computer-network

一、计算机网络概述

一、计算机网络的早期发展

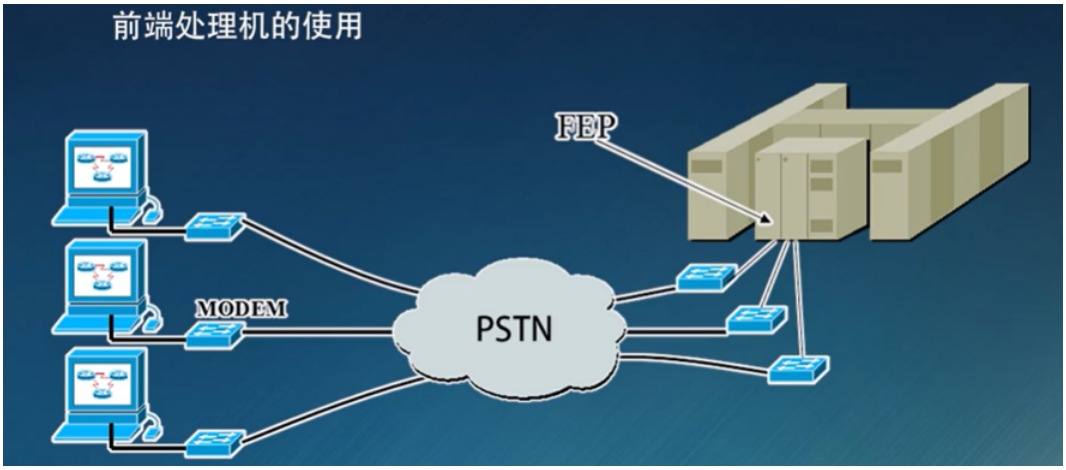
1. 网络的作用：
    - 知识经济、信息化：以网络为核心的信息时代
2. 载体：“三网”、三网合一
    - 电信网
    - 有线电视网
    - 计算机网络（核心）
3. 网络的雏形：使用（transceiver）收发器进行简单的结合，被称为“面向终端的计算机通信网”
    - 

    - 拓扑结构：星形
    - 网络中心是一台高性能计算机，负责整个网络的控制工作。多个终端围绕在它的周围，在多个终端上安装有收发器设备，**作用是为终端发送或接收数据**，此时的网络工作是在电信网络的基础设施上完成的。
    - PSTN：公用电话交换网（常说的电话网），电话网只能传说模拟型号，为实现计算机中数字数据的传输，通信的双方使用**调制解调**

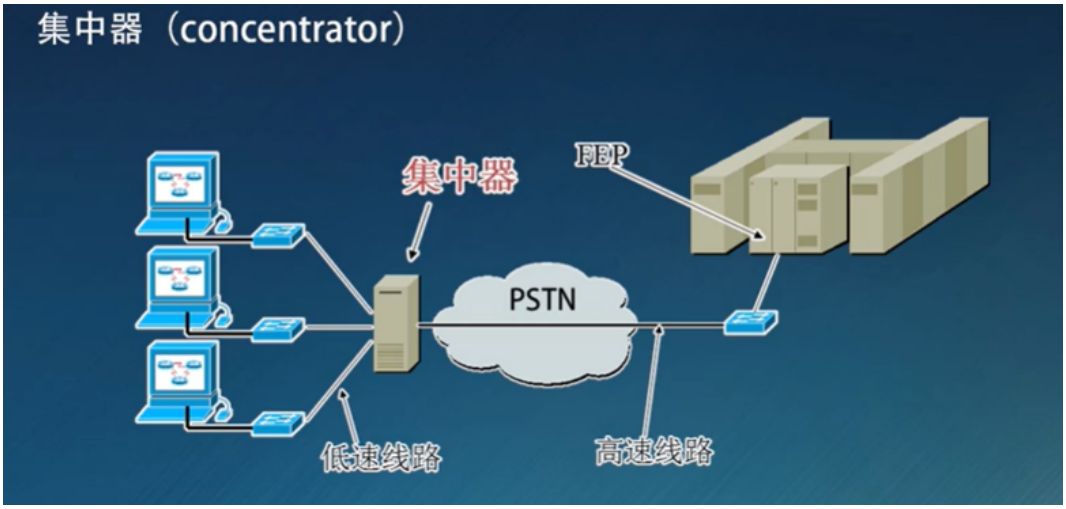
器来进行数字信号和模拟信号之间的转换。

- Line controller--线路控制器:**多重线路控制器可以实现将计算机与多个终端相连，同时线路控制器还具有数据串并行转换和差错控制功能。**

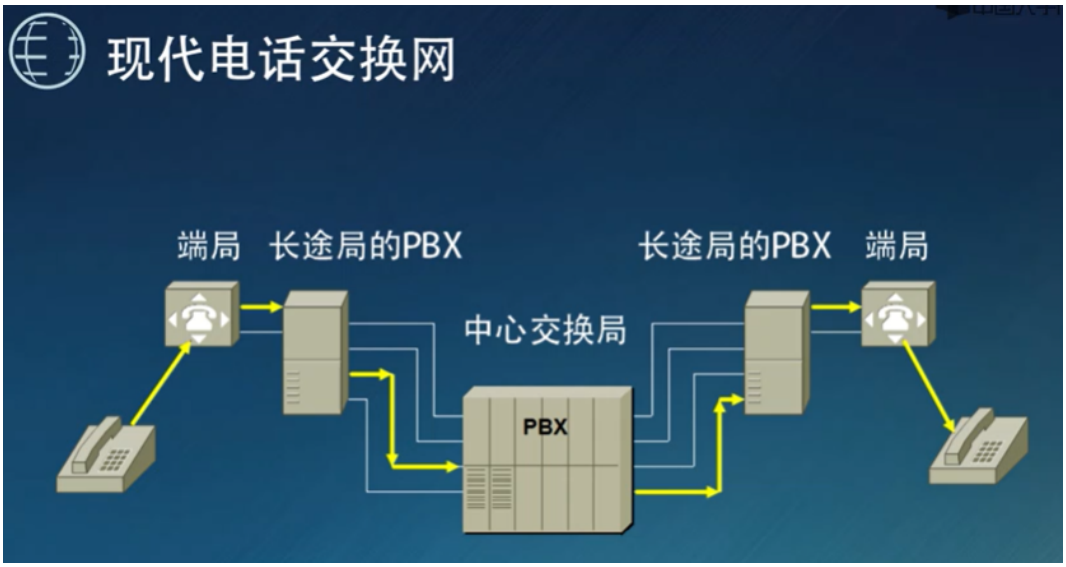
4. 随着计算机用户的增加，网络中每增加一个终端，线路控制器要付出很大的代价，同时对主机造成了很大的负担，严重的影响了主机的工作。此时迫切的需要一个能够帮助主机分担工作的设备，于是**前端处理机FEP**就出现了。

   - 前端处理机也称为：通信控制器，承担原来主机的全部通信任务，将主机从繁重的通信任务中释放出来，更好的运行应用程序。（现在普遍存在的网卡就是前端处理机的化身）
   - 

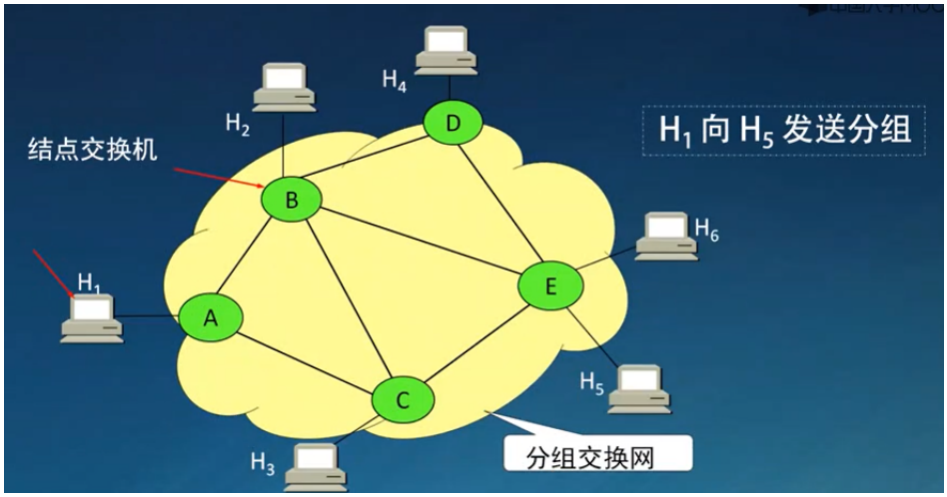
5. 远程终端的增加不仅会给主机带来负担，同时通讯费用也会迅速增加，特别是利用高速通信线路进行传输的时候，为了降低通信成本，在终端密集的地方出现了线路集中器，它是一种智能复用器，可以通过低速线路将多个终端集中起来，对后面的高速线路进行复用，这样高速线路的容量就可以远远小于各低速线路容量和，从而降低通信费用。

   - 

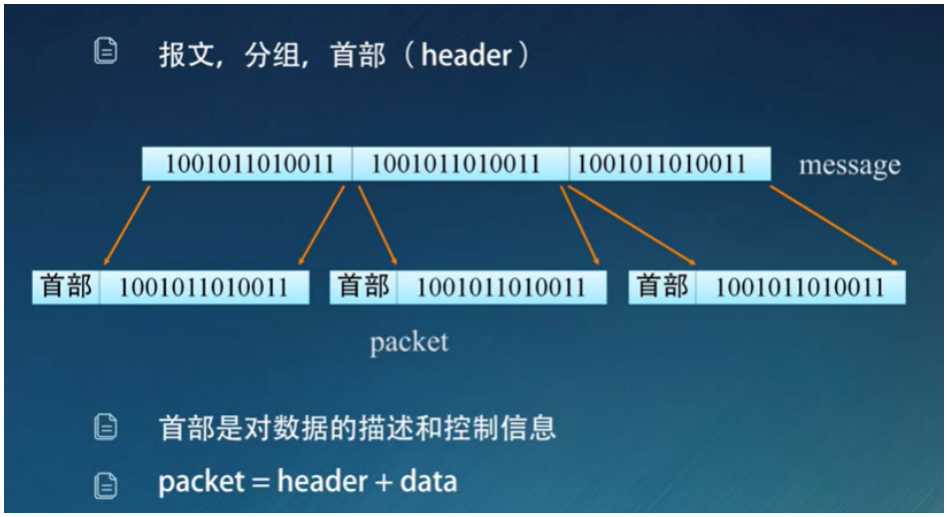
二、分组交互网

1. 从通信资源的分配角度来看，交换就是按照某种方式动态分配传输线路的资源。
    - 电路交换是电信网中采用的数据传输方式，在使用电路交换时，必须在通信双方之间建立一条物理连接，这条连接占用了双方通信所需要的资源，而这些资源在双方通信时不会被其他用户占用，直到通信结束。
    - 

