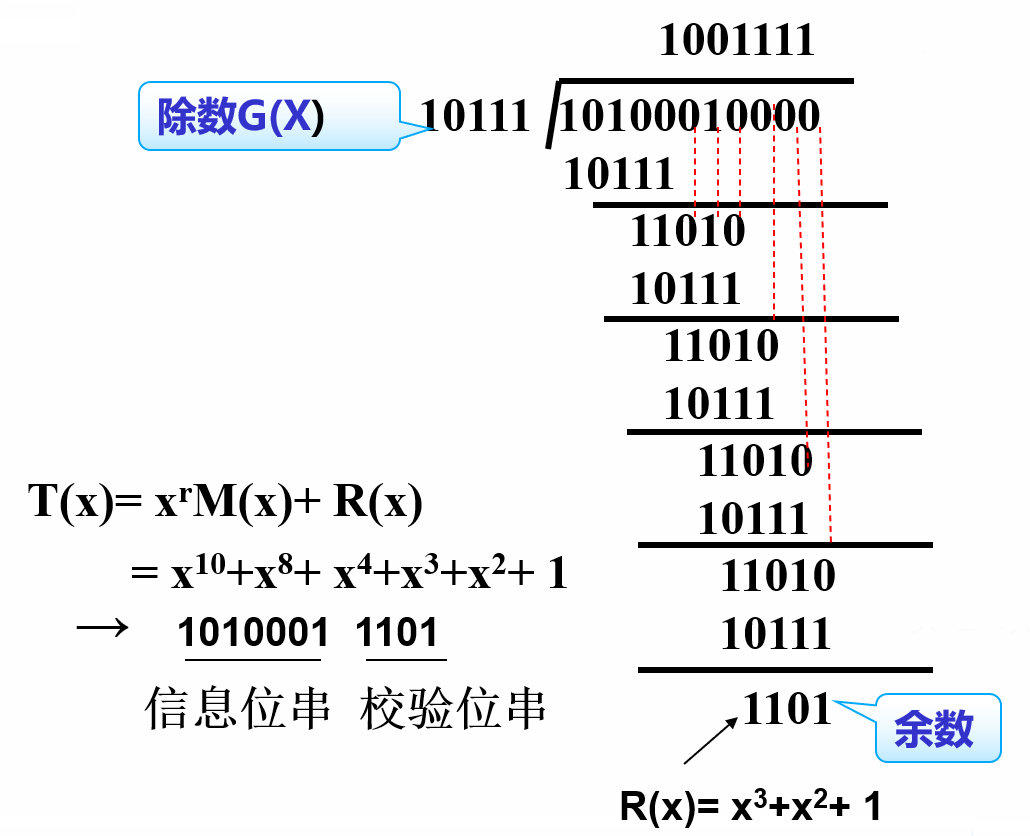
&& “可靠传输”：数据链路层发送端发送什么，接收端就收到什么。

∴ CRC检验能实现无比特差错的传输，但还不是可靠传输。

（3）海明码：**检**测**双**比特错，**纠**正**单**比特错。

4. CRC校验码计算。说明发送和接收的原理。





（1）**G（x）**生成多项式，最高次xr

（2）**xrM（x）**发送的数据帧后**加**r位“0”

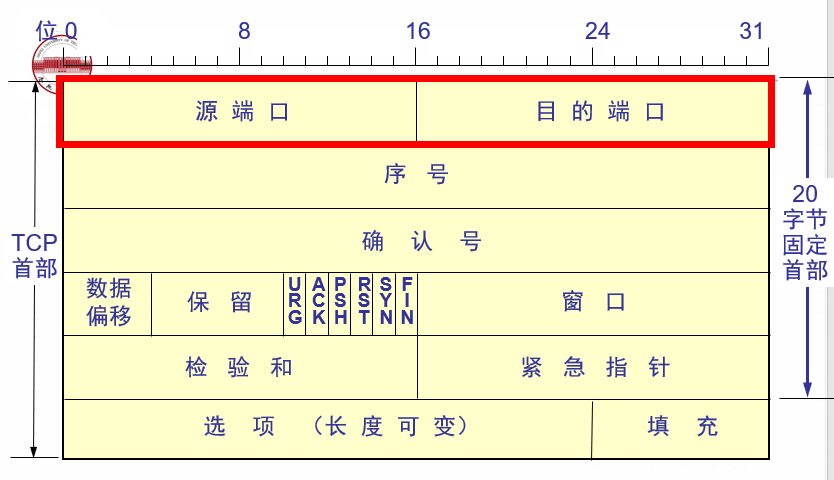
（3）**R（x）** 以“**模2除法**”计算xrM（x）/G（x），得到的余数即CRC校验码。

（4）**附加**：把校验码附加在原数据帧后发送。

（5）**检验**：接收端把数据帧除以G（X），如果余0则传输过程中没出错。

5. TCP报文格式

5.1序号、确认序号、FIN含义



**端口**是传输层与应用层的服务接口。实现传输层的复用和分用功能。

**序号seq**：数据流**按字节**编**号**。

序号值为**本报文段**的**数据**的**第一个字节**的序号。

**确认号ack**：**期望**收到的下一个报文段的数据的第一个字节的序号。

**数据偏移（首部长度）**：指出数据距离起始处有多远。（单位 32 位字）

**保留字段**：保留为今后使用，置为 0。

**紧急URG：**有**紧急**数据，应**尽快传送**(相当于**高优先级**的数据)。

**确认ACK：**确认位，为1时确认号字段才有效

**推送 PSH**：**不等缓存都填满**，**尽快**向上**交付**。

**复位 RST**：由于主机**崩溃**或其他原因出现**严重差错**，必须**释放重**连。

**同步 SYN**：表示**连接请求**或**连接接受**报文。

**终止 FIN**：**数据已发送完毕**，并**要求释放**连接。

**窗口**：用来让对方设置发送窗口的依据（单位字节）

**检验和**：检验范围包括**首部**和**数据**。计算时，要加上 **12 字节**的**伪首部**。

**紧急指针**：指出**紧急数据字节数**（紧急数据放在本报文段数据的**最前面**）

**选项**：（1）**MSS**选项：**数据**字段的**最大长度**。（加上首部 整个TCP报文段。）

（2）**窗口扩大**选项

其中有一个字节表示移位值 S。

新的窗口值等于TCP 首部中的窗口位数**增大到(16 + S)**

（3）**时间戳**选项，主要是两个：时间戳**值**和时间戳**回送回答**

用作**乱序**的时间判断依据，以及**计算RTT**

（4）**选择确认**选项

**有选择的确认收到**的某些报文，以**提高效率**。

**填充**：使整个首部**长度是 4 字节的整数倍**。

5.2如何流量控制

5.2.1滑动窗口实现

（1）滑动窗口**面向字节流**：**传输**和**确认**都是**基于字节序号**而不是报文序号。

（2）**每端**都有：一个**发送窗口**和一个**接收窗口。**

**发送方**根据**接收方**给出的窗口值，构造发送窗口尺寸。**不大于**对方的窗口值。

（3）发送窗口 [P1, P3)

接收窗口对**不按序到达**的数据，先**临时存放**在接收窗口中，等**收到**字节流中所缺少的字节后，再**一并交付**上层应用程序。

（4）收到**确认号增加K1**且**窗口值增加K2**

窗口（左）**后沿+K1**，（右）**前沿+(K1+K2)**。

理解：K1来自对面确认后**返现**的，K2来自接收窗口值**增加/减少**的**BONUS**

（5）缓存与接收缓存

**① 发送缓存**暂时存放：**准备发送**的数据和**已发出但未确认**的数据。

**② 接收缓存**暂时存放：**按序到达但未被读取**的数据和**不按序到达**的数据。

5.2.2坚持计时器

① 背景：接收方发出**零窗口确认报文**，**发送方停止**发送。

接收方的**非零窗口确认报文**丢失，通信会陷入**死锁**。

② 发送方**收到零窗口确认报文**时，启动**坚持计时器**：

若设定时间内收到**非零窗口确认**报文，则**撤销**。

若设定时间内未收到，则发出只有**一个字节**数据的**探测报文**并**消耗一个序号**（该序号会被确认报文所忽略）若截止时间后没有收到发送**另一个探测报文**，并**加倍设置**坚持计时器值，如此**反复**直到收到接收方的非零窗口确认报文为止。（最大值限制60秒）

5.2.3糊涂窗口综合症的解决

糊涂窗口综合症：接收端和发送端**速率不匹配**时**产生很多小包**。

法一：Nagle算法——治疗“发送端糊涂”

法二：Clark & 延迟ACK——治疗“接收端糊涂”

任何**小于MSS**的数据报文都被视为小数据报文。

**小数据报文**大量的出现会造成广域网上的**拥塞**。

**增加发送的数据规模**不但可以**提高信道利用率**，也有助于**减少广域网的拥塞**。

① 描述

这个问题可以归结为**小包**的问题：由于发送端和接收端上的**处理不一致**，导致网络上**产生很多的小包**。在滑动窗口机制下，如果发送端和接收端**速率差异很大**，就会产生这种比较犯傻的状态：发送方发送的数据**有效数据很少**。

② 解决

法一：Nagle算法——治疗“发送端糊涂”

法二：Clark、延迟ACK——治疗“接收端糊涂”

（4.1）Nagle算法

尽可能**避免发送小数据报文**，一个TCP连接上**最多只能有一个**未被确认的小数据报文。

在小数据报文的确认报文到达之前不发送其他小数据报文。并且**持续接受上层小数据报文**，并在**确认到来时以一个较大的报文发送**出去 或 当到达的数据已达到发送窗口大小的**一半或报文段最大长度**时，就**立即发送**一个报文段！

（4.2）Clark方法

只要有数据**到达就发送确认**，但**宣布的窗口大小为零**，直到或者缓存空间**已能放入具有最大长度的**报文段，或者缓存空间的**一半已经空**了。

（4.3）延迟ACK

延迟ACK就是**累积确认（+捎带应答）**。

如果这段时间内有数据发送到对端，则**捎带发送ack**；

如果在延迟ack定时器触发时，则立即单独发送。

（4.4）当Nagle遇上延迟ACK

发送端产生了多个小数据报文需要向接收端发送

第一个小数据报文到接收端后，因为延迟ack而没有及时ack。

发送端等待对接收端对第一次数据确认。

结果：网络延迟。

5.3如何差错控制

5.3.1检验和

报文检验和字段可检查报文段是否损伤：如果检验和无效则认为损伤，丢弃。

5.3.2确认ack和选择确认SACK选项

（1）肯定积累确认（ack）：**期望接收**的下一个字节序号（忽略所有失序到达的报文段）。仅在**ACK标志为1**时才有效。

（2）选择确认（SACK）：报告**失序**以及**重复**的数据块。

5.3.3超时重传时间RTO

当**无法快速交付**数据报文时**减小网络负担**。

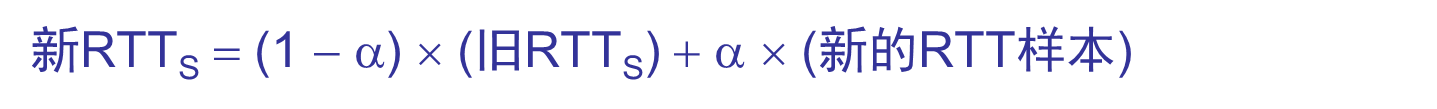
（1）**RTO 小**于 RTT，则会造成很多**不必要的重传**；

**RTO 远大**于 RTT，则会**降低整体网络利用率**。

（2）

每发生一次**重传**，RTO加倍，直到收到重传数据包的应答，退避因子恢复1。

（2）加权平均往返时间RTTS



若 α 更接近于**0**，表示 RTT 值更新较**慢**。

若选择 α 更接近于 **1**，则表示 RTT 值更新较**快**。

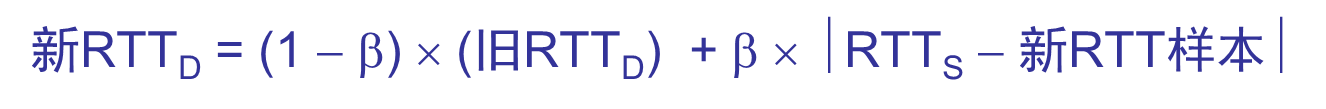
RFC 2988 推荐的 α 值为 **1/8，即 0.125**。

（3）超时重传时间



其中：RTTD 是 RTT 的**偏差加权平均值**。

第一次测量，RTTD = RTT / 2



推荐值是 **1/4，即 0.25**。

（4）RTT测量的复杂性和困难

① 重传后收到了上一个未确认报文的ACK

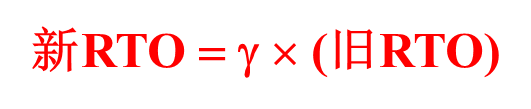
② IP数据报所选择的路由变化很大。因而运输层的往返时间的方差也很大。

（5）**Karn 算法**：不采用重传报文段的往返时间样本。

① 理由：当**重传发生**时，**RTO**已经采用了**指数退避**。下一次的传输将按照这个**已经增大的RTO**计时，不应该再将其纳入到RTT的统计样本中。

② 问题：当**发生重传**时，时延**增大**或**出错**，可能会带来**连续重传**，从而使超时重传时间 **RTO一直得不到更新**，无法反映当前现状。

③ 修正：每发生一次**重传**，即**增大一些RTO值**

系数 γ 的典型值是 2

（6）指数避退

**超时而发生连续多次重传**时，**超时阈值**遵循指数避退：

下一次是这次的2倍，2为退避因子

直到收到重传数据包的应答，RTO退避因子回复为1。

目的：为了当网络处于**无法快速交付**数据报文状态时**减小网络负担**。

5.4连接建立——三次握手

第一次: 客户端发出**同步请求**报文。

第二次：服务器发出**同步确认**报文。

第三次: 客户端发出**确认**报文。



5.5连接释放——四次挥手

第一次：客户端发出**连接释放**报文。

第二次：服务器发出**确认**报文。

第三次：服务器发出**连接释放确认**报文。

第四次：客户端发出**确认**报文。



MSL（Maximum Segment Lifetime）

任何报文在网络上的存在的最长时间超过这个时间报文将被丢弃。

RFC 规定为2分钟，常用的有30秒、1分钟、2分钟等。

6. IP地址和物理地址区别

“行可曾倡议？”