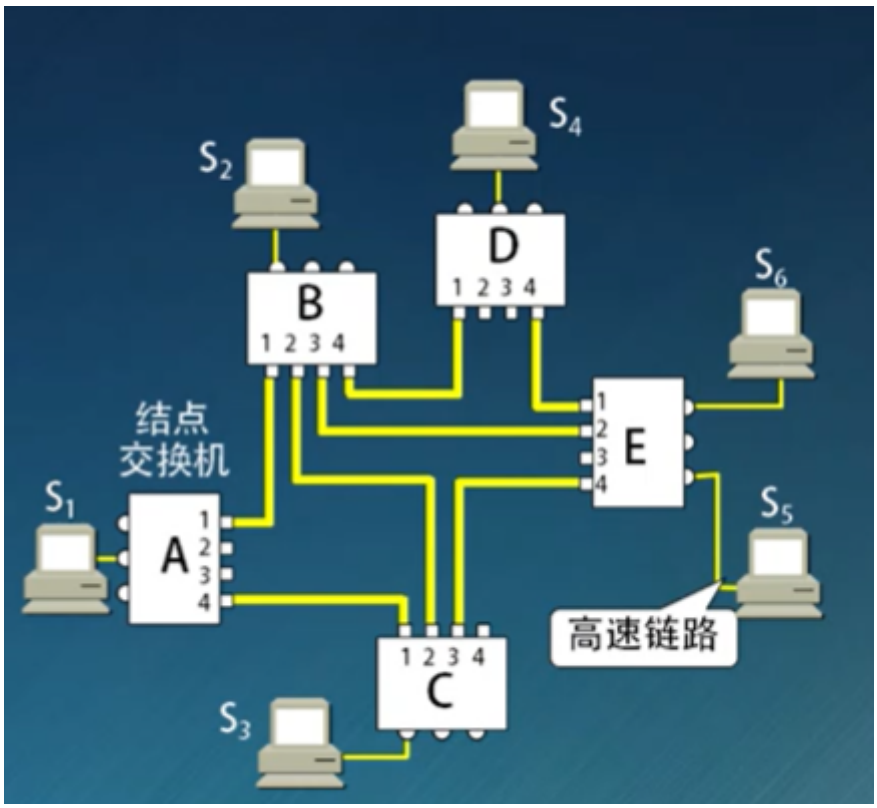
2. 为什么传统电路交换不适合计算机数据传输？
    - 浪费资源：使用专用的物理通路
        - 电路传输在双方没有数据传输时，所占用的通信资源会白白浪费
        - 而计算机的数据传输具有突发性，非连续性，因此在通路建立的这段时间内可能会有大量的时间是出于空闲状态，造成了严重的资源浪费
    - 不同规格的终端互连不便：需要协调缓冲机制和转换机制
        - 计算机不像电话那样种类规格单一，它的系统结构和规格多种多样，所以需要付出额外的转换代价
    - 不灵活：单点故障，全线瘫痪------需要重拨
        - 在电路交换的线路中任何一处出现故障都会导致正太通路中断。在远距离计算机数据传输中，这个问题会严重影响数据传输效率
3. **分组交换**
    1. 分组交换的特点
        1. **采用存储转发技术**，基于数据从发送方到接收方中间要经过多个节点的暂时存储，然后根据接收方的地址找到下一个接收该数据的节点，并将数据转发除去，知道到达接收方为止。

- 

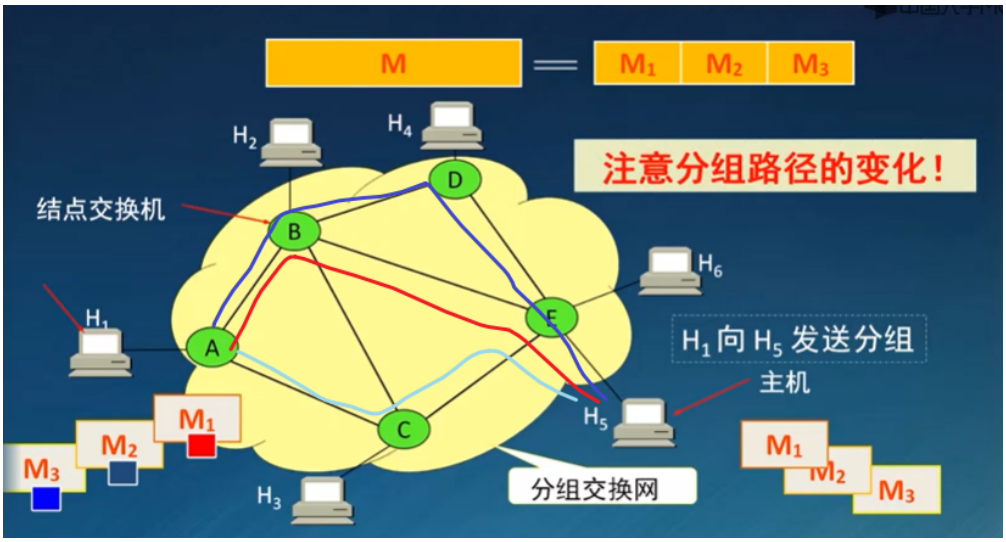
  - 比如现在要从主机H1向H5发送分组，分组从H1出发后，经过结点交换机A C E的转发，最终到达目的主机H5。

2. 分组交换技术的另一个特点：发送数据之前要对数据进行分段处理。通常我们把要发送的整块数据称为一个报文，在发送报文之前，先把较长的报文划分成为一个个更小的等长数据段，在每个数据段前面加上一些必要的控制信息组成首部，这样就形成了一个分组。
   - 分组有时候也称为包，分组的首部称为包头
   - 

2. 分组交换网的组成
   1. 主机（host）
      - 主要为用户进行信息处理，并且可以通过网络和其他主机交换信息
   2. 结点交换机（node switch）
      - 节点交换机则是进行分组的存储转发
      - 路由器

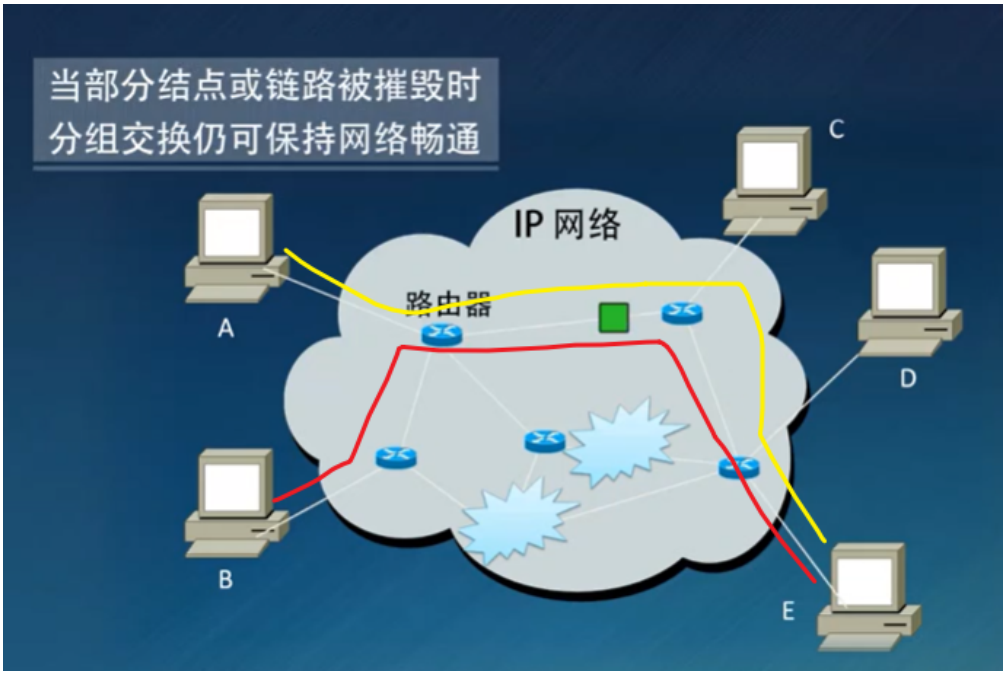


3. 分组交换网中一个报文的传输全过程

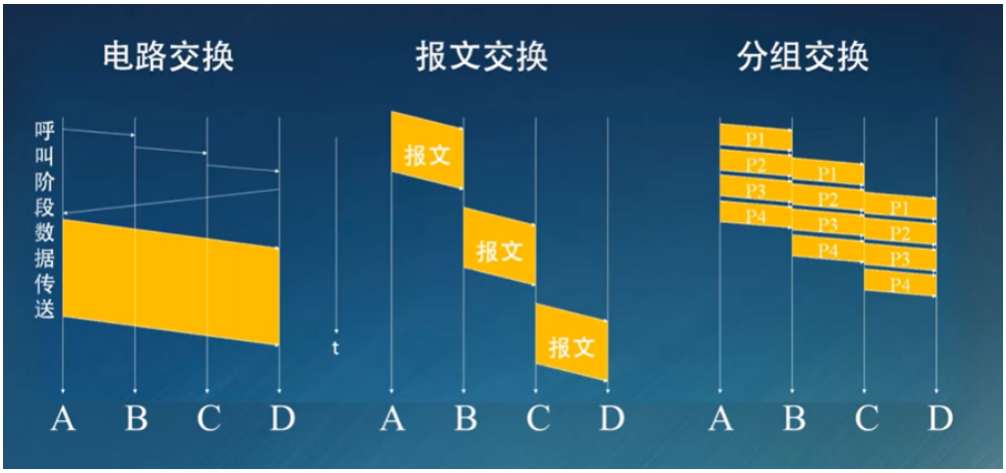
1. 

2. 对于一个报文M，分成了三个分组M1，M2，M3，分别用不同的颜色表示
3. 对于M1，M2，M3三个分组都到达了目的主机，但是中间经过的结点交换机并不相同，即它们是经过不同的路径传输的

4. 当网络中出现故障时的通信情况

1. 

    2. 分组传输过程中，当网络中某条通路断开或结点交换机出现故障时，分组避开故障点，重新选择新的路径，传输工作基本不受影响。
    3. 由此可见，当部分结点或链路被摧毁时，分组交换仍然可以保持网络畅通
5. 分组交换的优点
    1. 高效：逐段占用通信链路，动态分配传输带宽
    2. 灵活：智能节电能够独立处理数据分发
    3. 迅速：分组作为传输单位，无需建立连接
    4. 可靠：完善的网络协议
    5. 分布式多路由的网络结构提高了网络的生存性
6. 分组交换的缺点
    1. 传输时延：分组交换在各中间结点进行存储转发时需要排队，会造成一定的时延
    2. 首部数据存在开销：各分组必须携带一些控制信息，造成了一些传输开销
7. 20世纪40年代的电报通信中也采取了基于存储转发原理的**报文交换**，通过实例比较**电路交换**，**报文交换**，以及**分组交换**的区别
    1.

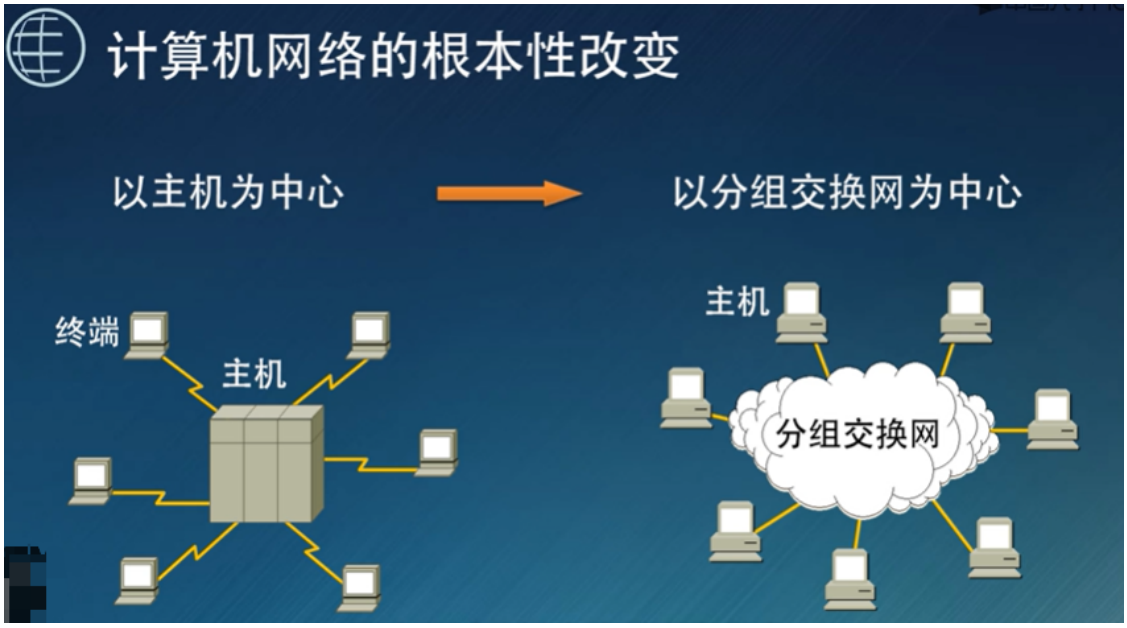


2. A，D分别是原点和终点，B，c是在A和D之间的中间节点，箭头方向代表时间
3. 电路交换：
    - 首先进行连接建立工作
    - 连接建立好后，整个报文的比特流连续的从原点直达终点（类似管道）
4. 报文交换：
    - 整个报文现到达相邻节点，全部存储下来后，查找转发表，然后转发到下一个节点
    - 再重复存储转发的过程，直到到达终点为止
5. 分组交换：
    - 一个分组传送到相邻节点，存储下来后，查找转发表，然后转发到下一个节点
    - 而此时，刚刚被这个分组占用的链路上就可以继续传输下一个分组了
    - 在整个报文传输过程中，单个分组可以最大程度的充分利用通信链路资源

8. 三种交换方式的对比
    - 电路交换时何传输大量、连续的数据
    - 如果要传输少量、突发的数据则应该使用分组交换
    - 分组交换链路分配灵活

三、Internet的发展与网络的分类

1.



- 计算机从早期的以主机为中心变为现在以网络为中心
- 早期计算机网络是面向终端的星形网，个终端通过通信线路共享昂贵的中心计算机的硬件和软件资源
- 分组交换网是以网络为中心，主机出于网络外围，用户通过分组交换网可以共享网络上各种丰富的资源

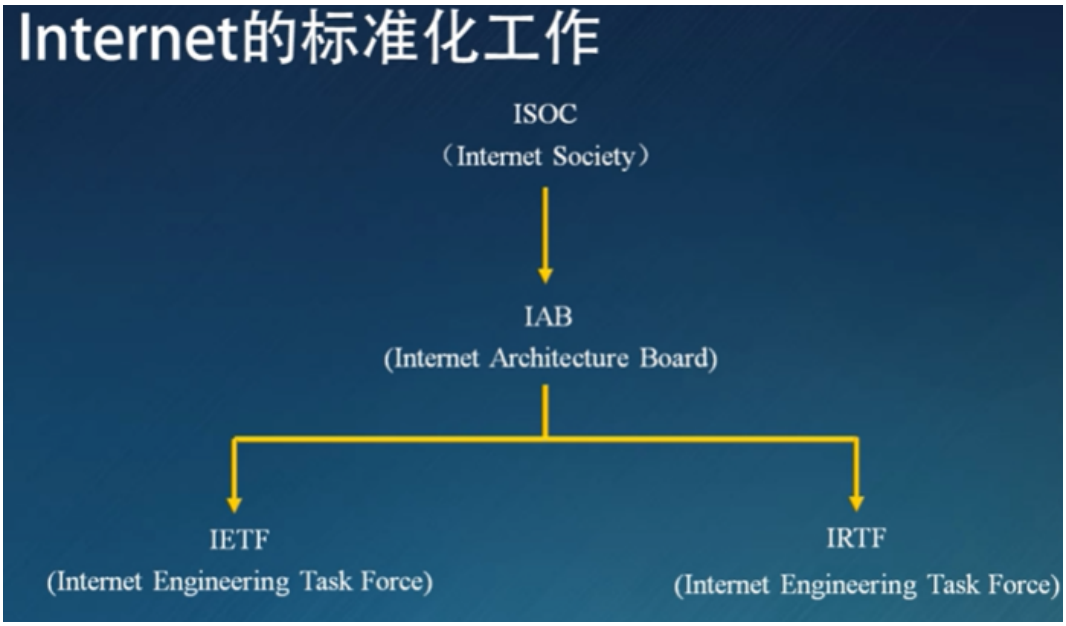
2. Internet发展的第一阶段

- 第一个分组交换网ARPANET最初只是一个单个的分组交换网
- ARPA研究多种网络互连技术
- 1983年TCP/IP协议成为**标准协议**
- 1983~1984年，形成因特网Internet
- 1990年ARPANET正式宣布关闭

3. Internet发展的第二个阶段

- 三级计算机网络：主干网、地区王、校园网
- 各网络之间使用路由器连接，两个主机之间的通信可能需要经过多级网络
- 

4. Internet标准化工作

- 

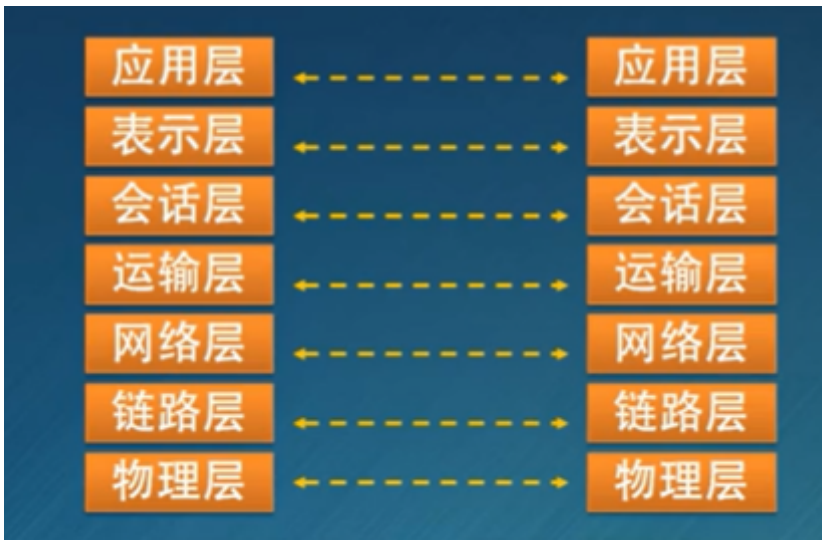
- ISOC(Internet Society)Internet协会
- ISOC下面有个技术组织因特网体系结构委员会---IAB（Internet Architecture Board），负责Internet有关协议的开发
- IAB下面设有两个部门
  - 因特网工程部---IETF（Internet Engineering Task Force）：工程部负责协议的开发和标准化
  - 因特网研究部---IRTF（Internet Researching Task Force）：研究部进行理论方面的研究