（1）性质

IP地址是逻辑地址

MAC地址是物理地址

（2）可变性

IP地址可以更改，不具备唯一性

MAC地址不可改变，具有唯一性，硬件出厂时固定。

（3）层次

IP——网络层

MAC——数据链路层

（4）长度

IP地址主流是32位长通过ARP协议联系到一起。

MAC地址长度为48位是Ethernet网卡上带的地址

（5）分配依据

IP分配是基于网络拓扑

MAC分配是基于制造商

7.如何进行IP地址和MAC地址解析？

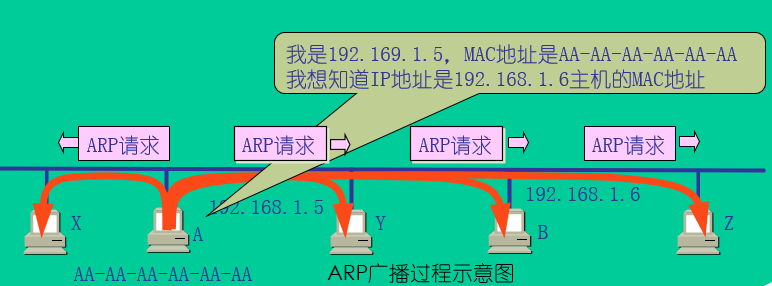
**地址解析协议ARP：**

（1）若本地**ARP表**中有主机B的**IP地址**，

则**按照对应的MAC地址**，将IP数据包封装成帧发送。

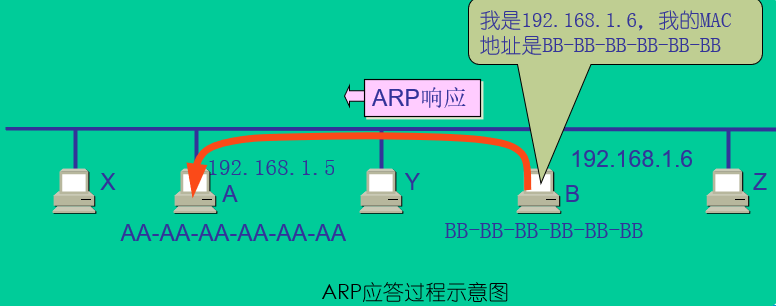
（2）若ARP表中**没有**主机B的数据项

① **广播ARP请求包**，在本网段内寻找主机B的MAC地址。



② 主机B收到后，向A发出**单播ARP响应包**，将A的IP和MAC填入表中。

③ 主机A收到后，将B的MAC地址填到表中，将IP数据包封装成帧发送。



8.解释RIP和 OSPF协议的三个要点?

**RIP 协议**

（1）仅和**相邻路由器**交换信息

（2）交换的信息是本路由器所知道的**全部信息**，即自己的路由表

（3）按**固定**的时间间隔交换路由信息。

**OSPF协议**

（1）**洪泛法**向本自治系统中的**所有路由器**发送信息

（2）发送的是与本路由器相邻路由器的**链路状态**，但这只是路由器所知道的**部分信息**。链路状态：本路由器都和哪些路由器相邻以及该链路的“度量”

（3）只有当**链路状态**发生**变化**时才发送

9.画出 TCP/IP协议体系,并简述各层功能?

**应用层**：提供网络服务，比如 FTP、DNS等

**传输层**：提供端到端可靠数据流传输服务

**网际层**：负责异构网或同构网的进程间通信,

将传输层分组封装为数据报格式进行传送,

每个数据报必须包含目的地址和源地址。

**网络接口层**：是网络访问层，负责与物理网络的连接。

10.以太网使用的CSMA/CD协议是以争用方式接入到共享信道。这与传统的时分复用TDM相比优缺点如何?

**（1）动态静态**

CSMA/CD 是动态的媒体随机接入|共享信道方式

TDM 静态划分信道

（2）利用率和碰撞

CSMA/CD 是用户共享信道更灵活，可提高信道利用率，

但也可能发生碰撞而降低信道利用率。

TDM为用户按时隙固定分配信道，如果没有数据要传送则被浪费，

但在分配的时隙中不会发生冲突。

（3）局域网和计算机

对**局域网**来说，连入信道的用户相距较近、信道带宽较宽，如果使用TDM 方式，在自己的时隙内没有数据发送的情况会更多，不利于信道的充分利用。

对**计算机**通信来说，突发式的数据更不利于使用TDM方式。

11.试简单说明IP、ARP 、RARP 和ICMP协议的作用。

（1）**IP网际协议**：

① 是TCP/IP体系中两个最重要的协议之一，使互连的计算机网络能够通信。

② 提供无连接的数据报传输，尽最大可能交付，无法控制是否成功。

（2）**ARP地址解析协议**：IP地址 -> 物理地址

**RARP逆向地址解析协议**：物理地址 -> IP地址

（3）**ICMP 网际控制报文协议**：

① 进行差错控制，减少分组的丢失，**提高 IP 数据报交付成功的机会**。

② ICMP 允许主机或路由器报告**差错情况**和提供有关**异常情况**的报告。

12.电路交换和分组交换的原理,比较其优缺点?

（1）电路交换：**独占信道资源**、信道利用率**低**、实时性**好；**

（2）分组交换：采用**存储转发**、信道利用率**高**、实时性**差；**

13.网桥的工作原理和特点是什么?网桥与转发器以及以太网交换视有何异同?

13.1网桥的工作原理和特点是什么?

（1）原理

网桥的**每个端口**与**一个网段**相连，网桥**从端口接收网段上传送的各种帧**。

每收到一个帧，就**暂存在其缓冲**中：① 若此帧**没有差错**，且目的MAC地址属于另一网段，则通过**查找站表转发**。② 若**出现差错**，则**丢弃**。

（2）特点：

① **过滤了**通信量，**扩大了**物理范围，**提高了**可靠性，

**可互连**不同物理层、不同MAC子层和不同速率的局域网。

② **增加了**时延，**不适合**用户**太多**和通信量**太大**的局域网。

13.2网桥与转发器以及以太网交换机有何异同?

**网桥与转发器：**

（1）**层次**：网桥工作在数据链路层

而转发器工作在物理层

（2）**转发**：网桥只转发未出现差错，且目的站属于另一网络的帧或广播帧

转发器转发所有的帧

（3）**检测**：网桥在转发前必须执行CSMA/CD算法

转发器转发时不用检测传输媒体

（4）**扩网**：网桥和转发器都能**扩展局域网**

但网桥还能**提高局域网的效率**并**互连不同MAC子层和不同速率局域网**

**网桥与以太网交换机：**

（1）**层次**：都工作在数据链路层;

（2）**端口**：以太网交换机通常有十几个端口

网桥一般只有2-4个端口;

（3）**接到**：网桥端口一般连接到局域网，允许每个网段上的计算机同时通信

交换机端口直接与主机相连，允许多对计算机间能同时通信。

（4）**实质**：以太网交换机是一个多端口的网桥

连到交换机上的每台计算机就像连到网桥的一个局城网段上。

（5）**转发方式**：网桥采用存储转发方式进行转发

以太网交换机还可采用直通方式，速度更快。

14.数据链路层如何进行流量控制？

（1）停止等待协议

发完一个分组，停止发送，等待确认后发送下一个

（2）GBN的连续ARQ协议

发送多个数据帧，以增大信道利用率

（3）选择重传协议SR

15.CSMA/CD与CSMA/CA对比

CSMA/CD：总线以太网、电压变化检测、检测

CSMA/CA：无线局域网、能量载波混合检测、避免

相同点：都是先听再说。都是有限重传。

16.令牌环网工作原理

（1）原理：工作站通过环形信道通信，通过令牌传递方式控制发送权

（2）过程：截获令牌后发送、接收和转发、撤销帧和重发令牌

17.慢开始和拥塞避免（2021）

17.1慢开始和拥塞避免

（1）发送方维持**拥塞窗口 cwnd**.窗口大小**取决于网络的拥塞程度**，**动态变化**。

原则：

网络**没有拥塞**，拥塞窗口就再**增大**一些，以便**把更多的分组发送出去**。

网络**出现拥塞**，拥塞窗口就**减小**一些，以**减少注入到网络中的分组数**。

（2）慢开始算法的原理

**刚刚开始**发送时设置拥塞窗口 **cwnd = 1**，即一个MSS的数值。（一倍）

在**每收到**一个**确认**后，将**拥塞窗口加 1**，即增加一个 MSS 的数值。（加倍）

（3）传输轮次

每经过**一个传输轮次**，拥塞窗口 cwnd 就**加倍**。

一个传输轮次所经历的时间其实就是**往返时间 RTT**。

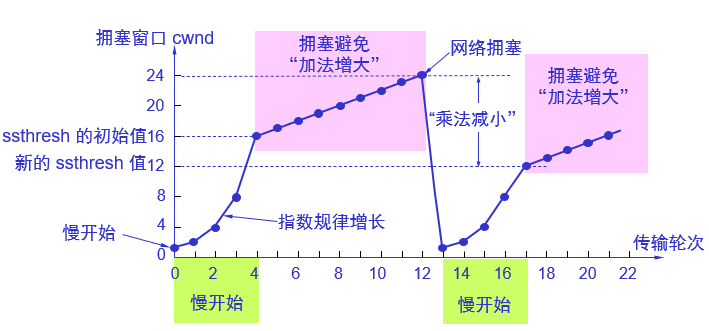
（4）门限状态变量（ssthresh）

当 cwnd < ssthresh 时，使用**慢开始算法**。

当 cwnd > ssthresh 时，**改用拥塞避免算法**。

**拥塞避免算法**：

是让拥塞窗口 cwnd **缓慢地增大**，即每经过一个往返时间 RTT 就把发送方的拥塞窗口 cwnd **加 1**，使拥塞窗口 cwnd 按线性规律缓慢增长。



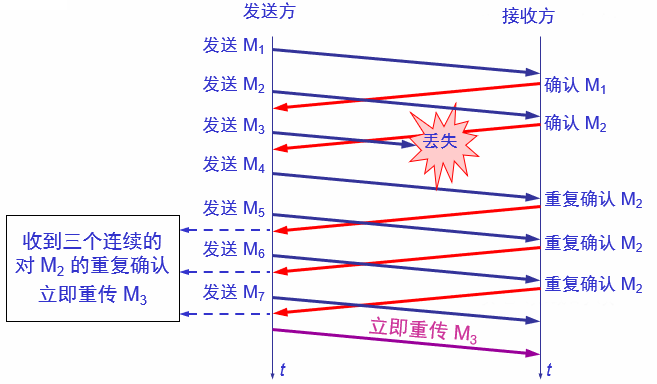
① 加法增大：执行**拥塞避免算法**后，经过一个往返时间，就把拥塞窗口**增加一个 MSS 大小**，使拥塞窗口**缓慢增大**，以**防止网络过早出现拥塞**。