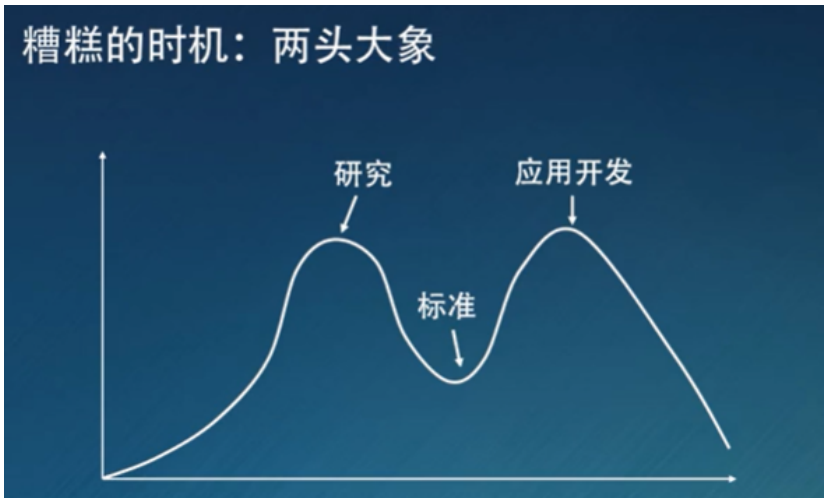
- 所有Internet标准都是以RFC文档的形式在网上发表（Request for Comment---请求评论）
  - 文档级别
    1. 草案（Internet Draft）
    2. 建议标准（Proposed Standard）
    3. 草案标准（Draft Standard）
    4. 正式标准（Official Standard）

5. 中国互联网建设
   - 第一个分组交互网-----CNPAC铁道部
   - 1994年正式介入Internet：首次使用64k比特每秒的专线正式介入Internet，成为国际上承认计入Internet的国家
6. 计算机网络分类
   1. 按照网络的作用范围进行分类
      - 广域网（Wide Area Network）：几十到几千公里，Internet的核心部分，任务是进行远距离数据传输
      - 局域网（Local Area Network）：地理上局限于比较小的范围内
      - 城域网（Metropolitan Area Network）：一个城市，5~50公里，一个或几个单位所有，也可以是公用设施
   2. 按照网络使用者分类
      - 公用网：电信公司出资建造的大型网络，所有按照电信公司缴纳费用的人都可以使用
      - 专用网：某个部门为本单位特殊业务工作的需要建造的网络，不对外提供服务
   3. 按照拓扑结构分类
      - 星形、总线型、环形、树形、网状
      - 

      - 规则拓扑结构的网络往往是局域网，而不规则的网状结构通常应用于城域网和广域网

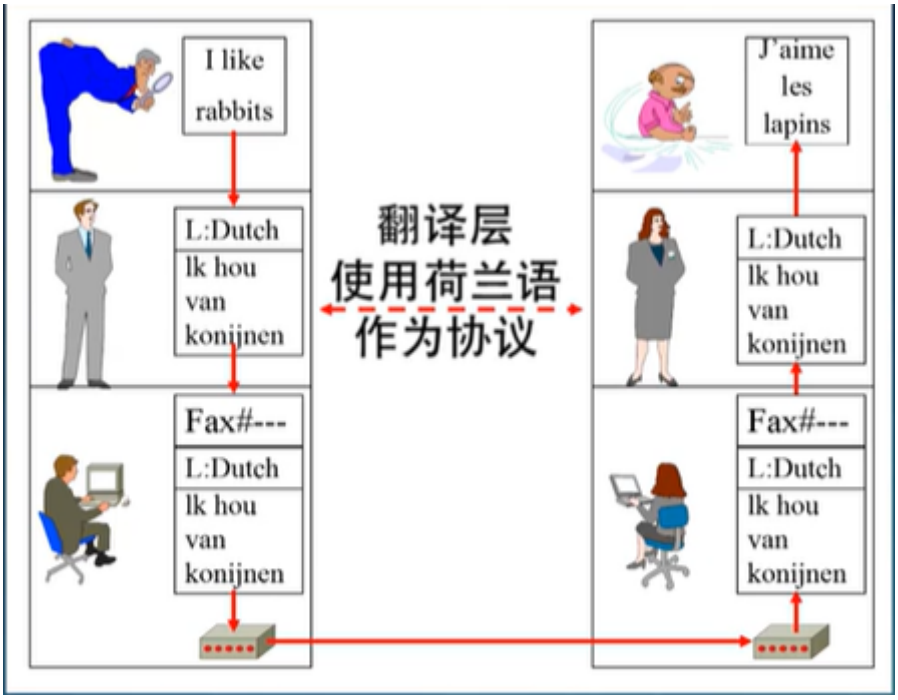
四、计算机网络体系结构

1. 网络体系结构的发展
    - 网络体系结构的工作内容
        - 激活：使用信号确保数据能够在这条通路上正常的发送和接收
        - 进行数据分析，识别接收方
        - 发送放对接收方的就绪检测
        - 文件系统的格式转换
        - 差错控制
        - 其他
    - 两台通信的计算机必须高度协调才行，这种高度协调是非常复杂的，所以早期就提出了**分层**的办法来解决，而这些层次的集合就是网络的体系结构
    - 网络体系结构发展历史
        - 第一个阶段：“群雄逐鹿”
            - 各个公司各自拥有自己的网络体系结构，如：IBM和SNA、Novell
            - 不同公司的设备不兼容导致大公司的垄断，技术不兼容，所以导致了OSI模型的出现
            - OSI七层模型---ISO7498
                - ISO：International Standard Organization
                - OSI/RM：Open System Interconnection Reference Model：开放系统互连基本参考模型
        - 第二阶段：OSI“一统天下”
            - 

- 两个使用该标准的主机可以在世界上任何地方通信
- OSI七层模型失败的教训：
    1. 糟糕的时机
        - 

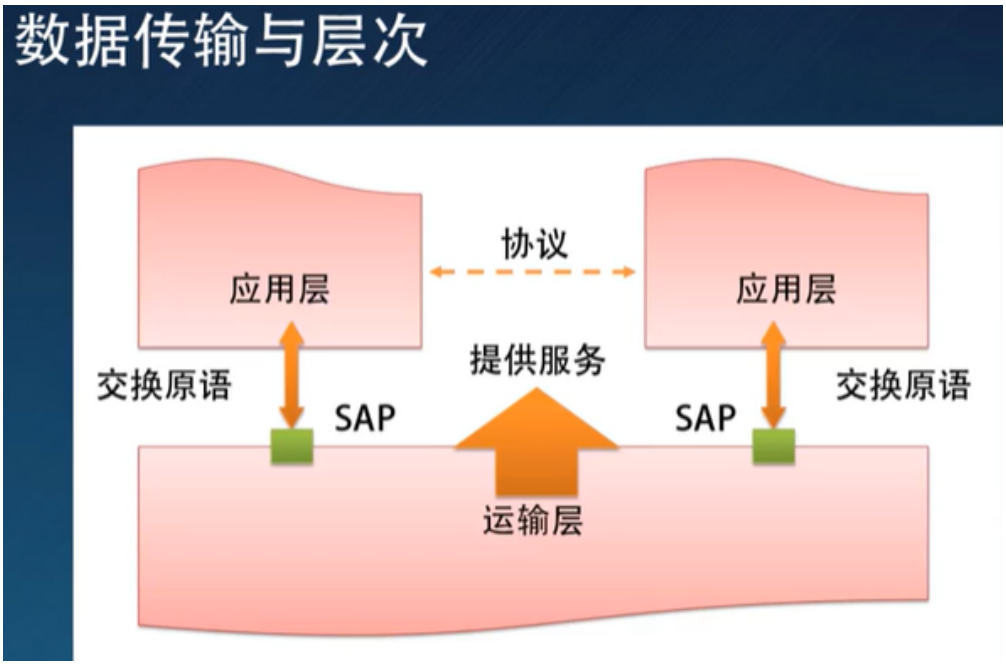
    2. 糟糕的技术
        - 会话层和表示层几乎是空的，而数据链路层和网络层的内容又太多，模型中的定义过于复杂，实现起来特别困难，有些功能在不同层上有多次出现
    3. 糟糕的实现
        - 由于协议和定义过于复杂，实现的非常糟糕，效率低，使用感受特别差
    4. 糟糕的策略
        - 政府的产物，被认为是政府将糟糕的产物强加于人们，与此同时，TCP/IP协议得到广泛的应用

2. 不论是TCP/IP模型还是失败的OSI模型，它们在结构设计上都采用了**分层的思想**。那么网络体系结构为什么要分层，分层有什么好处？
3. 分层的网络体系结构
    1. 哲学家交流模型

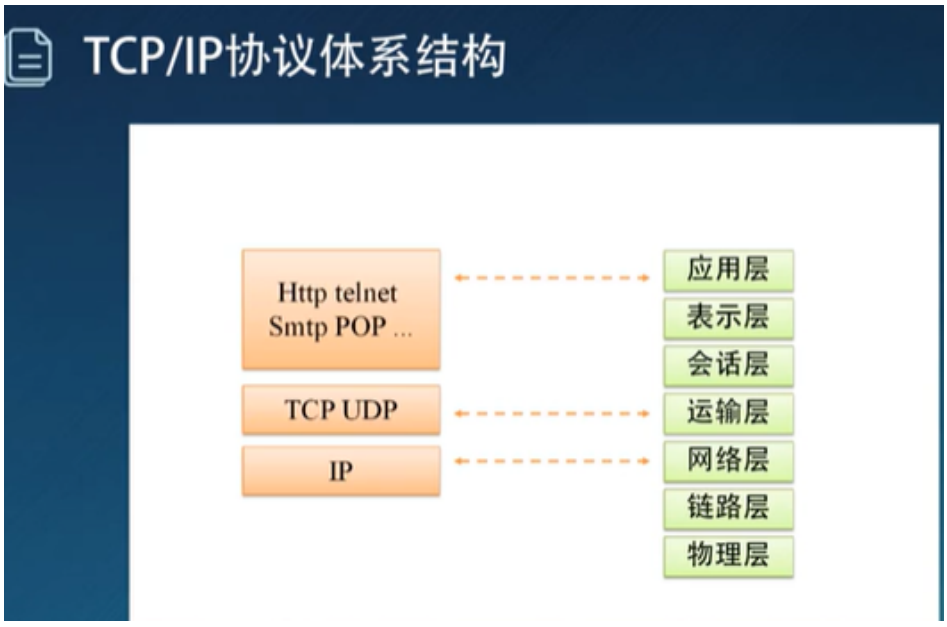
- 

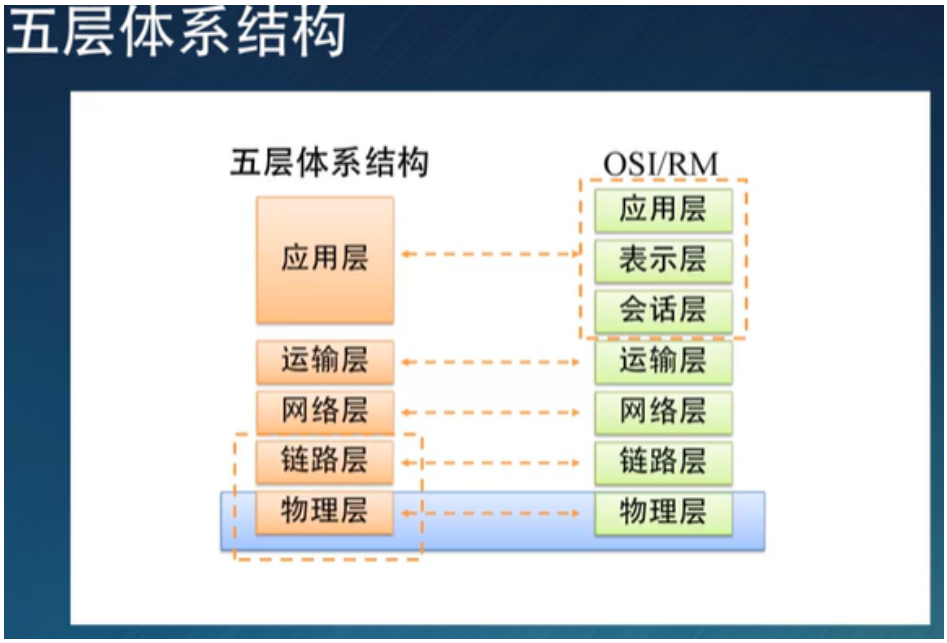
  - 假如有两位来自不同国家的哲学家，现在需要对某个问题进行讨论，其中一个哲学家会乌尔都语/英语，另一位哲学家会法语/汉语。两个哲学家位于最高层。由于语言不同，所以他们雇了两个翻译----第二层，两位翻译用双方都会的语言荷兰语。所以哲学家1把他对兔子的情感传达给哲学家2的时候。先将其传给他的翻译，然后他的翻译翻译成荷兰语。接下来翻译将翻译的内容交给秘书，让它发送出去给哲学家2的秘书。哲学家2的秘书将消息送给哲学家2的翻译，翻译将内容翻译给哲学家看。
2. 从哲学家交流模型里可以发现分层的好处：
    1. 层次间的相对独立性
    2. 灵活性
    3. 易于分割
    4. 易于实现和分割
    5. 易于标准化
3. 设计体系结构是分层的原则
    1. 层次数量不能过多----过多会增加传输成本和开销
    2. 层次数量不能过少----过少导致各层次功能还是过于复杂，难以实现
    3. 类似的功能放在同一层----利于实现
    4. 在实现技术经常变化的地方增加层次-----利用结构的灵活性，是变化带来的影响最小
    5. 层次边界选择要合理，使得信息量最小
4. 对等的协议需要在协议的规范下工作。协议主要由三个要素描述：语义、语法、时序。

5. 某一层次的协议对于其他层次来说是透明的。在协议规定下，把对等层次下传输的数据单元称为这一层次的协议数据单元---PDU。下层负责向上层提供服务，上层不需要知道下层如何实现的。形象的比喻**协议是水平的，服务是垂直的。**下层和上层进行信息交换的接口称为服务访问点----SAP，主要用于上下层次服务原语的交换。服务原语----上层使用下层服务时需要传输的一些命令。



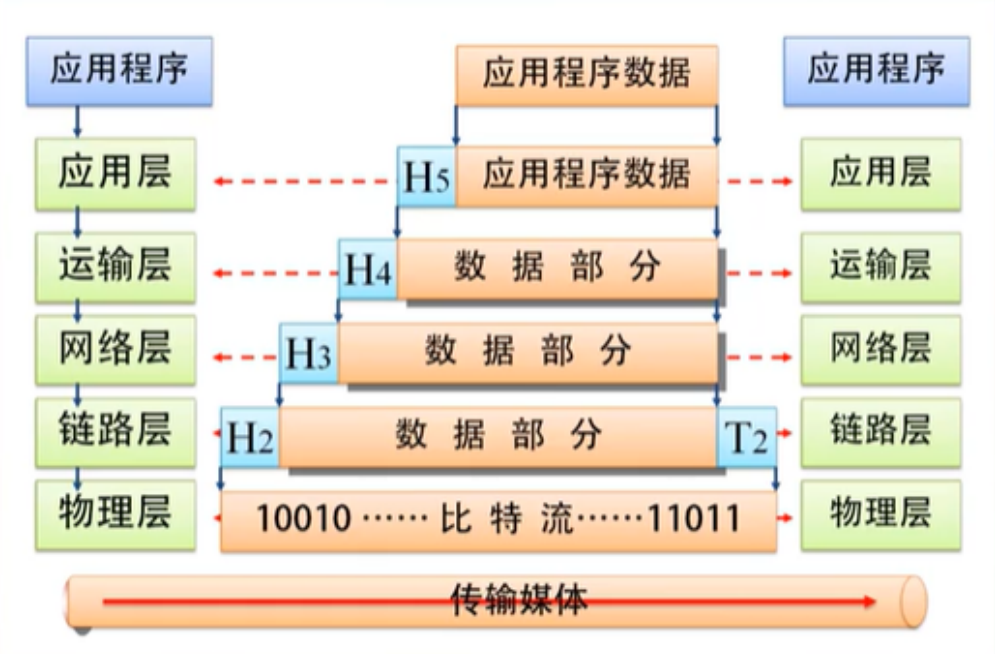
6. 五层体系结构
   - 尽管与OSI模型相比，TCP/IP模型获得了巨大的成功，但这种体系结构并不是完美的。比如它并没有区分服务、接口和协议的概念，模型不通用，不能使用TCP/IP之外的其他协议。上三层的功能主要由各种实用的协议来定义；网络层一下定义了一个接口，没有区分数据层和物理链路层。



- 综合TCP/IP和OSI模型的优点，采用5层结构模型来了解计算机网络
- 

    1. 应用层---最高层，直接为用户进程提供服务。
    2. 运输层：以报文为单位在两个主机之间提供可靠的端到端的数据传输。
    3. 网络层：以分组为单位，实现数据在不同子网中的传输。
    4. 数据链路层：负责将数据封装成帧，实现透明的无差错传输。
    5. 物理层：是透明的传输bit流，需要考虑如何在不同的传输媒体上的传输问题

7. 以一次数据传输为例，感受数据在不同层次间传输的问题

- 

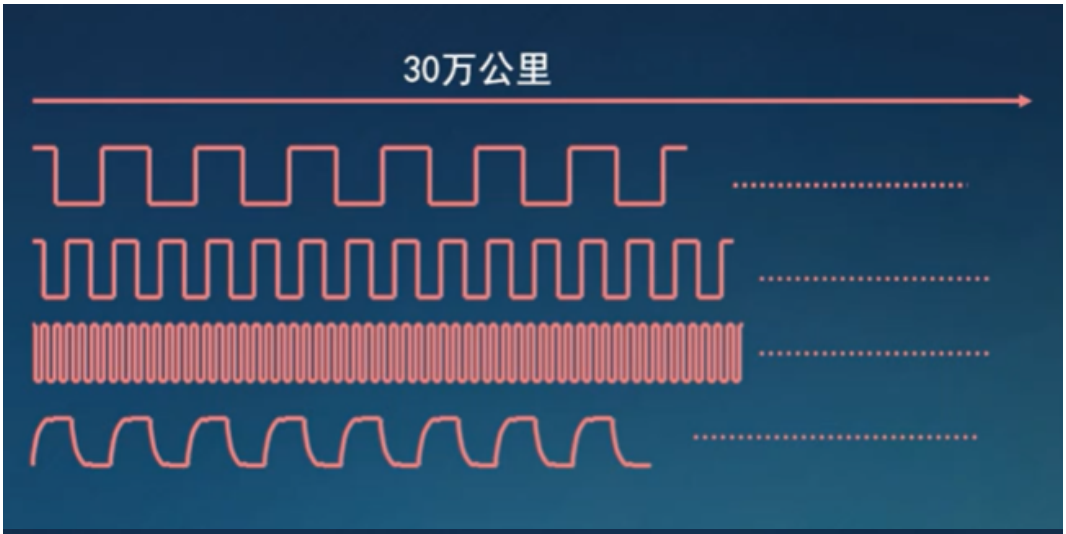
- 上图中两边代表两台网络主机，现在这两台主机应用进程需要通过网络进行数据传输。假设左边的主机是数据的发送端，右

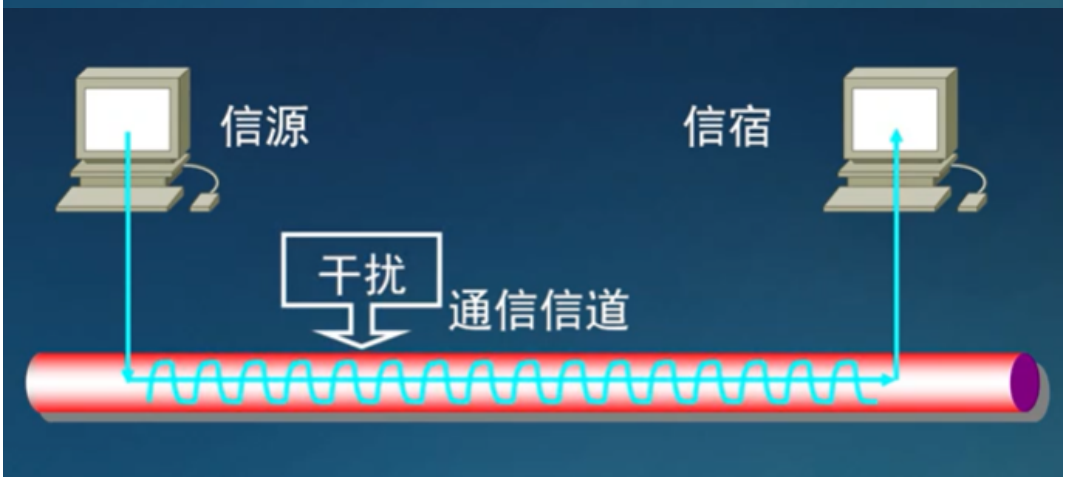
边的主机是数据的接收端。应用进程将需要发送的数据首先交给应用层处理。应用层根据相应的应用层协议将该数据进行封装，也就是加上协议的头部字段，这样就形成了应用层的PDU。接下来这个PDU被送到运输层，运输层将整个PDU全都看成数据部分，然后在它的前面加上运输层协议的头部，形成运输层的PDU。然后继续向下交给网络层，网络层进行类似的处理，在收到的数据前面加上网络层的协议头部，形成网络层的PDU。然后交给链路层，链路层加上头部和尾部进行最后一次封装，形成数据帧。然后交给物理层的主要工作是将链路层封装好的帧转换成何时传输的信号，通过传输媒体将数据发送给接收方。而接收方主机在收到这个数据后，执行和发送方相反的操作，即逐层解析各PDU的头部字段，取出其中的数据部分，提交给上层，直到应用进程为止。