② 乘法减小：只要出现一次超时（网络拥塞），就把**门限值 ssthresh** 设置为当前的**拥塞窗口值乘以 0.5**。

17.2快重传和快恢复

（1）快重传

接收方收到一个**失序的报文段后**就**立即发出重复确认**。让发送方**及早知道**。发送方只要一连收到三个重复确认就应当立即重传对方尚未收到的报文段。



（2）快恢复

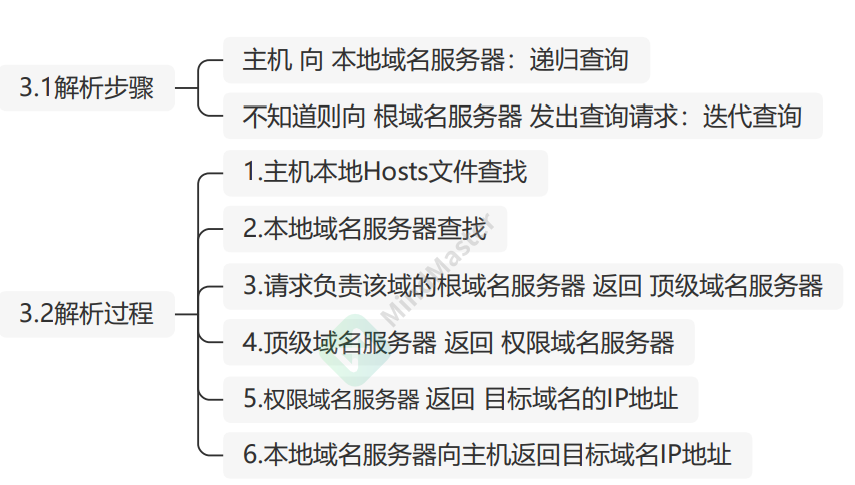
当发送端**收到连续三个重复**的确认时，

执行“乘法减小”算法：把慢开始**门限 ssthresh 减半**。

**拥塞窗口**不设置为 1，而是**设置为ssthresh 减半**，

然后开始执行**拥塞避免算法（“加法增大”）**，使拥塞窗口缓慢地**线性增大**。

18.DNS过程（2021）



线程和协程的区别与联系

**区别：**

执行单元：

线程： 是操作系统调度的最小执行单元，由操作系统进行管理和调度。

协程： 是在用户空间进行调度的执行单元，由程序员显式控制。

调度方式：

线程： 由操作系统进行抢占式调度，可以被中断和切换。

协程： 由程序员控制，在需要时显式地挂起和恢复。

并发性：

线程： 线程可以并行执行，由操作系统管理多个线程的调度。

协程： 协程一般在单个线程内执行，通过非抢占式调度实现协作式并发。

状态保存：

线程： 线程的状态由操作系统负责保存和恢复。

协程： 协程的状态由程序员显式保存和恢复，通常保存在栈上。

**联系：**

并发编程：

无论是线程还是协程，都是为了实现并发编程，让程序能够有效地处理多个任务。

异步编程：

协程常用于异步编程，通过非阻塞的方式处理I/O等操作，提高程序的性能。

资源消耗：

协程通常消耗的资源较少，因为它在用户空间进行调度，减少了操作系统线程切换的开销。

可读性和控制：

协程通常具有更好的可读性，因为它在编写代码时更接近自然的同步写法，而不需要考虑锁和线程间通信的复杂性。

程序员对协程的控制更细粒度，可以手动控制协程的执行顺序。

**总体而言**

线程更适用于 CPU 密集型任务

协程更适用于 I/O 密集型任务，以及需要更好的可读性和控制的情况。

**1.** **阐述计算机网络体系结构的基本思想和主要特点。**

1.1基本思想：分层思想

每一层都直接使用内层提供的服务，并完成其自身确定的功能，然后向外层提供“增值”后的更高级的服务。

1.2主要特点

（1）耦合度低、独立性强：上层可以直接调用获得下层服务

（2）适应性强：只要服务接口不变，内部可以任意改变

（3）易于实现和维护：将复杂问题拆解为简单问题

**2. 结合定义，谈谈你对计算机体系结构的认识。**

2.1定义

用分层研究方法定义的网络通信各层的功能、协议和接口的集合

2.2一些概念

（1）实体：可以发信息的硬件、软件进程

（2）接口Service Access Point：相邻层的交互通道

（3）（本层）协议：控制对等层数据交换的规则、约定、标准

（4）（相邻层）服务：下层向上层提供的功能

2.3认识

计算机体系结构通过分层过后，各层依据任务实现相应功能，对等层间按照协议交流，层级间通过接口交互。

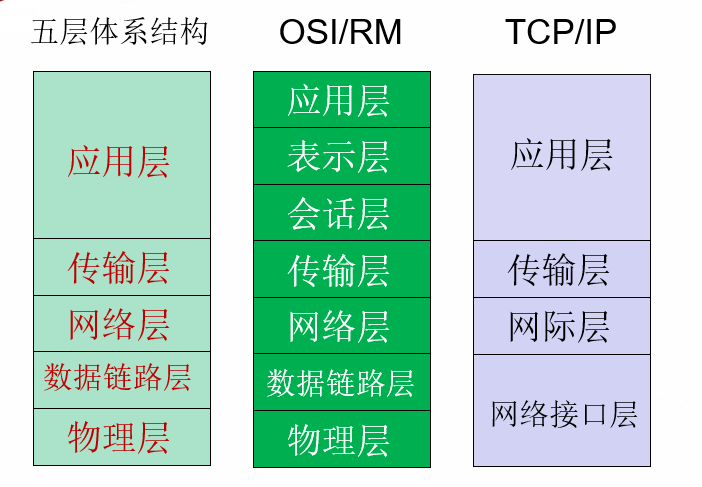
**3. 阐述你对OSI/RM、Internet模型和五层结构模型的认识。**

3.1三种模型对应特点

（1）OSI/RM：概念清楚、理论完整

（2）Internet模型（四层）：简单易用

（3）五层结构模型：结合优点，仅供学术研究，没有实际意义



3.2 OSI 模型

①应用层（最高）

任务：为用户应用进程提供网络通信服务

服务：完成和实现用户请求的各种服务

功能：在用户应用程序与网络间：作为接口，使能交互。

②表示层

任务：使应用程序能够理解数据报文的真实含义

服务：数据加密、压缩、格式转换等服务。

功能：对数据进行编码、编译、加密、转换。

③会话层

任务：建立、管理、维持、协调会话（不同实体之间的表示层的连接）。

服务：

会话连接的建立和释放

通信方式管理（半双工、全双工、优先级）

会话单元分割和主同步

数据传输异常中断恢复

功能：建立、保持、结束会话

注：宏观连接、较长、不易感知中断

④传输层（通信子网与资源子网的分界层）

任务：提供端到端的可靠的透明数据传输，使上层不必关心实现细节

服务：提供面向连接的和面向无连接的传输服务。

功能：

Ⅰ传输连接管理（包括连接建立、维护和拆除等）

Ⅱ传输差错处理（包括差错控制、流量控制和数据重发等）

注：微观连接、较短、容易感知断开。

⑤网络层（通信子网的最高层）

任务：

Ⅰ将网络地址翻译成对应的物理地址；

Ⅱ通过路由选择算法为分组通过通信子网选择最适当的路径。

服务：提供基于逻辑地址的寻址和不同网络间的路由选择

功能：逻辑地址寻址、路由选择、流量控制

⑥数据链路层

分为介质访问控制（MAC）和逻辑链路控制（LLC）两个子层。

任务：

ⅠMAC是解决共享型网络中的信道竞争问题，完成网络介质的访问控制。

ⅡLLC的主要任务是建立和维护网络连接，执行差错校验、流量控制和链路控制。

服务：为上层提供可靠、无差错的节点间数据传输链路。

功能：物理地址寻址、数据的成帧、流量控制、差错控制。

⑦物理层（最底层）

任务：

Ⅰ规定激活、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，为上层协议提供一个传输数据的物理媒体。

Ⅱ在物理介质上正确地、透明地传送比特流，即也就是数模转换与模数转换。

服务：为上层提供正确、透明的结点间数据传输介质。

功能：

Ⅰ实现相邻计算机节点之间比特流的透明传送

Ⅱ屏蔽具体传输介质和物理设备的差异

Ⅲ使数据链路层不必关心网络的具体传输介质。

3.3 Internet 模型（TCP/IP）模型

3.3.1分层介绍

（1）应用层

对应OSI的应用层、表示层和会话层

（2）传输层

TCP：可靠，面向连接；报文以字节流形式发送

UDP：不可靠，无连接协议

（3）网际层

IP：①Internet 体系结构的核心协议。

②把IP分组以数据报方式发送。

（4）网络接口层

对应OSI的物理层和链路层

3.3.2 Internet模型特点

（1）协议标准是完全开放的，免费使用；独立于特定的计算机硬件与操作系统。

（2）独立于网络硬件系统，可以运行在广域网，更适合于互联网。

（3）网络地址统一分配，网络中每一设备和终端都具有一个唯一地址。

（4）高层协议标准化，可以提供多种多样可靠网络服务。

**1. 用自己的语言，总结传输层的作用、位置、主要功能和主要服务。**

（1）作用

在网络通信中，传输层有着**承上启下**的作用：

实现了一种将**面向应用(进程)**的通信需求转变为**面向网络设备**的通信服务。

（2）位置

处于**面向应用部分的最低层**和**面向通信部分的最高层**。

（3）主要功能

① 为运行在网络边缘的不同主机上的各应用程序之间**提供通信服务**。

传输层是**面向应用/进程**（不是主机）的网络通信服务。

是一种真正的**端到端**的通信服务。

传输层通信为**不同进程**之间提供了**逻辑通信**服务。

② 在**应用层和网络层**之间**充当复用器**的作用。

各种各样的网络应用，通过传输层的端口被提交统一的网络层，

并由IP协议用统一的方式将它们发送到网络中。

③ 对高层应用**屏蔽了通信（低层)的细节**

无需过多考虑各种通信因素对网络通信过程本身的影响。

④ 提供**端到端**之间的**无差错保证**

弥补网络层提供服务的差异和不足。

（4）主要服务

**① 面向连接**的服务：TCP服务

通信可靠，对数据**有校验和重发**

实现**复杂**，代价较大

通信**速率相对较低**。

如TCP/IP模型中应用层协议**FTP、Telnet**等

**② 面向无连接**的服务：UDP服务

对数据无校验和重发

实现简单

通信速率高

如TCP/IP模型中应用层协议SNMP、DNS等

**2. 在传输层中，端口、传输层地址和套接字分别是指什么？相互之间有什么关系？**

2.1端口、传输层地址和套接字

（1）端口

协议端口号(protocol port number)，简称为**端口**(port)

虽然**通信的终点是应用进程**，但我们可以把**端口**想象是通信的终点

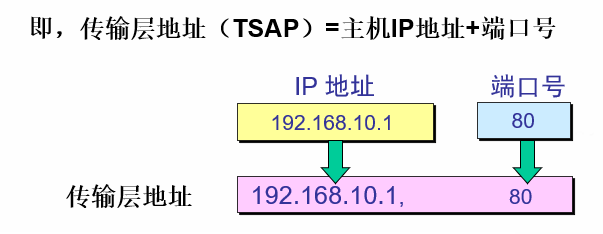
**硬件端口**：不同硬件设备进行交互的接口。

**软件端口**：应用层与传输层协议交互的一种接口。

**端口号**：16 位，只具有**本地意义**，只是为了标志本计算机应用层中的各进程。不同计算机的相同端口号之间无关联。

（2）传输层地址

当两个不同主机的两个进程需要通信时，必须指明对方是哪一个进程，这个标记称为传输服务访问点TSAP



（3）套接字

① 描述

半相关：Socket={协议，本地地址，本地端口}

完整：{协议，本地地址，本地端口，远程地址，远程端口}

② 地位

上联应用进程，下联网络协议栈，

是**应用程序**与**网络协议栈**进行**通信的接口**、**交互的接口**。

③ 分类

1、流式套接字(Stream Socket)

提供**面向连接**、可靠的数据传输服务

数据**无差错、无重复**的发送,且**按发送顺序**接收。

内设**流量控制**，避免数据流超限;

数据被看作是**字节流**，无长度限制。

2、数据报式套接字(Datagram Socket)

提供**无连接**服务。数据包以**独立包**形式发送

**不提供无差错**保证，数据可能**丢失或重复**，并且**接收顺序混乱**。

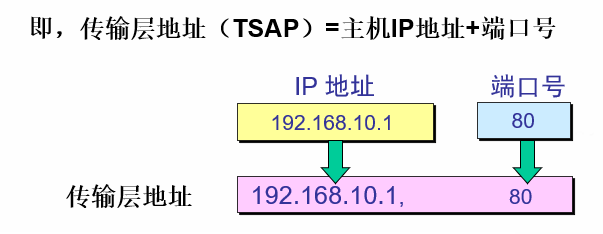
3、原始套接字(Raw Socket)

原始式套接字允许对**较低层次的协议**，如IP、ICMP**直接访问**，

用于**检验新的协议的实现**。

2.2关系

（1）传输层与端口



（2）套接字与端口

半相关套接字：Socket={协议，本地地址，本地端口}

完整套接字：{协议，本地地址，本地端口，远程地址，远程端口}

**3. 什么是UDP协议？有什么特点？**

3.1用户数据报协议UDP

UDP 只在 IP 的数据报服务之上增加了端口的**功能和差错检测**的功能。

3.2UDP 的主要特点

（1）UDP是**无连接**协议：在发送数据之前不需要建立连接。

（2）UDP使用**尽最大努力交付**，不保证可靠交付，同时也**不使用拥塞控制**。

（3）UDP是**面向报文**，没有拥塞控制，很适合多媒体通信的要求。

（4）UDP支持一对一、一对多、多对一和多对多的交互通信。

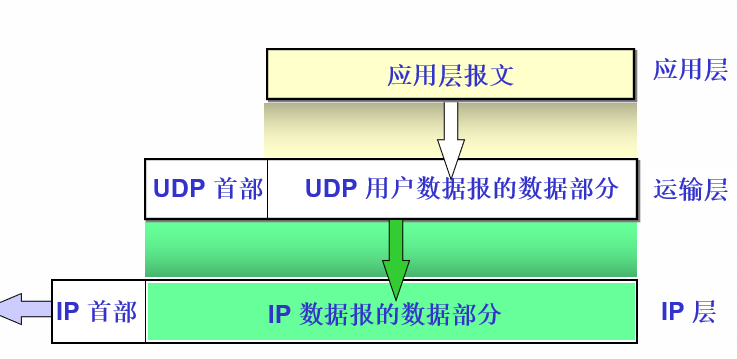
（5）UDP 的**首部开销小**，只有 8 个字节。

（6）发送方 UDP 对应用程序交下来的报文，在**添加首部后**就向下交付 IP 层。UDP 对应用层交下来的报文**，既不合并，也不拆分，而是保留这些报文的边界**。

（7）**不管**应用层给 UDP **多长**的报文，UDP 都**照样发送**，即**一次发送一个报文**。

（8）**接收方** UDP 对 IP 层交上来的 UDP 用户数据报，在**去除首部**后就**原封不动**地交付上层的应用进程，一次交付一个完整的报文。

（9）**应用程序**必须选择**合适大小**的报文。



**4. 在TCP报文段的首部中，你认为哪些字段非常重要？原因是什么？**

（1）**源端口和目的端口**

**端口**是传输层与应用层的服务接口。

传输层的复用和分用功能都要通过其实现。

（2）**序号字段**

TCP 连接中传送的数据流中的**每一个字节**都编上**一个序号**。

序号字段的值则指的是**本报文段**所发送的数据的**第一个字节的序号**。

（3）**确认号**字段

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。

（4）**数据偏移（首部长度）**

指出 TCP 报文段的**数据**距离起始处有多远。

“数据偏移”的单位是 32 位字。

（5）**保留字段**——保留为今后使用，置为 0。

（6.1）**紧急URG：**有**紧急**数据，应**尽快**传送(相当于**高优先级**的数据)。

（6.2）**确认ACK**

（6.3）**推送 PSH (PuSH)**：**尽快地交付**，而**不等缓存都填满**了后再向上交付。

（6.4）**复位 RST (ReSeT)**：表明 TCP **连接**中出现**严重差错**（如由于主机**崩溃**或其他原因）

必须**释放连接**，然后再**重新建立**运输连接。

（6.5）**同步 SYN**：表示这是一个**连接请求**或**连接接受**报文。

（6.6）**终止 FIN (FINis)**：**数据已发送完毕**，并要求释放运输连接。

（7）**窗口**

用来让对方设置发送窗口的依据，单位为字节。

（8）**检验和**

检验的范围包括**首部**和**数据**。

在计算检验和时，要在 TCP 报文段的前面加上 **12 字节**的**伪首部**。

（9）**紧急指针**

指出**紧急数据共有多少个字节**（紧急数据放在本报文段数据的**最前面**）。

（10）**选项**：

① **最大报文段长度 MSS**选项

**数据**字段的**最大长度**。

数据字段加上 TCP 首部才等于整个的 TCP 报文段。

② **窗口扩大选项**

其中有一个字节表示移位值 S。

新的窗口值等于TCP 首部中的窗口位数**增大到(16 + S)**

相当于把窗口值向**左移动 S 位**后获得实际的窗口大小。

③ **时间戳**选项，其中主要是两个字段

**时间戳值**字段（4 字节）和**时间戳回送回答**字段（4 字节）。

④ **选择确认**选项——**有选择的确认收到**的某些报文，以便**提高效率**。

（11）**填充**：使整个首部**长度是 4 字节的整数倍**。

**5. 简述TCP连接建立和释放的基本过程**

5.1连接建立——三次握手

第一次: 客户端发出**同步请求**报文。

第二次：服务器发出**同步确认**报文。

第三次: 客户端发出**确认**报文。



5.2连接释放——四次挥手

第一次：客户端发出一个**连接释放**报文。

第二次：服务器向客户端发出**确认**报文。

第三次：服务器发出**连接释放确认**报文。

第四次：客户端发出**确认**报文。



MSL（Maximum Segment Lifetime）任何报文在网络上的存在的最长时间超过这个时间报文将被丢弃。

RFC 793规定MSL为2分钟，不同的操作系统在实现时各不相同，常用的有30秒、1分钟、2分钟等。

**6. 在TCP中有哪些基本的计时器？这些计时器在TCP协议中各自发挥什么样的作用？**

TCP要保证正常工作，至少应该有四种计时器：

**重传**计时器、**持久**计时器、**保活**计时器、**时间等待**计时器。

（1）重传计时器

当发送方发出数据报文后即启动该计时器（一般为60秒）

在设定时间截至之前收到确认报文，则传输成功，**撤销**计时器；

否则，传输失败，**重新发送**数据报文。

（2）坚持计时器

① 背景

接收方发出一个**零窗口确认报文**，发送方立即停止发送数据报文。

接收方在发出**非零窗口确认报文**后即进入等待。

如果此确认报文在中途丢失，在通信会陷入**死锁**。

② **发送方收到一个零窗口确认时**，即启动坚持计时器。

若在设定时间到达之前收到**非零窗口确认**报文，则**撤销**。

若设定时间到达仍未收到接收方的非零窗口确认报文，则发送方发出一个只有**一个字节**数据的**探测报文**并**消耗一个序号**，但该序号会被确认报文所忽略。

③ 若在截止时间后没有收到接收方的非零窗口确认报文，则发送**另一个探测报文**，并**加倍设置**坚持计时器值，如此反复直到收到接收方的非零窗口确认报文为止。最大值限制（门限值，通常为60秒）

（3）保活计时器

用于判断两个TCP端点之间**长久的连接**是否**正常**。

当建立了TCP连接后，保活计时器被**激活**，并设置计时值（2小时）。

每当服务器**收到**来自客户端的报文，即**重置**计时器。

**未收到**，服务器发出**探测报文**，并每隔75秒发送一个探测报文。

**10个**探测报文后依然没有得到，**强制关闭**这条连接。

**注：**客户端已经崩溃并重启

在收到服务器的探测报文后，客户端将返回一个RST报文。

服务器在收到RST报文后即关闭连接。

（4）时间等待计时器/2MSL计时器

目的是为了能够正常关闭服务端的连接！

**7.什么是停止等待协议？为什么说在有流控的停止等待协议中可能会出现死锁？如何破除死锁？**

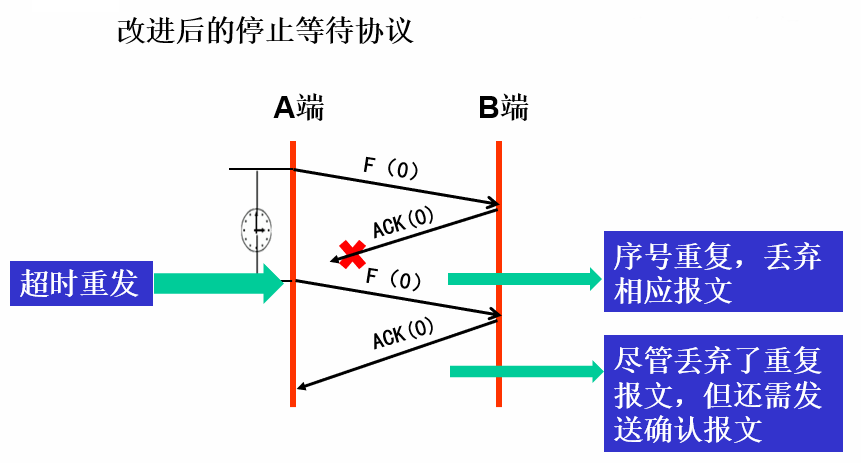
7.1停止等待协议

在有差错控制、报文丢失的流控的协议基础上，进行以下改进：

（1）编号重复检查，丢弃编号重复的报文，从而解决重复报文的问题

在丢弃重复报文的同时，也要发出针对该重复报文的确认报文

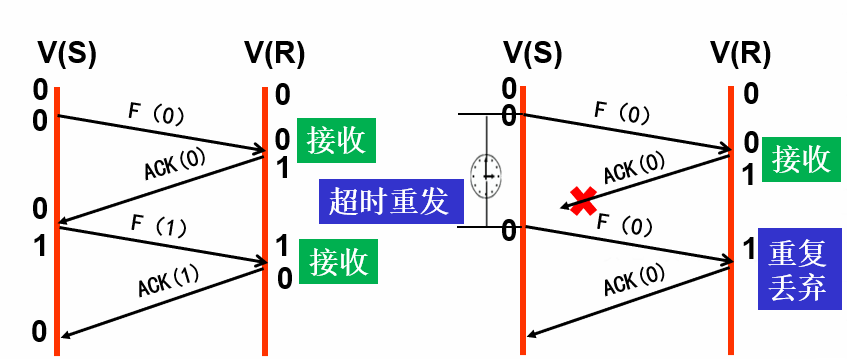
（2）只发送肯定应答报文，添加确认信息，有错误的等着超时重发



（3）改进后的停止等待协议的报文序号

序号会消耗通信资源，序号越长消耗的资源越多。

只需要区分前后报文，在停止等待协议中报文序号只需要一位即可。



7.2死锁原因

无论是数据报文（发送端发送）还是应答报文（接收端发送），都有可能在传输过程中**丢失**。

① 若**数据报文丢失**，接收端未收到数据报文，等待发送端发送。

② 若**应答报文丢失**，发送端未收到应答报文，等待接收端发送。

上述两种情况，都会使发送端等待接收端发送应答报文，而接收端等待发送端发送数据报文，从而造成**死锁**。

7.3死锁解决

设置**重发计时器**。

发送端在发出报文后即启动重发计时器，若在设置的时间tout内未收到接收端的应答报文，则认为报文已丢失，从而重发报文。 通常，Tout略大于报文平均往返时间。

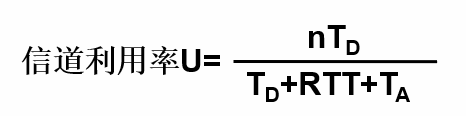
**8. 什么是连续ARQ协议？为什么说连续ARQ协议可以大幅度的提高信道利用率？**

8.1连续ARQ协议

在停等ARQ协议基础上采用**流水线**传输解决信道利用率低的问题。

8.2提高信道利用率的原因

（1）采用流水线连续发送，具有很高的信道利用率



（2）代价：**n个缓冲区**暂存报文，以备重发。

① 当n达到一定值后即不能再增大，因为**TD有相对固定的值**。

② 信道利用率不会超过100%

因此在n达到某个值之后，信道利用率不可能继续提高。

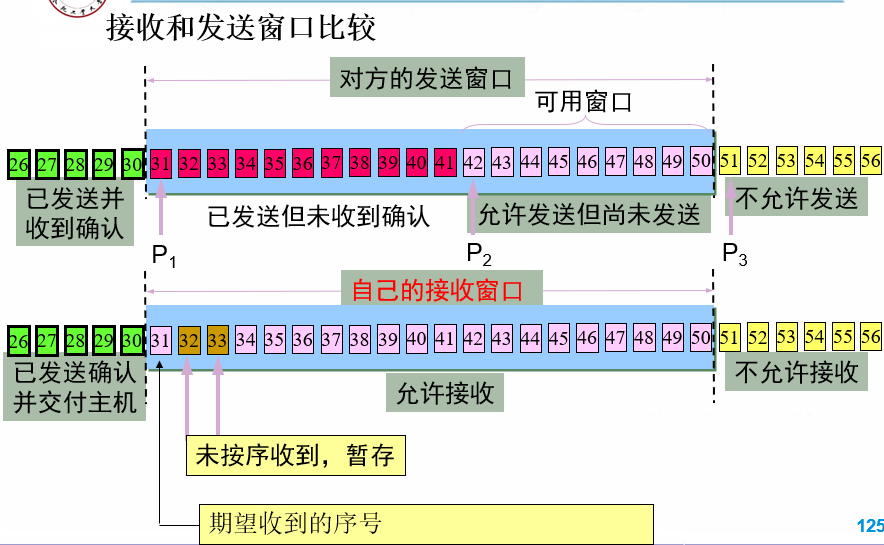
**9. 什么是滑动窗口？举例说明其基本运行过程。**

9.1滑动窗口

滑动窗口（Sliding window，SW）是一种**流控方法**，用于约束发送方可发送报文的数量。窗口是指发送方最多可发送未被确认报文的数量，而滑动则是指每收到一个确认报文，窗口可向前滑动一个报文，从而纳入新的待发送的报文。

9.2举例





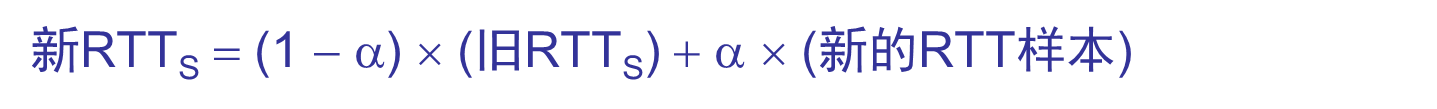
**10. 在确定超时重传时间时，RTT、RTTs和RTTD各自发挥着什么作用？**

10.1 RTT在报文重传过程中的重要性

**RTO 小**于 RTT，则会造成很多**不必要的重传**；

**RTO 远大**于 RTT，则会**降低整体网络利用率**。

10.2加权平均往返时间RTTS



若 α 更接近于**0**，表示 RTT 值更新较**慢**。

若选择 α 更接近于 **1**，则表示 RTT 值更新较**快**。

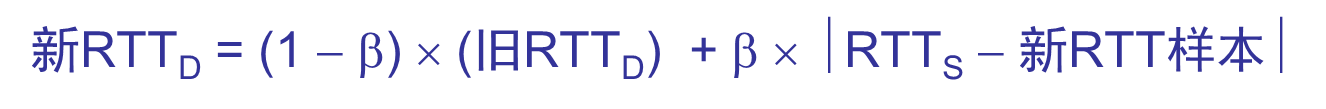
RFC 2988 推荐的 α 值为 **1/8，即 0.125**。

10.3偏差加权平均值RTTD



其中：RTTD 是 RTT 的**偏差加权平均值**。

第一次测量，RTTD = RTT / 2



推荐值是 **1/4，即 0.25**。

**11. 什么是RTO的指数避退？有什么作用？**

（1）**超时而发生连续多次重传**时，**超时阈值**遵循指数避退：

多次超时重传中，下一次是这次的2倍，2为退避因子

直到收到重传数据包的应答，RTO退避因子回复为1。

（2）作用：为了当网络处于**无法快速交付**数据报文状态时**减小网络负担**。

**12. 在TCP中，流量控制是怎么实现的？**

（1）滑动窗口实现

（2）持续计时器

（3）传输效率

任何**小于MSS**的数据报文都被视为小数据报文。

**小数据报文**大量的出现会造成广域网上的**拥塞**。

即设法**增加发送的数据规模**不但可以**提高信道利用率**，而且也有助于**减少广域网的拥塞**。

（4）糊涂窗口综合症：接收端和发送端**速率不匹配**时发生。

① 描述

这个问题可以归结为**小包**的问题：由于发送端和接收端上的**处理不一致**，导致网络上**产生很多的小包**。在滑动窗口机制下，如果发送端和接收端**速率差异很大**，就会产生这种比较犯傻的状态：发送方发送的数据**有效数据很少**。

② 解决

法一：Nagle算法——治疗“发送端糊涂”

法二：Clark、延迟ACK——治疗“接收端糊涂”

（4.1）Nagle算法

尽可能**避免发送小数据报文**，一个TCP连接上**最多只能有一个**未被确认的小数据报文。

在小数据报文的确认报文到达之前不发送其他小数据报文。并且**持续接受小数据报文**，并在**确认到来时以一个较大的报文发送**出去。

当到达的数据已达到发送窗口大小的**一半或报文段最大长度**时，就**立即发送**一个报文段！

（4.2）Clark方法

只要有数据**到达就发送确认**，但**宣布的窗口大小为零**，

直到或者缓存空间**已能放入具有最大长度的**报文段，或者缓存空间的**一半已经空**了。

（4.3）延迟ACK

延迟ACK就是**累积确认（+捎带）**。

如果这段时间内有数据发送到对端，则**捎带发送ack**；

如果在延迟ack定时器触发时，则立即单独发送。

（4.4）当Nagle遇上延迟ACK

发送端产生了多个小数据报文需要向接收端发送

第一个小数据报文到接收端后，因为延迟ack而没有及时ack。

发送端等待对接收端对第一次数据确认。

结果：网络延迟。

**13. 简述慢启动和拥塞避免的基本概念**

（1）发送方维持**拥塞窗口 cwnd**

拥塞窗口的大小**取决于网络的拥塞程度**，**动态变化**。

原则：

网络**没有拥塞**，拥塞窗口就再**增大**一些，以便**把更多的分组发送出去**。

网络**出现拥塞**，拥塞窗口就**减小**一些，以**减少注入到网络中的分组数**。

发送窗口 < 拥塞窗口

（2）慢开始算法的原理

**刚刚开始**发送时设置拥塞窗口 **cwnd = 1**，即一个MSS的数值。（一倍）

在**每收到**一个**确认**后，将**拥塞窗口加 1**，即增加一个 MSS 的数值。（加倍）

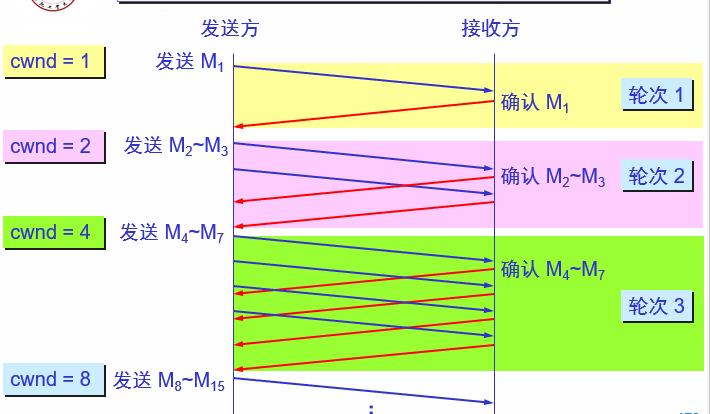
（3）传输轮次

每经过**一个传输轮次**，拥塞窗口 cwnd 就**加倍**。

一个传输轮次所经历的时间其实就是**往返时间 RTT**。

例如，**拥塞窗口 cwnd = 4**，这时的往返时间 RTT 就是发送方**连续发送 4 个报文段**，并**收到这 4 个报文段的确认**，**总共经历的时间**。