二、物理层

一、物理层的基本概念

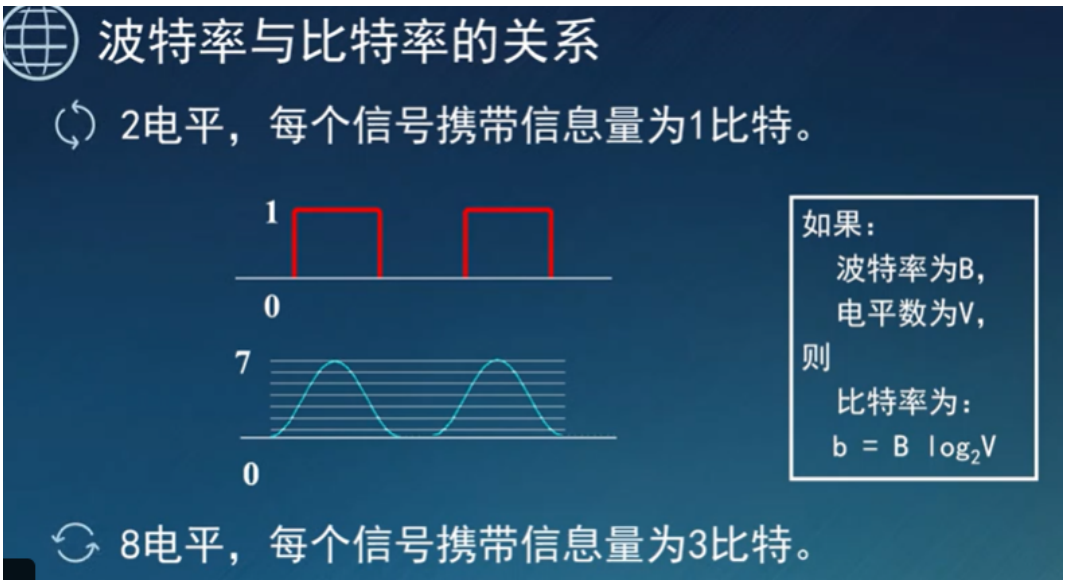
1. 物理层的任务：确定与传输媒体的接口特征
    - 主要是机械特征：指接口所用接线器的形状，尺寸，引线数目及排列等等
    - 电气特征：指接口电缆所承载电压范围
    - 功能特征：指某条传输线上电压的意义，比如高电压代表数据1，低电压代表数据0
    - 规程特征：指在功能实现过程中各种事件出现的顺序
2. 信道：指向某一方向传输信息的媒体。
    - 通信信道=发送信道+接受信道
    - 传输方式上有3类：
        1. 单向通信：也叫单工通信，特点是只能向一个方向进行信息传输，没有反方向的交互，因此单向通信只需要一条信道。比如使用耳机收听音频文件
        2. 双向交替通信：也叫半双工通信，特点是通信双方都可以发送信息，但是双方不能同时进行。由于这种信息传递方式有两个传输方向，所以需要两条通信信道。比如使用对讲机进行通话
        3. 双向同时通信：也叫全双工通信，特点是通信的双方可以同时发送和接收信息。比如打电话，也需要两条通信信道

3. 信道带宽：指信道所能通过的信号的频率范围。
    - 信道带宽越宽，能够通过的信号的高频分量就越多，也就可以用更高的速率来传送码元。因此**信道带宽**是影响信道传输速率的一个因素。
    - 信号在信道上的传输存在差异，有的频率高一些，有些低一些，但信道所能通过的频率范围总是有限的。当信号的频率超出这个范围时，信号中的高频分量就会收到衰减，坡形边界变模糊，导致接收方对码元的提取困难。
    - 

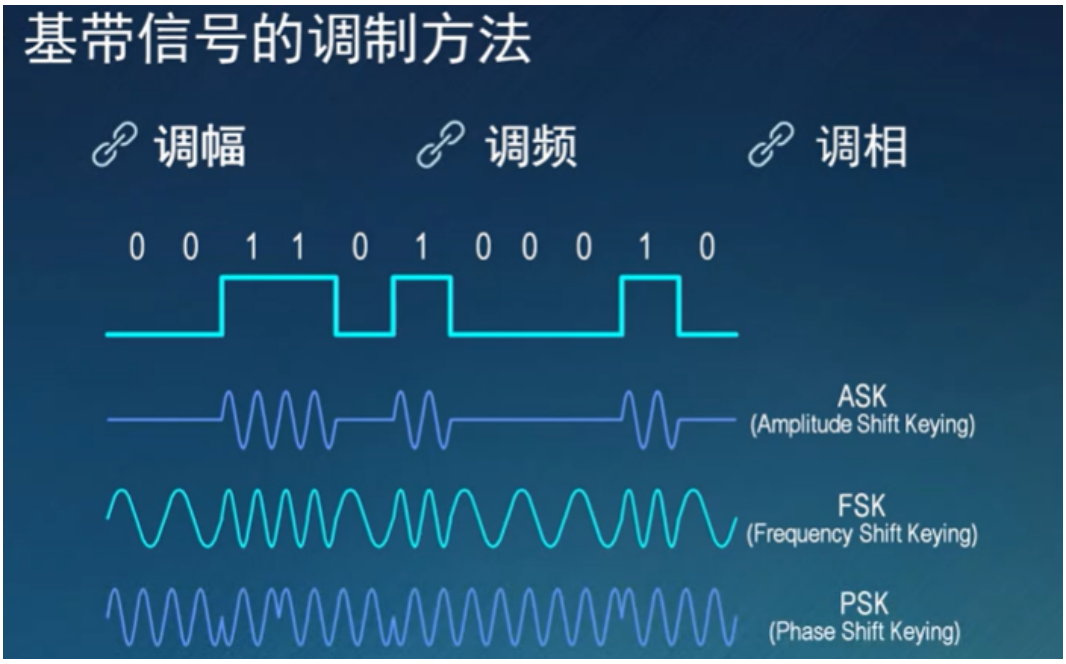
    - 

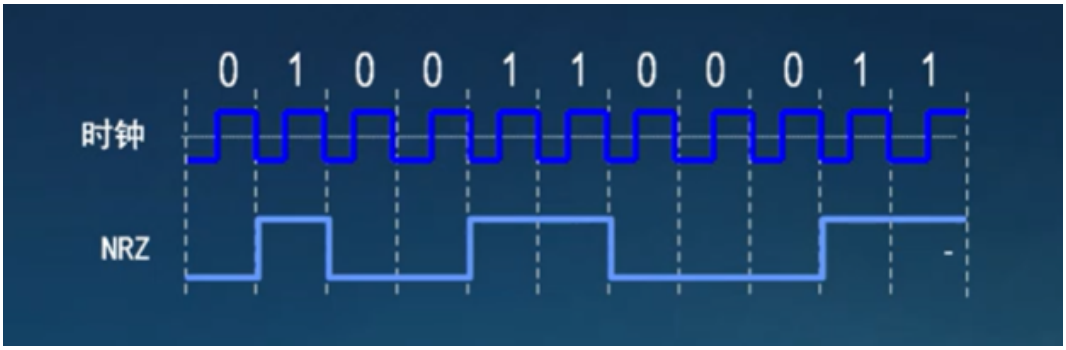
    - 任何实际信道都是不理想的，在信号传输过程中会有外界干扰，有噪声且带宽受限，所以信号会产生失真。如果传输距离很远或者噪声干扰很大，或者传输媒体很差就会导致信号失真严重，接收方无法识别出正确的信号波形，最后出现错误。考虑这种错误的情况，我们采用**信道容量**来衡量一个信道的**数据传输率**。
4. 信道容量：单位时间内信道正确传输的比特数，bps。
    - 香农定理---给出了信道集先传输速率的计算公式：
      $C = Wlog_2(1 + \frac{S}{N})bps$
    - S：信号平均功率
    - N：噪声平均功率

- $\frac{S}{N}$：信噪比
- W：带宽
- 香农定理对如何提高信道传输速率并没有给出明确办法，但意义在于只要信道传输速率低于极限速率，就一定可以找出某种办法来实现无差错的传输。

5. 波特率和比特率是衡量信道传输速率的两个常用单位
   - 波特率是指信号值每秒钟变化的次数，代表码元的传输速率
   - 比特率是指每秒钟传输的数据位数
6. 对于一个带宽确定的信道，如果信噪比不能再提高，码元的传输速率也达到了极限，还有什么方式可以提高数据传输速率？
   - 