（理解：火车完全过山洞）



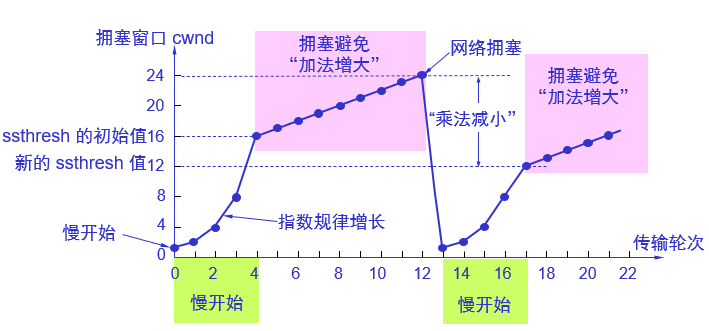
（4）门限状态变量（ssthresh）

当 cwnd < ssthresh 时，使用**慢开始算法**。

当 cwnd > ssthresh 时，**改用拥塞避免算法**。

**拥塞避免算法**：

是让拥塞窗口 cwnd **缓慢地增大**，即每经过一个往返时间 RTT 就把发送方的拥塞窗口 cwnd **加 1**，使拥塞窗口 cwnd 按线性规律缓慢增长。



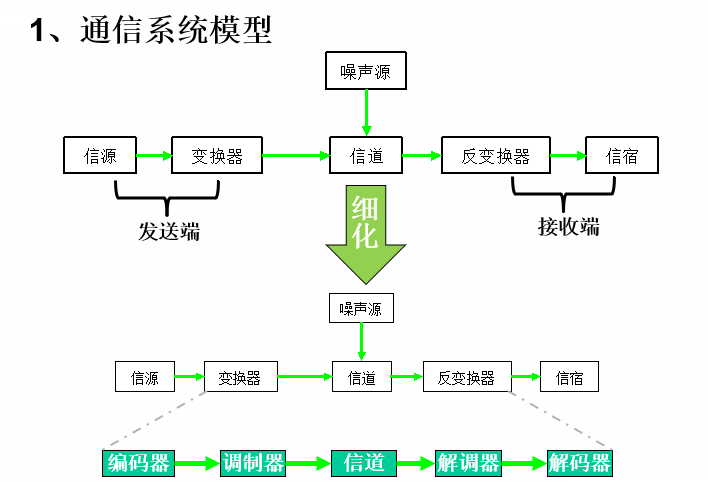
① 加法增大：执行**拥塞避免算法**后，经过一个往返时间，就把拥塞窗口**增加一个 MSS 大小**，使拥塞窗口**缓慢增大**，以**防止网络过早出现拥塞**。

② 乘法减小：只要出现一次超时（网络拥塞），就把**门限值 ssthresh** 设置为当前的**拥塞窗口值乘以 0.5**。

**1.阐述通信系统模型结构、指出各部分的主要作用和功能**

1.1通信系统模型结构

主要包含信源、变换器、信道、噪声源、反变换器、信宿六部分。（如下图）



1.2各部分及主要作用和功能

（1）信源：把各种信息转换成原始物理信号，如计算机，FAX，电话机等。

（2）变换器：将原始物理信号转换成适合在信道上传输的信号，如MODEM,TA,光电转换器等。

（3）信道：传输信号的一条通路。通常一条物理线路可构成一或多个信道。

（4）噪声源：信道自身的噪声以及周围环境对信道的干扰。如热噪声，闪电，强电磁场干扰等。

**2.数据通信方式有哪几种？各有什么特点？**

2.1串行/并行方式

（1）串行：各个比特位按一定顺序依次发出

特点：信道数1，成本低

用于远程通信

（2）并行：各个比特位使用多信道同时发出

特点：信道数多个，传输速率是串行n倍，成本高

用于近距离通信

2.2同步/异步方式

通信基本要求：接收方知道接收的每个比特位的开始和持续时间

（1）异步

①原理

每个字节作为一个单元独立传输，字节间传输间隔任意

每个字节标志开始结尾：前加1个起始位、后加1-2个停止位

②特点

- 起始结束标识位

- 标识位兼做线路两端同步时钟，不需要额外时钟

- 间隔任意

- 速率低，适合误码率高、对速率要求低的传输

（2）同步

①原理

对一组字符组成的数据块进行同步

在数据块前面加上特殊模式的位组合

通过位填充、字符填充技术，保证数据不与同步字符混淆

e.g.转义 /000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sync | Sync | Data Block | BCC | EOB |

Sync同步符号、BCC 块检查符号、EOB 块结束符号

②特点

- 开销小、效率高

- 适合高速数据传输

- （缺点）数据块有一位错误，整体重新传输

2.3单工/半双工/双工模式

（1）单工：单向传输，如广播

（2）半双工：可以双向，但同一时刻只允许单向，如无线对讲机

（3）全双工：允许同时双向传输，如计算机通信

**3.阐述信源编码的目的、作用和主要过程。**

3.1信源编码的目的

信源编码将复杂信源信息用数字信号表示，是一个信息压缩过程。

保证不失真的前提下，信源符号序列->最短码字序列。

目的：模拟信号数字化，方便处理传输

3.2信源编码的作用

在保证通信质量的前提下，尽可能通过对信源的压缩，提高通信时的有效性。

3.3信源编码的主要过程

（1）抽样：模拟信号离散化

（2）量化：无限幅值有限化

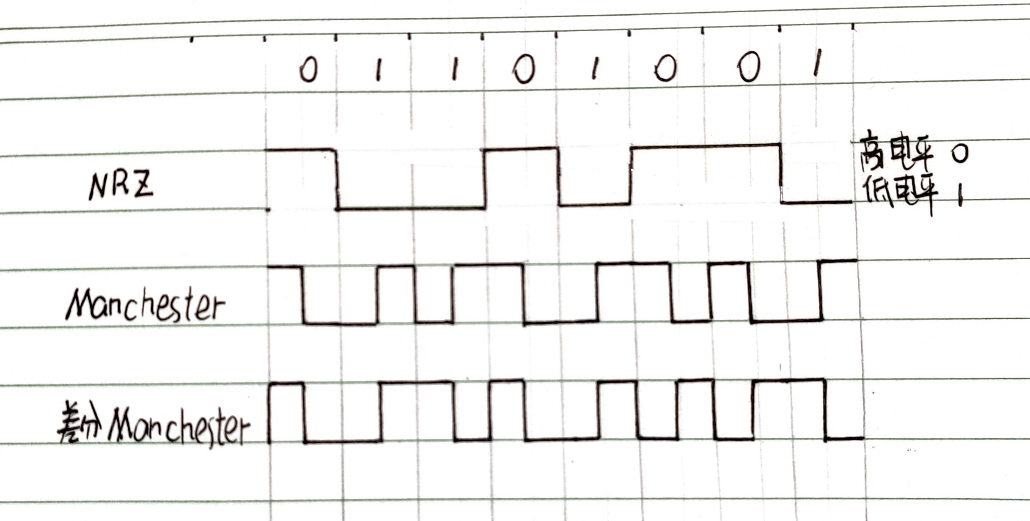
（3）编码：量化结果码组化

**4.阐述数字数据编码的目的和作用，并对二进制字符串“01101001”分别给出非归零、曼彻斯特和差分曼彻斯特编码。**

4.1数字数据编码的目的和作用

数字数据编码也称为信道编码。在数字信号在传输中由于各种原因，使得在传送的数据流中产生误码。信道编码就是码流进行相应的处理，使系统具有一定的纠错能力和抗干扰能力。

4.2 01101001



**5.阐述多路复用的目的和作用，分别说明频分、时分和码分复用的原理和特点。**

5.1多路复用的目的和作用

一路信号独占一个信道对于充分发挥信道利用率、扩大通信的普及范围是极为不利的，也是不可想象的。多路复用将多路信号复用在一条信道上，使一条信道能够同时传输多路数据信号。

5.2原理特点

5.2.1频分复用

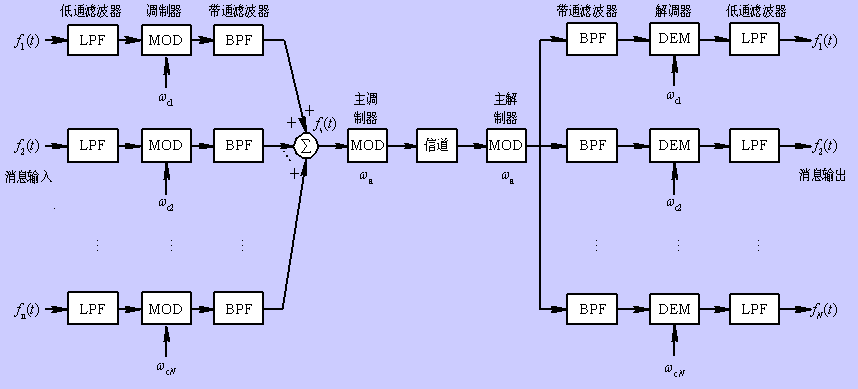
（1）基本思想

信号带宽被分成若干互不重叠的频段

每路信号占用其中一个频段

接收端采用适当带通滤波器把多路信号分开、恢复

（2）实现原理



注：每路信号要调制到各自子载波频率上

两个子载波频率间要有足够保护频带

（3）特点

所有信号带宽和 < 信道总带宽 （相邻子信道预留隔离信道）

各路信号并行传输，无需考虑时延

每个信号在全部时间只占部分频率谱

5.2.2时分复用

（1）基本思想

单路信号离散相邻脉冲间的空隙，插入若干其他信号

只要各个信号时间上不重叠并能区分开

模拟或数字信号一般采用时分复用

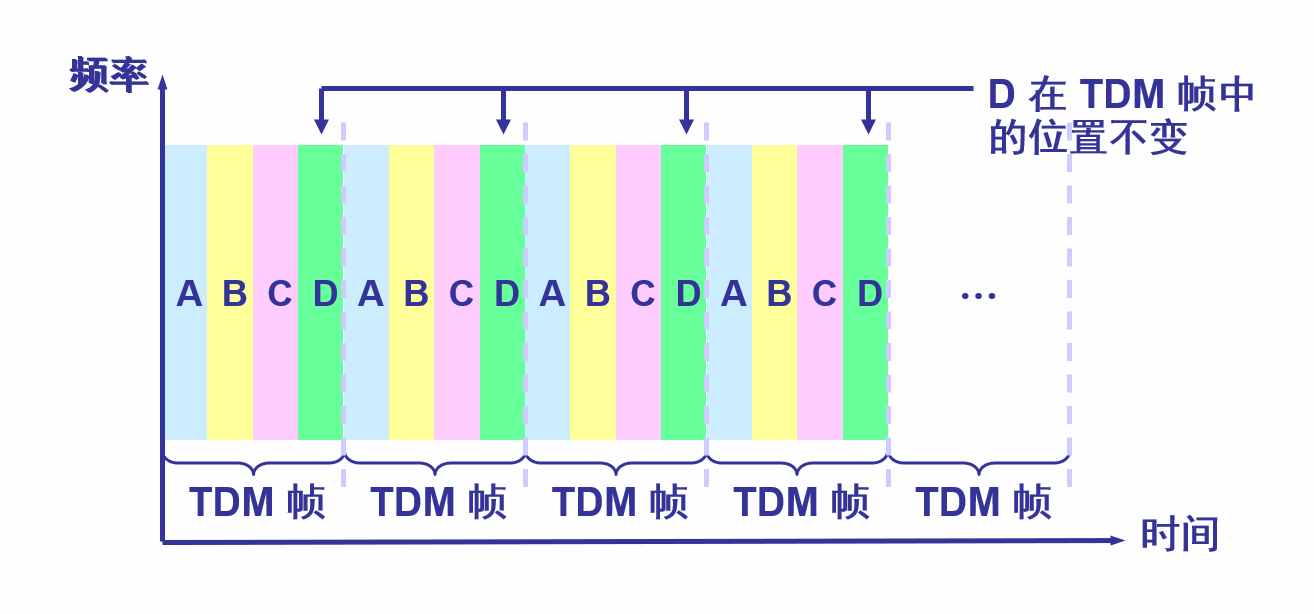
（2）实现原理

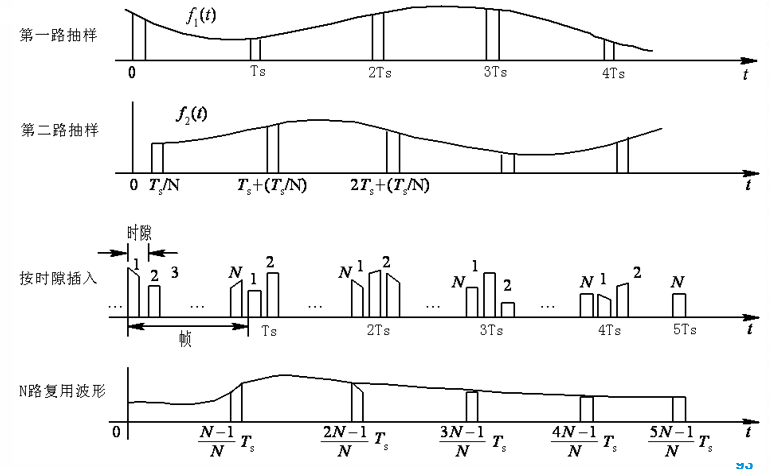
把时间划分为等长时分复用帧（TDM帧）

每个信号占用固定序号的时隙（周期性出现=TDM帧长度）

TDM信号又称作等时信号

各用户在不同时间占用同样频带宽度（所有）





（3）特点

TDM帧宽度一定，脉冲宽度越小，信号数量越多

各信号在各时隙占用全部频率资源，各时隙有保护时隙

时隙预先规划且固定不变，在时隙就算空闲也不分配其他

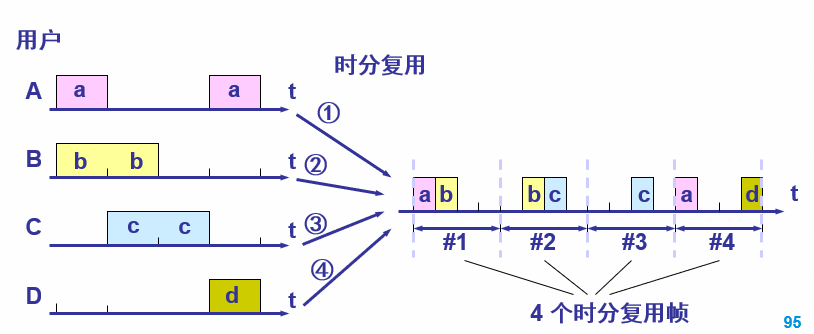
收发两端严格同步

（4）问题

数据传输有突发性、阵发性，

在时隙就算空闲也不分配其他

解决：统计时分多路复用



5.2.3码分复用

（1）基本思想

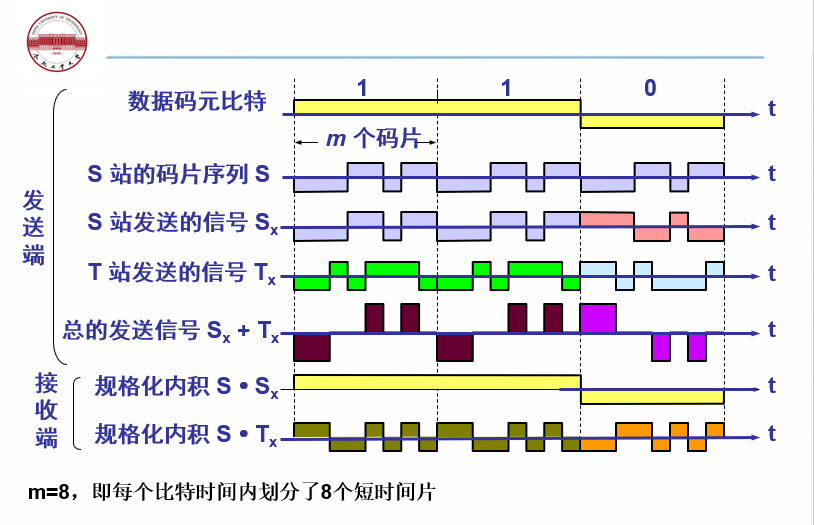
- 每个比特时间划分为m个短时间片——码片（chip）

- 每个用户被分配一个m位（128或64，最大264个）码型，各个码型成正交关系，不会干扰

- 要发1——发自己的码型、要发0——发自己码型的反码

- 公共信道上的码型叠加在一起与信道白噪声相似，破译困难、安全性高

（2）实现原理



（3）特点

- 每路信号在全部时间内使用全部带宽资源。

- 只要码片足够长，除非拥有相应的码型，

否则无法从“白噪声”中获得有价值的信息。

**1、数据链路层提供的哪些服务？具有哪些主要功能？**

1.1提供服务

（1）无确认无连接

①- 无需连接、无需释放

- 只发送，其他不关心

- 只接收，不确认。有错丢弃，不通知源结点。

- 传输可靠性由高层完成

②- 用于较可靠网络，以太网的主要方式

（2）有确认无连接

①- 按顺序对每个数据帧编号发送

- 收到向源结点发出确认帧

- 规定时间未收到确认帧，判定丢失重发

②- 用于不可靠信道，如无线通信系统

（3）有确认有连接

①- 连接、传输、释放

- 编号发送

- 收到确认

- 有数据链路建立、数据传输、数据链路释放的三个阶段

②- 用于广域网的通信子网

1.2主要功能

（1）拆分成数据帧，按顺序发送

（2）保证可靠传输

①一种合适的帧结构——以便拆分上层数据

②数据识别机制——确保所有数据得到传输

③检错纠错机制——发现并解决传输中的错误

④流量控制机制——维持有序通信

⑤获得发送数据权的判断和决策机制

**2、总结数据链路层的基本问题及其解决办法**

2.1封装成帧

（1）问题

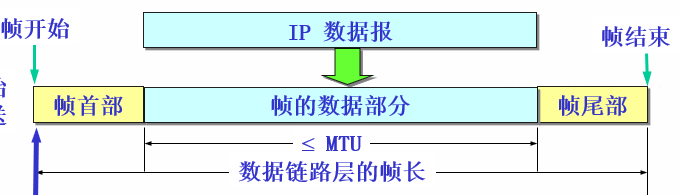
为了方便**接收方数据链路层**能以帧为单位处理数据，

发送方必须正确识别帧的起始和终止，进行帧定界。

（2）定义

在网络层报文的前后分别添加首部尾部，构成帧。

首部尾部重要作用之一：帧定界。



（3）解决方法：帧定界符

①数据由ASCII码组成，使用控制字符作为帧定界符。

②SOH（0x01）表示帧首部开始；EOT（0x04）表示帧结束

③发送中断，重新发送。通过SOH和EOT匹配判断不完整丢弃。

2.2透明传输

（1）问题

帧定界符出现在数据中

（2）定义及特点

透明传输：无论数据是什么样的比特组合，都能够在链路传送。

若存在比特组合与控制信息相同，发送方采取适当措施，使解除误解。

- 被传数据对数据链路层不可见、透明。

- 透明传输相当于黑盒，进来什么就出去什么。

（3）解决方法

①字节填充法

- 在控制字符SOH、EOT前面加转义字符ESC（0x1b）

- 接收端的数据链路层在数据送往网络层之前删除插入的转义字符

- 如数据包含转义字符，再插转义字符，删除前一个。

②比特填充法

- 适用于连续n个0或1作为帧定界符时

- 发送端发现n-1个连续1时立即填入一个0

- 接收端发现n-1个连续1时立即删除后面0

2.3差错控制

数据通信中差错不可避免。

（1）原因

①热噪声：传输介质内分子热运动

②\*冲击噪声：外界干扰

（2）特征

①热噪声：干扰幅度小，持续性，对模拟通信影响大

②\*冲击噪声：干扰幅度大，突发性，对数字通信影响大

（3）解决方法

差错控制码，使接收端发现和纠正。

①检错码 ②纠错码

**3、比较分析HDLC和PPP协议的异同**

3.1异

（1）HDLC协议面向比特；

PPP协议面向字符，且PPP协议比HDLC简单

（2）HDLC协议有三种帧类型：信息帧、监督帧（应答帧）、无编号帧

（3）PPP协议分层体系结构

- LCP作用于数据链路层，负责数据链路建立、配置和测试

- NCP作用于网络层和数据链路层之间，根据网络层不同协议互相交换特定的控制报文、配置不同上层协议。

（4）HDLC协议只能封装在同步链路上

PPP协议既支持异步链路，也支持面向比特的同步链路

（5）HDLC协议不支持IP地址协商；

PPP的NCP协议支持协商。

（6）两者运行过程对比

① HDLC三个阶段

建立数据链路连接、传输数据帧、拆除链路连线

② PPP协议运行阶段

- 链路不可用阶段

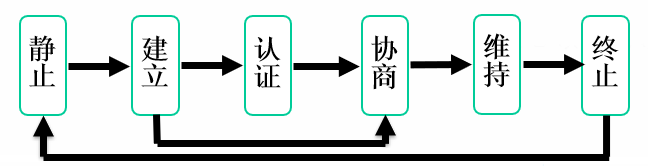
- 链路建立阶段：LCP协商

-（验证阶段）：PAP/CHAP验证

- 网络层协议阶段：NCP协商

- 维持阶段

- 网络终止阶段



3.2同

封装数据格式与HDLC极为类似，即帧格式类似

**4、PPP协议的LCP子协议中有哪几类报文？各有什么作用？**

4.1链路配置报文

（1）- Config-Request：代码为0x01，启用链路以及配置请求报文。

- Config-Ack：代码0x02，收到请求报文后的接受请求应答。

- Config-Nak：代码0x03，收到请求报文后部分拒绝请求应答。

- Config-Reject：代码0x04，收到请求报文后全部拒绝请求应答。

（2）LCP配置选项

① 一个LCP报文可以携带多个TLV组（Type、Length、Value）

② 包括：

-- 最大接收单元：通知对方自己可接收的最大数据包长度

-- 异步控制字符映射：通知对方哪些需要转义

-- 链路鉴权协议：协商认证协议，默认不认证

-- 魔术数：测试链路是否存在环的检测方法。

若本次与上次魔术数相同，则可能存在环；

发送一个携带不同魔术数的Nak进一步检测

-- 协议域压缩：把编号小于256的协议压缩为一个字节（原来两个）

-- 地址和控制域压缩：若没有接收到FF，则进行了压缩

4.2链路终止报文

（1）Terminate-Request报文

主动终止链路的一端持续发送Terminate-Request报文，直到收到一个Terminate-Reply。

（2）Terminate-Reply报文：应答链路终止请求。

当收到一个Terminate-Request报文后，接收方必须回应一个Terminate-Reply报文，同时等待对端先将链路断开后，再完成本端的所有断开的操作。

4.3链路维护报文

两端分别发送Echo Request，对方回复Echo Reply

**5、阐述CSMA/CD协议的基本原理、工作机制和主要特点。**

5.1基本原理

①基本思想

针对纯CSMA在发送数据时无法检测碰撞提出

- 碰撞检测：边发数据边检测信道上信号电压大小

若多个结点同时发送，电压叠加、变化较大

超过阈值则认为发生碰撞，立即停止发送，等待随机时间后重发

- 是以太网的基础。

- 主要贡献：解决了多个结点同时传输时，由于距离等原因无法立即检测到冲突的问题。

②碰撞对数据传输的影响

- 原因：电磁波在信道的传播速度有限性，体现在传输时延

- 导致：同时或相差时间小的多结点发送数据前无法检测，导致碰撞

- 后果：信号失真

③ 可能发生的碰撞种类

- 同时发送导致碰撞

- 不同时发送导致碰撞

信号传输需要时间，发送方数据未送到，接收方发了数据

5.2工作机制

碰撞避退算法：随机等待时间的确定采用 二进制指数退避算法

① 每个站点确定一个基本推迟时间Ti

② 从整数集合{0，1，2，4…，2k-1}中随机抽取整数r

注：k为 min{重发次数，10}

③ 随即等待时间Tw = r \* Ti

④ 若连续16次发送失败，则放弃发送并向高层报告

算法要点：重发次数越多，等待时间越长

5.3主要特点

① 半双工

② 至多2τ就知道是否碰撞

③ 必须在发送完成前检测到碰撞

凡是小于 最小帧长度 都是发现了碰撞而没有发完的

**6、阐述经典以太网的基本概念，比较经典以太网协议与802.3协议在报文格式方面的异同。**

6.1基本概念

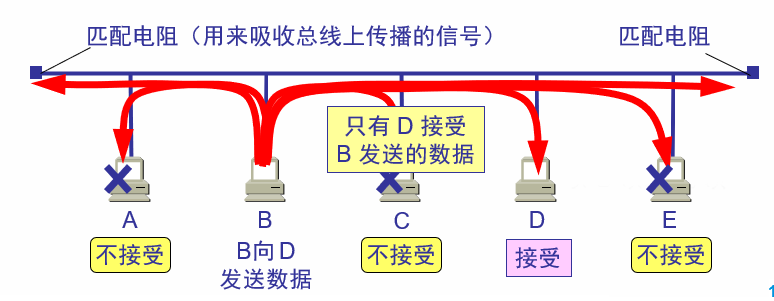
（1）标准

①两个标准：（DEC\Intel\Xerox）DIX Ethernet V2 和 802.3标准

②TCP/IP使用的局域网是DIX Ethernet V2标准。

因此，逻辑链路控制子层LLC（802.2标准）作用不大

（2）最初连接方式：总线型。简单可靠，因为没有有源器件。



-

（3）概念

①基本工作机制：CSMA/CD协议

②传输媒介：粗电缆

单端传输500m，最多连接4台中继器达到2500m

③无连接工作方式

发送不编号也不要求确认：局域网通信质量好

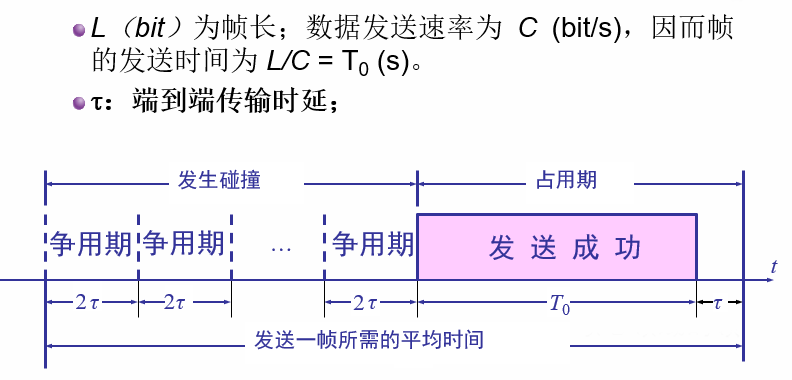
④曼彻斯特编码

⑤争用期 = 2倍往返时延， 51.2μs

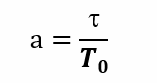
争用期内可发送64字节，若均未碰撞则不再碰撞

最短有效帧长64字节

（4）以太网信道利用率



①占用期：成功发送一个帧所需要时间

②a=传输时延/占用期 

a->0表示一碰撞即可检测出来、立即停止发送，信道利用率很高

a越大，表明争用期比例越大，每次碰撞浪费许多信道资源，信道利用率低

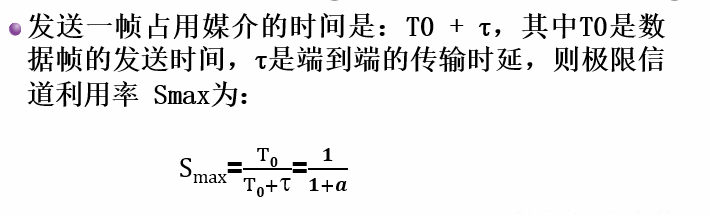
③数据率一定，媒介过长，传输时延会很大

④帧长度不能太短，否则占用期小，导致a很大

（5）最大信道利用率

理想状态：各结点发送数据时都不会产生碰撞

媒介一旦空闲就有结点立即发送数据



6.2比较异同

（1）DIX Ethernet II：也有称为Ethernet V2(ARPA)，以示其来源的正宗性。由DEC、Intel和Xerox三家企业开发的经典以太网，目前的主流。

（2）IEEE 802.3/802.2 LLC：**以Ethernet V2为基础**，除了将原协议**类型字段**替换为**长度字段**外，还**加入802.2 LLC头**用以**标志上层协议**。

（3）IEEE 802.3/802.2 SNAP：IEEE为**支持更多的上层协议而发布**，在**保留LLC头**的同时，新**增加了一个2Bytes的协议类型域**。