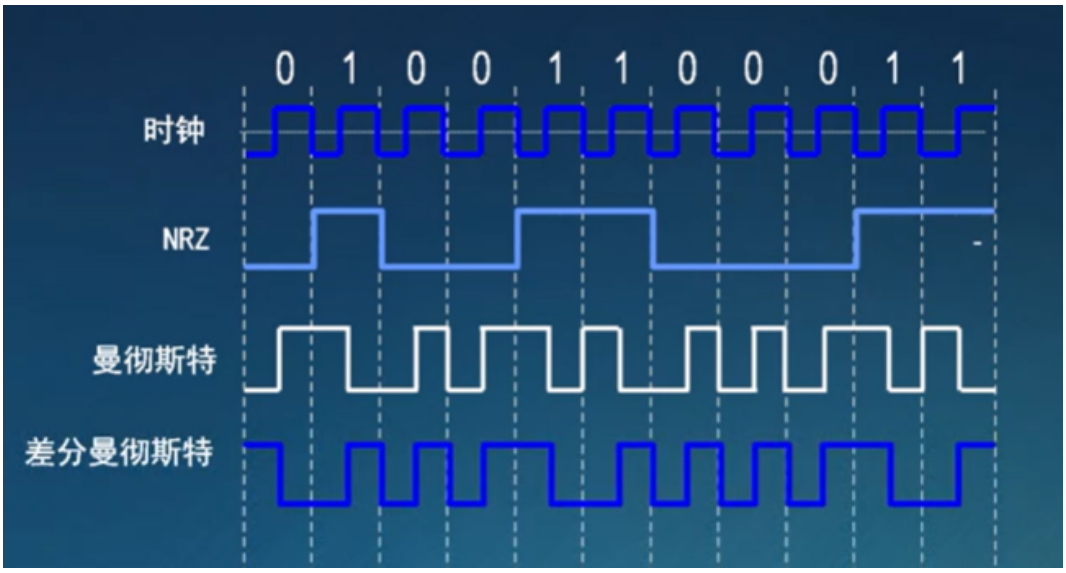
   - 如果将数字信号直接用不同的高低电平传送，每一个码元信号携带的信息量就是1bit，此时波特率和比特率相等。
   - 如果将原始数字数据按照每三位一组进行编码，然后采用某种调制的方法来表示这样的信号，那么每一个调制后的码元信号，携带的信息量就是3bit。如果采用相同速率来传输码元的话，那么编码后的数据传输率就是没有编码时的3倍。由此可见，通过编码的方式，可以提高数据传输速率。
7. 数据：是运输消息的实体；信号：数据的电气的或电磁的表现。
   - 分类上来讲都可以是**模拟的**和**数字的**
   - 模拟型的特点是：数字变化呈连续变化的形式
   - 而数字型的数值则是离散的
8. 数据与信号的分类
   - 模拟数据用模拟信号发送-----载波
   - 数字数据用数字信号发送-----编码
   - 模拟数据用数字信号发送-----采样

    - 数字数据用模拟信号发送-----调制
9. 基带信号：来自信源的信号，比如将计算机输出的数字信号直接用两种不同的电压表示。单机带信号中往往包含较多的低频成分甚至是直流成分，而很多信道不能传输这种低频分量，因此常用调制的方法对基带信号进行处理。其中一种做法是利用载波将基带信号的频率范围调制到另一个较高的频段，以适应信道的传输。
10. 宽带信号：指经过上述调制后的信号，也称带通信号。
11. 利用载波进行调制的方法有三种
    1. 

    2. 调幅：使载波的振幅随信号的不同而变化
    3. 调频：使载波的频率随信号的不同而变化
    4. 调相：使载波的初始相位随信号的不同而变化
12. 基带数字信号的编码方法
    1. 当数字数据用数字信号发送时，**不归零码**是最简单的基带数字信号传输方法。但在传输连续的0或1时，很难确定每个码元的界限，这就会导致接收方无法从比特流中提取出同步的**位信号**。因此需要用某种方法使得接收方和发送方同步。
    2. 

    3. 编码就是一种可以用于同步的方法，常用的编码方法有**曼彻斯特编码**和**差分曼彻斯特编码**，都是通过在码元中间出现电平转换来产生

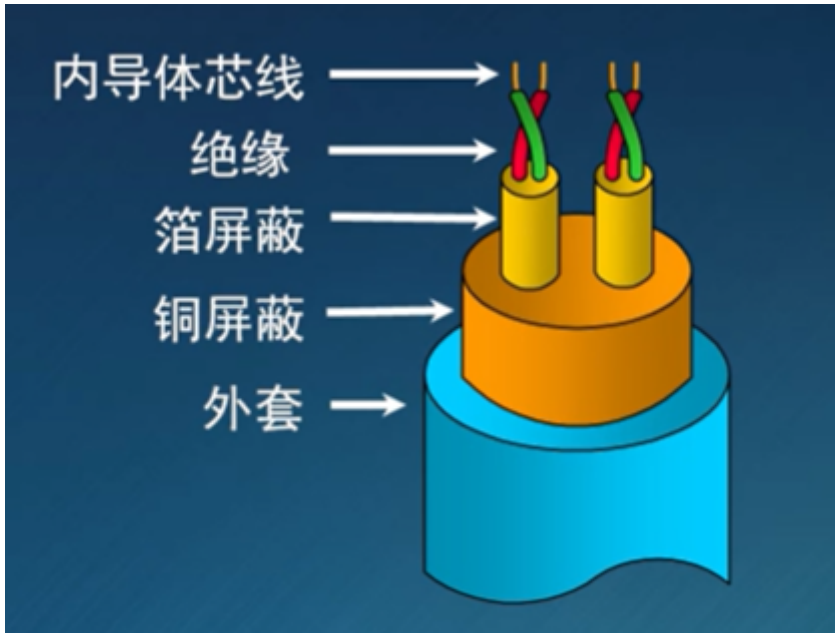
同步信号。

4. 

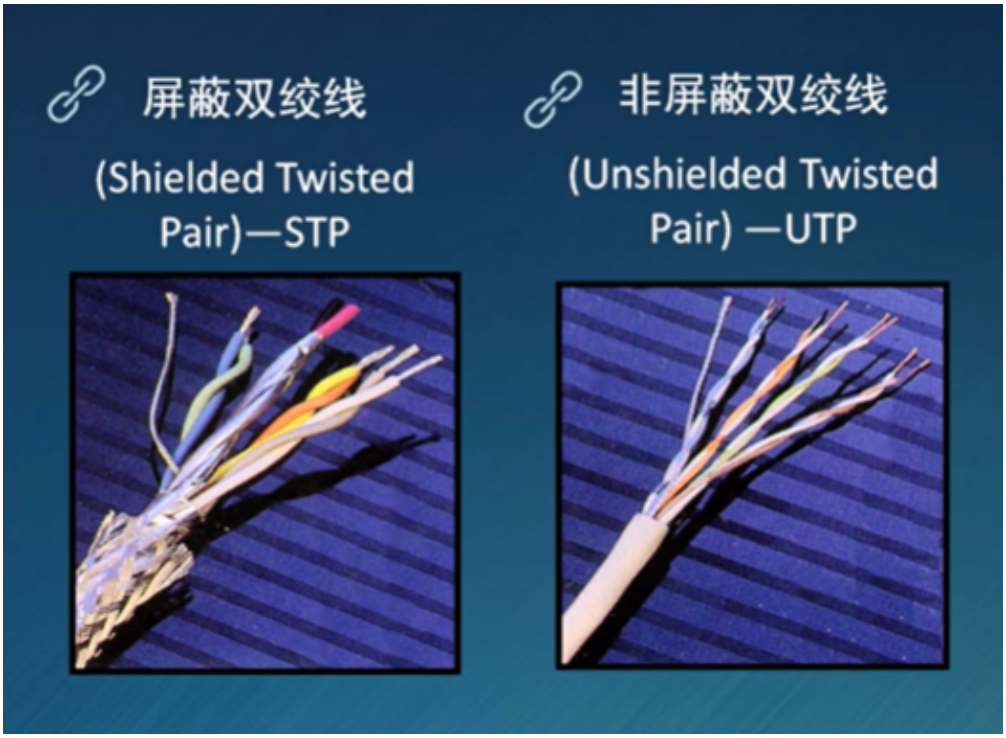
5. 在**曼彻斯特编码**中，每个码元的中间有一次电平的跳变。电平从低变高代表信号0，从高到低代表信号1
6. 在**差分曼彻斯特编码**中，每个码元的中间有一次电平的跳变，同时在信号开始时，不改变电平表示1，信号开始时改变电平表示0
7. 与曼彻斯特编码相比，差分曼彻斯特编码的抗干扰能力更好一些

二、传输媒体

1. 传输媒体时数据传输系统中发送器和接收器之间的物理通路，可以分为两大类，**导向传输媒体**和**非导向传输媒体**。
    - 在导向传输媒体中，电磁波沿着固体媒体（铜线/光纤）传播
    - 非导向传输媒体的传输就是无线传输
2. 导向传输媒体

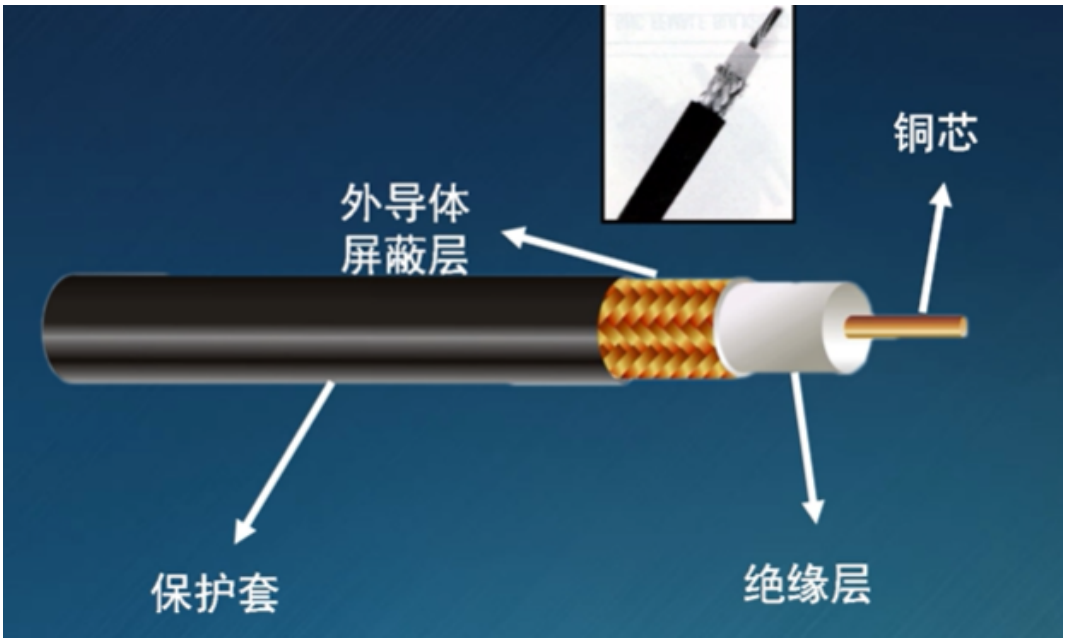
1. 双绞线， 

：最古老但又最常用的传输媒体，把两根相互绝缘的铜导线用规则的方法绞合在一起，就构成了双绞线。

- 模拟传输和数字传输都可以使用双绞线进行，通信距离一般是几公里到十几公里
- 双绞线的绞合可以减少相邻导线的电磁干扰
- 根据绞合的紧密程度，可以将双绞线分为三类线，四类线，五类线，六类线，七类线等等
- 绞合度越高，抗干扰性能越好，获得的带宽也就越宽
- 为了提高双绞线的抗干扰能力，还可以在双绞线的外面加上一层用金属丝编成的屏蔽层---屏蔽双绞线，价格稍高
- 