（4）RAW 802.3：也称为NOVELL Ethernet 802.3。**仅支持IPX/SPX网络**。

**7、为什么说网桥和交换机隔离了冲突域、扩大了广播域？**

7.1隔离了冲突域

（1）交换机实现**基于自学习算法的帧转发**：

① 网桥收到一帧后先进行**自学习**。查找转发表中与收到帧的源地址有无相**匹配**的项目。

-- 如没有，就在转发表中增加一个项目（源地址、进入的接口和时间）。

-- 如有，则把原有的项目进行更新。

② **转发帧**。查找转发表中与收到帧的目的地址有无相匹配的项目。

-- 如没有，则通过所有其他接口（但进入网桥的接口除外）按进行转发。**（广播）**

-- 如有，则按转发表中给出的接口进行转发。

③ 若**转发表**中给出的接口就是该帧**进入**网桥的接口，

则应丢弃该帧（因为这时不需要经过网桥进行转发）。

（2）**每一台主机都只会收到跟自己有关的数据，不会相互影响**。

7.2扩大了广播域

（1） 网桥工作在**数据链路层**，根据 MAC 帧的目的地址对收到的帧进行**转发**。

（2） 网桥具有**过滤帧**的功能

- 当网桥收到一个帧时，并不是向所有的接口转发此帧，

- 是先检查此帧的目的 MAC 地址，然后再确定将该帧转发到哪一个接口

（3） 利用网桥可以将两个不同的局域网互连

**8、什么是虚拟局域网？为什么说虚拟局域网可以隔离广播风暴和冲突？**

8.1定义

（1）是由局域网中不同网段上的结点所构成的**逻辑网段**

这些结点与其真实的物理位置和真实网段无关。

（2）通常，这些结点具有某些**共同的需求**。

（3）每个 VLAN 的帧中都会插入一个明确的**标识符**，指明发送这个帧的结点所属 VLAN。

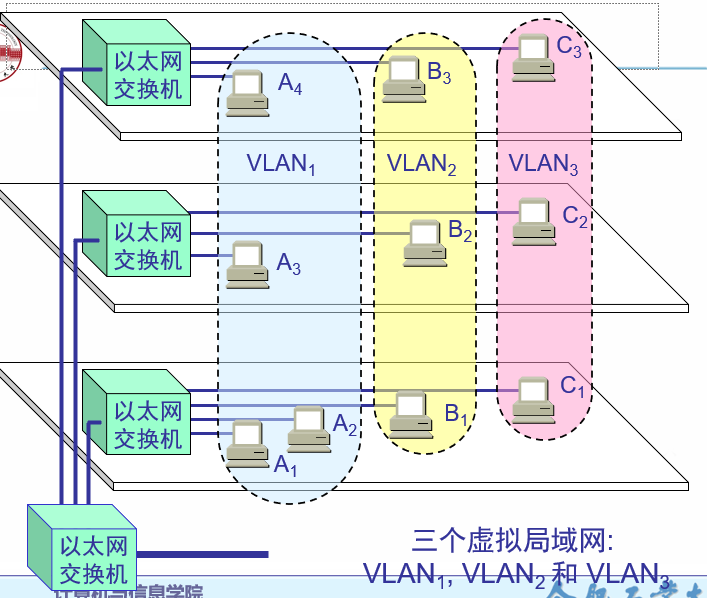
（4）VLAN**只是一种服务**，而不是一种新型局域网。

（5）和真实局域网一样，VLAN也**能隔离广播风暴和冲突**。

8.2原因

虚拟局域网**限制**了**接收广播信息的结点数**，

使网络不会因传播过多的广播信息(即“**广播风暴**”)而引起性能恶化。



**1. 阐述IP协议的作用，说明IP报文首部结构中各字段的作用**

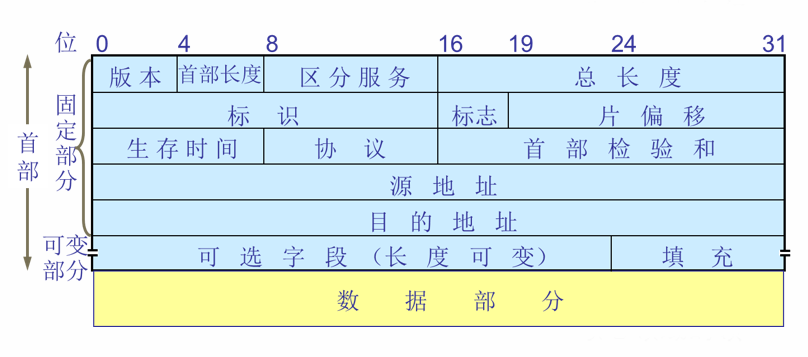
1.1 IP协议作用

（1）IP协议是Internet体系结构的**核心协议**，已成为**连接异构网络**的工业**标准**。

（2）IP提供**无连接**的数据报服务，每个IP分组长度**≤64K字节**，**不能保证**分组可靠的、按序到达，这些**留给高层协议**解决。

（3）IP协议需要**路由协议ICMP，ARP，RARP**等协议**支持**

1.2 IP报文首部各个字段作用



**版本**：4位，表示**IP协议版本**，通常为0100（v4），若为0110( v6)

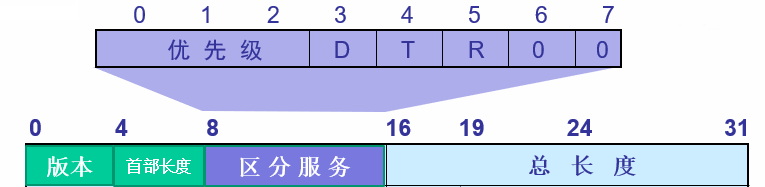
**首部长度**：4位，表示**IP包首部长度**，最短20字节，最长60字节

**区分服务**：8位，以前称为**服务类型**，从未使用过。1998年改称**区分服务**。

用于**指明要求网络提供的服务**，

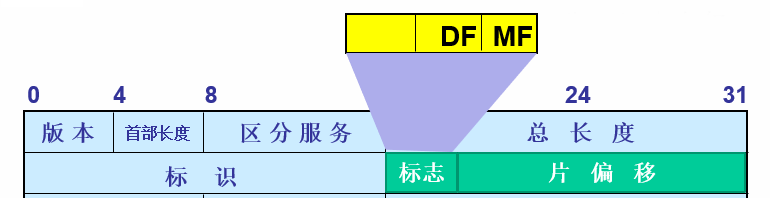
目前主要包括**D**延迟**、T**吞吐量**、R**可靠性等三种。

即使如此，依然没有真正使用。



**总长度**：16位，包括了**首部长度和数据长度**，最长为65535个字节。

**标识**：16位，**数据报计数器**，用于**区分数据报**的唯一标识符。在数据报传向目的地址时，如果将数据报**被分为多个片段**，那么**每个片段**都有**相同的标识符**。



**标志**：3 位，

- 最高位**保留**；

- 中间位是**不分片（Don‘t Fragment，DF）**标志，DF=**1**则**不允许**分片。

- 最低位是**有更多分片（More Fragment，MF）**标志

除**最后一个分片MF=0**以外外其余都是MF=1

**片偏移**：12 位，表示分片后，该片在原分组中的**相对位置**。

片偏移以 **8 个字节**为**偏移单位**。

**生存时间**：8 位，一般记为 **TTL (Time To Live)**。

表示数据报在网络中**可通过的路由器数**的**最大值**。

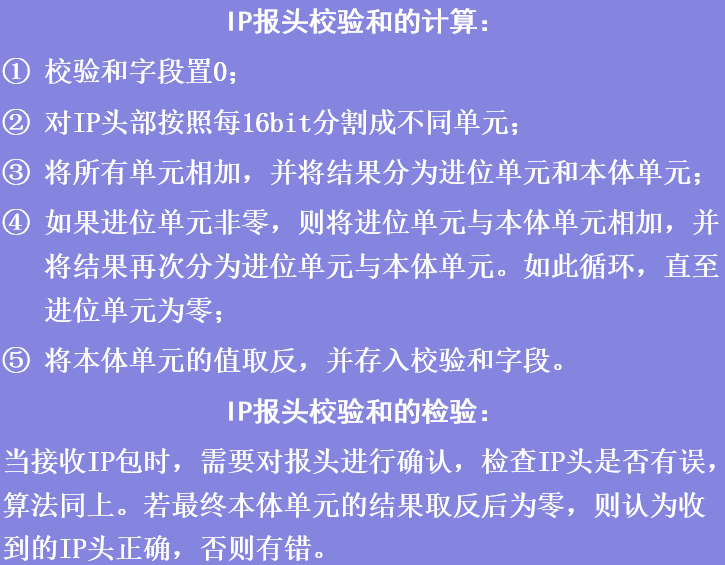
若**超出**最大值，则**丢弃**数据包，并返回“目标不可达”。

**协议**：8位，指出此数据报**使用何种协议**，

以便目的主机的 IP 层将数据部分上交给哪个处理过程。

如TCP、UDP、ICMP、IGMP、OSPF等。

**首部检验和**：16位，**只检验**数据报**首部**，**不检验数据**部分。

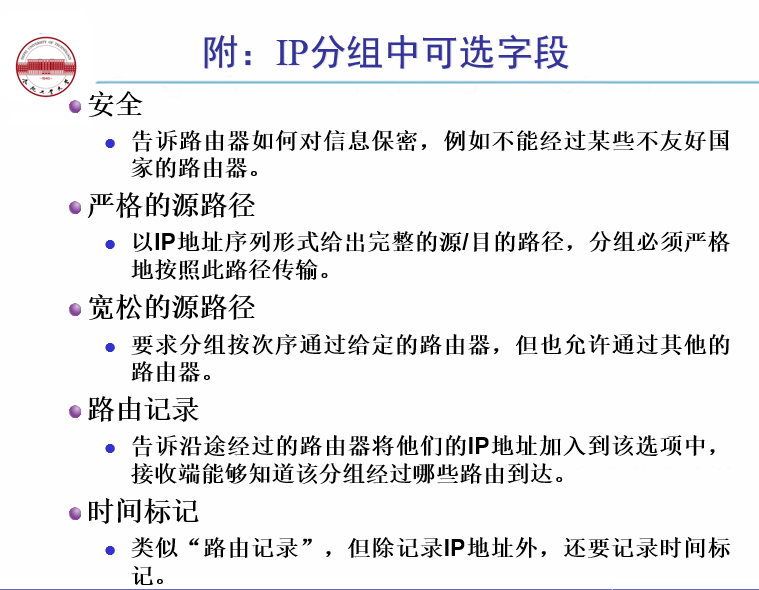


**源地址**：32位，发送端主机IP地址。

**目的地址**：32位，接收端主机IP地址。

**可选字段**：在整个包头长度**不超过60个字节**的情况下，可选字段的**长度可变**，

主要用来进行一些**测试工作**。主要**包括**：安全性（security）、松散源路由（Loose source routing）、 严格源路由（Strict source routing）、路由记录（Record route）、时间戳（Timestamps）等。



**填充**：根据可选字段的长度，填充**若干个0**，使得包头长度为32的整倍数。

**2. IP地址有什么作用？什么是保留地址？指出A、B、C三类IP地址各有哪些保留地址**

2.1 IP地址作用

**TCP/IP**网络模型将整个Internet视为一个**单一的、抽象的**网络，**IP地址**就是给每个连接在这个网络上的主机、路由器或其他设备分配的一个在全世界范围**唯一**的 32 位标识符。

2.2保留地址（私有地址）

各独立网络可以**重复使用**的IP地址，即**网络边界路由器**（通常就是**网关**）不会向目标地址为这些保留地址的主机转发IP分组。也就是说，保留地址**不会穿越内部网络**。

2.3三类IP地址的保留地址

- A类：10.0.0.0 （1个网络）

- B类：172.16.0.0—172.31.0.0（16个网络）

- C类：192.168.0.0—192.168.255.0（256个网络）

**3. 某公司有一个C类地址：192.16.12.0，请根据如下要求进行网络规划。**

**要求：划分7个子网，且每个子网可容纳11台主机。**

（1）对C类地址，要从最后8位中分出几位做子网地址：

∵23-2 < 7 <24-2

∴选择**4位**作为子网地址，共可提供14个子网地址。

（2）检查剩余位数能否满足每个子网主机台数要求：

∵子网地址为4位，故**还剩4位**可以用作主机地址

而24-2 > 11，所以可以满足每个子网11台主机要求

（3）子网掩码：255.255.255.240 （11110000B = 240）

（4）子网地址可在16、32、48、……、224共14个地址中任意选7个

**4. 已知IP地址：192.168.23.35/21，请说明其所属网络前缀，并给出该网络前缀所在CIDR地址块的范围**

192.168.23.35/21的**二进制**表示：

11000000 10101000 00010111 00100011 (1)

/21表示前21位为net-id，即**掩码**为:

11111111 11111111 11111000 00000000 (2)

**(1) and (2）**得到网络前缀:

11000000 10101000 00010000 00000000

即:**192.168.16.0**

CIDR地址块的范围：

从 11000000 10101000 00010000 00000000

到 11000000 10101000 00011111 11111111

即 **192.168.16.0~ 192.168.31.255**

**5. 阐述ICMP协议的作用和及其报文结构**

5.1作用

（1）作用

**- IP协议**提供了以无连接为特征的“尽力而为”的数据通信服务，并**不能解决**数据报**丢失、重复、延迟或乱序**等问题。

- **ICMP**的**目的**就是希望**对IP包无法传输时提供报告**，这些**差错报告**帮助发送方**了解网络中发生了什么问题，以确定应用程序后续操作**。

（2）**主要功能**：

- 通告网络错误

- 通告网络拥塞

- 协助解决故障

- 通告超时

5.2报文结构

（1）封装

① ICMP是少数几个可以直接从**应用层操作**的协议，不经过传输层。

② ICMP是一个**网际层协议**，

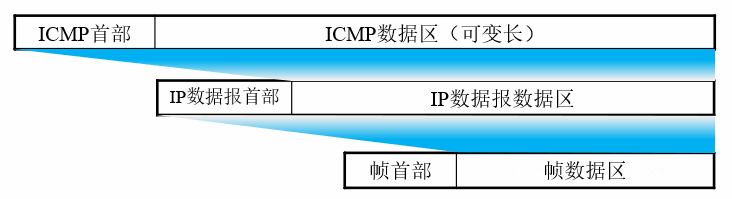
但却需要利用另一个网际层协议--IP协议进行封装，

因此ICMP是一种**需要两次封装的网际层协议**。

③ 首先由网络层的IP协议**对ICMP协议**报文进行**封装**，

其次再由数据链路层协议**对相应的IP报文**进行**封装**。

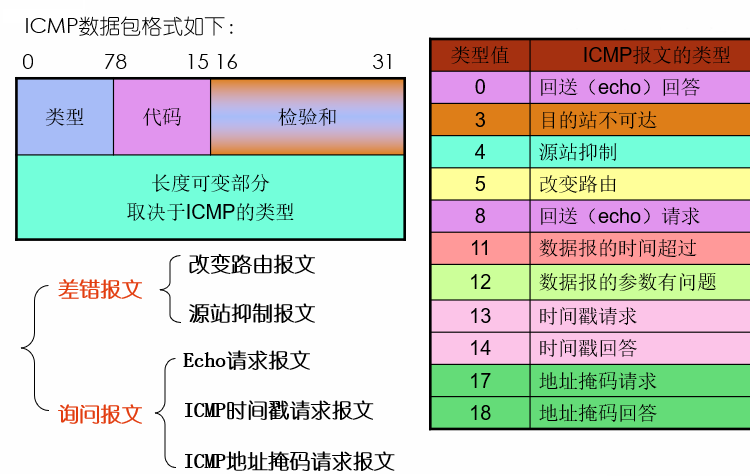
经过两次封装后的ICMP报文才能被**传输**。



④ ICMP报文是由IP协议封装，因此其也和IP协议一样是**不可靠的协议**，

既**不能保证**报文的**到达**，也不能保证报文**按序到达**。

（2）格式



**6. 在Windows中，实用tracert可以实现路由追踪目的。请说明tracert程序是获得路由的原理，以及说明tracert是如何知道发出的探测报文到达目的主机的。**

6.1获得路由的原理

Tracert完全**基于ICMP协议**实现**路由追踪**，**基本原理**是：

① 从**TTL=1**开始，Tracert**不断**从源IP向目标IP**发出**类型为8、代码为0的ICMP**查询报文**，并**逐次增加**TTL值。

② Tracert**根据不同TTL值**时，各路由器**返回的**“**TTL=0**”（类型为11代码为0）的**差错报文**，即可知道**经过了哪些路由器**。

实例理解：

（1）向目的发送TTL=1的UDP报文，第一个路由节点收到报文后发现TTL=0，向源端回应一个ICMP TTL超时报文，源知道了第一个路由节点。

（2）向目的发送TTL=2的UDP报文，第一个路由节点收到报文后TTL=1，继续转发给第二个路由节点，第二个路由节点收到报文后ttl=0，向源端回应一个ICMP TTL超时报文，源又知道了第二个路由节点。

（3）向目的发送TTL=3的UDP报文……

6.2如何知道到达目的主机

（1）方案一：针对UDP报文

**UDP报文头**有一个字段叫**目的端口**，用来**告诉目的端这个报文要请求服务类型**。当目的端服务（端口）不存在时，协议栈会向源发送一个ICMP端口不可达报文，告诉源，这儿没这服务，别来烦我。由此知道到达目的主机。

（2）方案二：针对ICMP报文

源持续向目的发送TTL递增的**ICMP echo报文**：收到ICMP TTL超时报文时认为是路径上路由节点，收到ICMP **echo replay**时认为**到达了目的**。

**7. IP地址与MAC地址有什么区别？**

（1）层级不同

**① IP地址**用于网络层，也常常被称为**逻辑地址**。

② **MAC地址**用于数据链路层。由于数据链路层向下紧邻物理层，受到物理层许多因素的影响，因此也被称为**物理地址**或**硬件地址**或**机器地址**。

（2）格式不同

① **IP地址**就是给每个连接在这个网络上的主机、路由器或其他设备分配的一个在全世界范围**唯一**的 **32 位**标识符。

便于记忆，采用**点分十进制记法**

② MAC地址有**48位二进制数**表示。

便于记忆，用**12个16进制数**代表，**每2个**16进制数之间用**冒号**分割。

比如：**08:00:20:0A:8C:6D**就是一个合适的MAC地址。

（3）分类不同

IP地址分为、分类IP地址、子网划分IP地址、超网IP地址

**8. 阐述ARP协议的作用及其报文结构**

8.1作用

**地址解析协议**（Address Resolution Protocol，ARP）是IP协议最重要的**配套协议**之一，其**作用**就是是**将IP地址转化成MAC（物理或硬件）地址**。

8.2报文结构



**硬件类型**：发送方硬件接口类型，**1**为**以太网**。

**协议类型**：发送方请求解析的协议地址类型，**0x0800**为**IP协议**地址。

**硬件地址长度**：发送方硬件地址长度，**以太网**地址为**48**。

**协议地址长度**：请求解析的协议地址长度，**IP地址**为**32**。

**操作类型**：表示本报文的类型

1为ARP请求，2为ARP响应，3为RARP请求，4为RARP响应。

**发送方硬件地址**：前4个字节和后2个字节分别在不同字段中表达。

**发送方IP地址**：前2个字节和后2个字节分别在不同字段中表达。

**接收方硬件地址**：前2个字节和后4个字节分别在不同字段中表达。

**接收方IP地址**

**9. 禁用主机的ARP协议后，最直观的现象是什么？为什么？**

9.1现象

主机上不了网，而且还不能与其他电脑连接

9.2原因

（1）无论OSI还是TCP/IP，都奉行的是**对等层通信原则**。在这一原则下，**低层**只是向上层**提供支持**：将上层的协议报文作为载荷**封装**在本层相关协议的数据单元中。在低层，**无法获得**也**不能理解**被封装的高层协议报文的内容。

（2）ARP协议**被禁用**，IP地址**无法转化**成MAC（物理或硬件）地址，**无法进行识别和连接**。

**10. 免费ARP有什么作用？**

（1）**宣告广播的作用**

告诉整个广播域，目前这个IP所对应的MAC地址是什么，可以加入ARP表。

（2）**确认广播域内有没有别的主机使用自己的IP**

① 如果使用了，则在界面上弹出“**IP冲突**”字样。过程：

- 普通ARP请求报文广播发送出去，广播域内所有主机都接收到

- 如果发现和本机的IP地址相同，则**将自己的MAC地址**填写到**该报文的目的MAC地址**字段，并**将该报文发回给源主机**。

② **免费ARP**的报文发出去是**不希望收到回应**的，只希望是起**宣告作用**；

如果**收到回应**，则证明对方也使用自己目前使用的IP地址。

③ 在所有网络设备（包括计算机网卡）up的时候，都会发送这样的免费ARP广播，以**宣告并确认有没有冲突**。

**1.什么是计算机网络?**

将若干台具有**独立**功能的计算机系统，用某种或多种通信介质连接起来，通过完善的网络协议，在**数据交换**的基础上，实现网络**资源共享**的系统称为计算机网络。

**2. 什么是计算机网络的拓扑结构？按照拓扑结构，计算机网络可以分为哪几种？各有什么特点？**

2.1计算机网络拓扑结构：

通信子网节点间连接结构的拓扑形式，通过结点与线段间的几何关系表示网络结构，反映网络中各实体的结构关系。

2.2按拓扑结构的分类及2.3特点：

（1）星型网

描述：传输介质从一个**中央结点**向外辐射连接其他节点。任何两个结点之间的信息交换必须经过中央结点转发。

特点：一旦中央结点发生故障，会引起整个网络瘫痪

（2）环形网

描述：网络上所有的结点通过传输介质连接成一个**闭环**，任何两个结点的数据交换必须沿环进行

特点：一旦结点或链路发生故障，则环路断开，导致网络瘫痪

（3）总线网络

描述：一条**总线**连接所有的结点，任何一个结点发送数据，其他节点都能收到。

特点：**共享信道**；任何结点故障都不会影响整个网络正常运行。

（4）不规则型网

描述：每个结点**至少要和其他两个结点**连接

特点：

①**可靠性好**：任何结点或链路发生故障，都不会影响网络的连通性。

②**布线灵活**：几乎不受任何拓扑结构的约束。

**3. 数据交换交换的作用是什么？按照数据交换形式，计算机网络可以分为哪几种？各有什么特点？**

3.1作用：

网络核心要向网络边缘中的大量主机提供连通性，使边缘部分中的任何一个主机都能够与其他主机通信。数据交换作用在于**接收来自源主机系统的数据，并向目的主机系统转发。**

3.2按形式分类及3.3特点：

（1）电路交换：**独占信道资源**、信道利用率**低**、实时性**好；**

（2）分组交换：采用**存储转发**、信道利用率**高**、实时性**差；**

（3）报文交换：**整个**报文存储转发，信道利用率、首部开销介于电路交换和分组交换之间。时延较长，很少使用。

**4. 网络中任意两台计算机系统之间可以采取哪些方式进行通信？分别有哪些特点？**

4.1通信方式及4.2特点：

（1）客户服务器方式（C/S）

特点：

①**客户**程序**必须知道**服务器程序的**地址**。

②不需要特殊的硬件和很复杂的操作系统。

（2）对等方式（P2P）

特点：

①可**同时处理多个**远地或本地客户的请求。

②系统启动后即自动调用并一直不断地运行着，**被动地等待**并接受来自各地的客户的通信请求。因此，**服务器**程序**不需要知道**客户程序的**地址**。

③一般需要**强大的硬件和高级的操作系统**支持。

**5. 阐述分组交换的工作过程，说明其特点。**

5.1工作过程

（1）将报文数据**划分**为等长数据块并加上包含地址等控制信息的首部。

（2）路由器收到一个分组，先**暂时储存**，再检查首部，查找路由表。

（3）按首部中的目的地址，找到合适端口**转发**给下一个路由器。

（4）重复以上步骤直到分组交付最终目的主机。

5.2特点

采用**存储转发**、信道利用率**高**、实时性**差**；

**6. 请分别阐述你对“带宽”、“速率”、“吞吐量”、“时延”和“信道利用率”的认识，并说明信道利用率并非越高越好的原因。**

**（说明：关于“速率”和“带宽”可参阅P53的相关内容，有兴趣的同学也可以阅读数据通信基础方面的资料）**

6.1认识

（1）带宽(bandwidth)

**本指**信号具有的频带宽度，单位是赫（或千赫、兆赫、吉赫等）。

**现指**数字信道所能传送的“最高数据率”的同义语，单位是“比特每秒”，或 b/s (bit/s)。

在**时间轴**上**信号的宽度**随**带宽的增大**而**变窄**。因此，更大的带宽往往与更高速率相联系。

（2）速率(data rate/bit rate)

数据传输速率，又称比特率，指**每秒钟实际传输的比特数。**

是信息传输速率（传信率）的度量。

单位是 b/s，或kb/s, Mb/s, Gb/s 等

速率往往是指**额定速率**或**标称速率**。

（3）吞吐量(throughput)

表示在**单位时间内**通过某个网络（或信道、接口）的**数据量**。

吞吐量更经常地用于对现实世界中的网络的一种测量，以便知道实际上到底有多少数据量能够通过网络。

**受**网络的**带宽**或网络的额定**速率**的**限制**。

（4）时延

①传输时延（发送时延 ）

是发送数据时，数据块**从结点进入到传输介质**所需要的时间。

也就是从发送数据块的第一个比特算起，到该块的最后一个比特发送完毕所需的时间。

②传播时延

电磁波在信道中需要传播一定的距离而花费的时间。

信号传输速率（即发送速率）和信号在信道上的传播速率是完全不同的概念。

③处理时延

交换结点为存储转发而进行一些必要的处理所花费的时间。

④排队时延

结点缓存队列中分组排队所经历的时延。

排队时延的长短往往**取决于**网络中当时**通信量**

（5）信道利用率

用于指出某信道有百分之几的时间是被利用的（有数据通过）。

**完全空闲**的信道的利用率是**零**。

网络利用率则是全网络的信道利用率的加权平均值。

信道利用率**并非越高越好**。

6.2原因

根据排队论的理论，当某信道的利用率增大时，该信道引起的时延也就迅速增加。

浏览器键入baidu.com所有用到的协议

当你在浏览器中键入baidu.com并按下回车时，浏览器会根据输入的网址进行一系列操作，涉及到多个协议。以下是在浏览器键入baidu.com时可能用到的主要协议：

DNS 协议（Domain Name System）：

浏览器首先会将域名 baidu.com 转换成对应的 IP 地址。这个过程使用 DNS 协议进行域名解析。

HTTP 协议（Hypertext Transfer Protocol）：

一般情况下，浏览器会默认使用 HTTP 协议进行网页的请求和响应。浏览器向服务器发送 HTTP 请求，请求获取 baidu.com 的网页内容。

HTTPS 协议（Hypertext Transfer Protocol Secure）：

如果网站启用了 HTTPS，浏览器会使用 HTTPS 协议进行加密的通信。这涉及到 SSL/TLS 协议用于加密数据传输，确保数据的安全性。

TCP 协议（Transmission Control Protocol）：

HTTP 和 HTTPS 协议都是基于 TCP 协议的，它负责在网络上建立可靠的连接，确保数据的可靠传输。

IP 协议（Internet Protocol）：

IP 协议用于在网络上进行数据包的寻址和路由。它是 TCP 和 UDP 的底层协议，负责在网络上进行数据包的传递。

ICMP 协议（Internet Control Message Protocol）：

在进行网络通信时，可能涉及 ICMP 协议，用于处理网络错误、诊断问题等。

以上是在浏览器键入 baidu.com 时可能用到的主要协议。实际情况中，根据网站的具体配置和浏览器的支持情况，可能还涉及到其他协议和技术，如 QUIC 协议、WebSocket 等。

TCP和UDP区别

TCP（Transmission Control Protocol）和UDP（User Datagram Protocol）是两种常用的传输层协议，它们在网络通信中有一些重要的区别：

1. 连接性：

TCP： 提供面向连接的通信。在数据传输之前，必须先建立连接，然后进行可靠的数据传输，最后再释放连接。

UDP： 是面向无连接的通信。每个数据包都是独立的，不需要先建立连接，直接发送数据。因此，UDP通信更为轻量，但不保证可靠性。

2. 可靠性：

TCP： 提供可靠的数据传输，通过序列号、确认和重传机制确保数据的完整性和顺序性。如果数据包丢失，TCP会进行重传。

UDP： 不提供可靠性保证，数据包可能会丢失，也不保证数据包的顺序性。对于某些实时性要求高的应用，可以容忍少量的丢包。

3. 数据传输方式：

TCP： 数据以流的方式进行传输，被划分为小的数据段，然后通过序列号进行重新组装。数据传输是可靠而有序的。

UDP： 数据以数据包的形式传输，每个数据包都是独立的。UDP更适用于需要快速传输的场景，如实时音视频。

4. 头部开销：

TCP： TCP头部相对较大，包含序列号、确认号、窗口大小等字段，用于保证可靠性和流控制。

UDP： UDP头部较小，只包含源端口、目标端口、长度和校验和等基本信息，开销较小。

5. 适用场景：

TCP： 适用于对数据完整性和顺序性有要求的应用，如文件传输、网页访问等。

UDP： 适用于实时性要求高、可以容忍少量数据丢失的应用，如语音通话、视频直播等。

6. 连接和断开：

TCP： 建立连接时有三次握手的过程，断开连接有四次挥手的过程。

UDP： 无连接的特性，不需要建立和断开连接的过程。

总体来说，选择TCP还是UDP取决于具体的应用需求。TCP适用于要求可靠性和有序性的场景，而UDP适用于实时性要求高、可以容忍少量数据丢失的场景。在某些应用中，也可以结合使用TCP和UDP，根据不同的数据传输需求选择合适的协议。