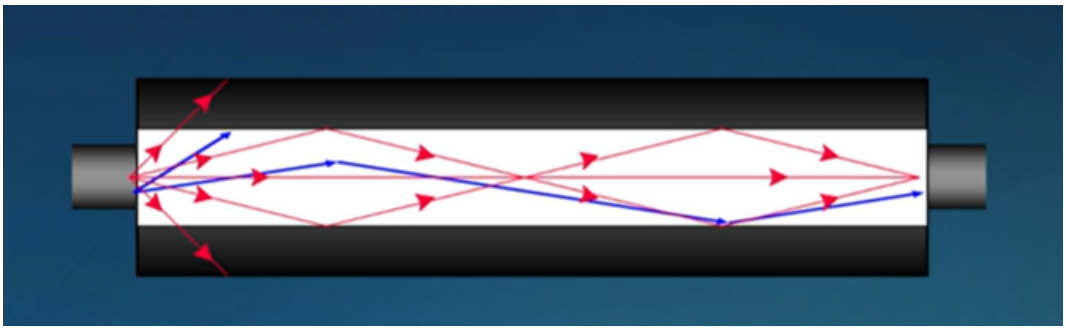
- 由于价格和性能都不错，广泛应用于电话系统和通信系统中

2. 同轴电缆，

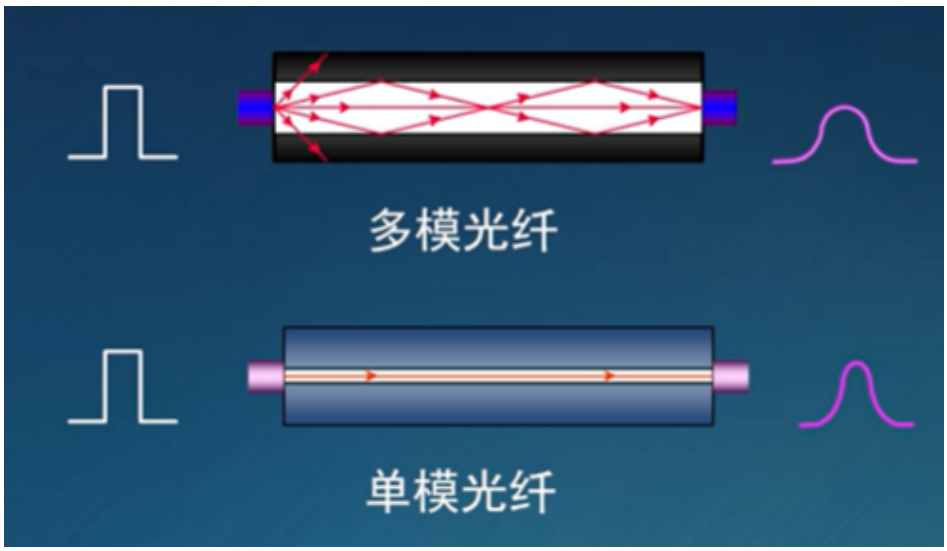


- 由于外导体屏蔽层的作用，同轴电缆具有很好的抗干扰性能，通常用于传输较高速率的数据，主要用于有线电视网的传输中

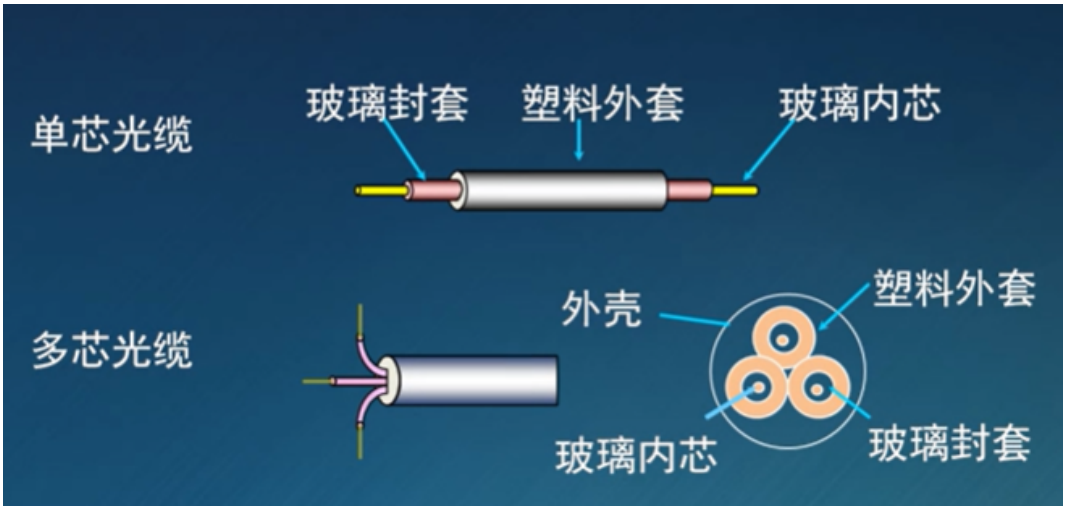
3. 光纤，



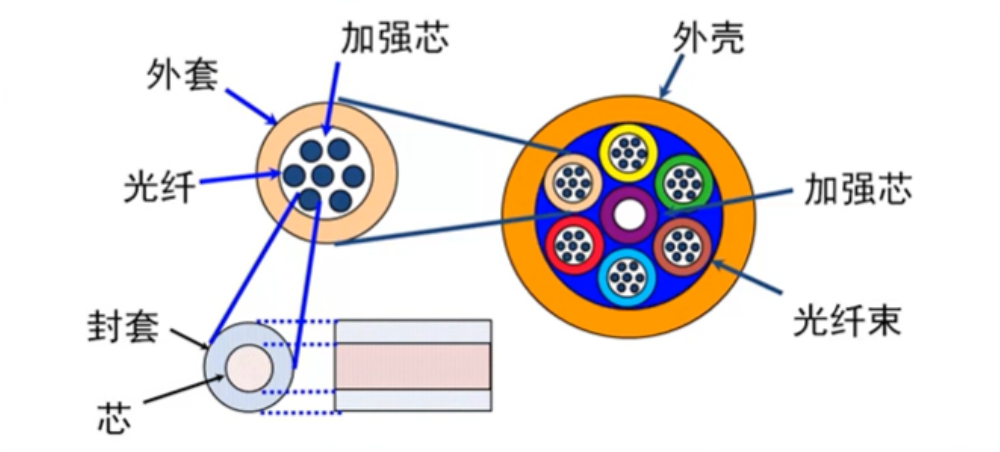
- 非常透明的玻璃芯和包层组成，利用光波在纤芯中传导时，遇到折射率较低的包层而发生全反射现象，将光沿着光纤传输下去
- 光纤包层分为玻璃和塑料两种。玻璃光纤损耗小，成本高，通常用于远距离的宽带传输中。塑料光纤损耗大成本低，通常用于短距离的基带传输中
- 光纤传送模式

1. 

    2. 光纤中只要光纤射在光纤表面的入射角足够大，就能产生全反射现象。因此，在一条光纤中，会存在多条入射角不同的光线，这种光纤被称为多模光纤。
    3. 由于光脉冲在多模光纤中传输时，会逐渐展宽造成失真，所以多模光纤只适合于近距离传输。
    4. 如果将光纤的直径减小到只有一个光的波长，那么光纤就能向波导那样使光线一直向前传输，而不会产生多次反射，这种就是单模光纤。
    5. 单模光纤的制作成本高，光源必须使用半导体激光器，但损耗小，适合长距离传输。
  - 光纤特点
    1. 依靠光波承载信息，衰减少，传输距离远
    2. 抗雷电和电磁干扰性能好
    3. 无辐射，保密性好
    4. 体积小，重量轻
    5. 光纤断裂的检测和修复比较困难
4. 光缆，

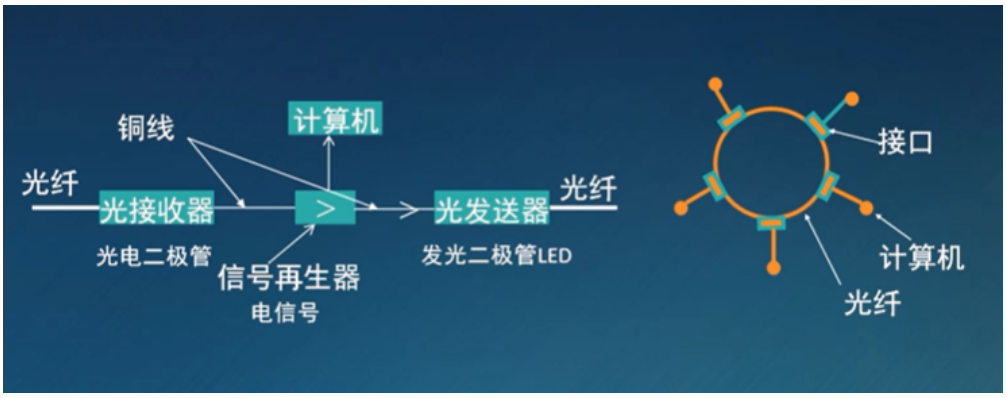


1. 光缆结构：



- 加强芯，填充物，包层和保护套可以是机械强度达到几公斤

2. 光线联网：

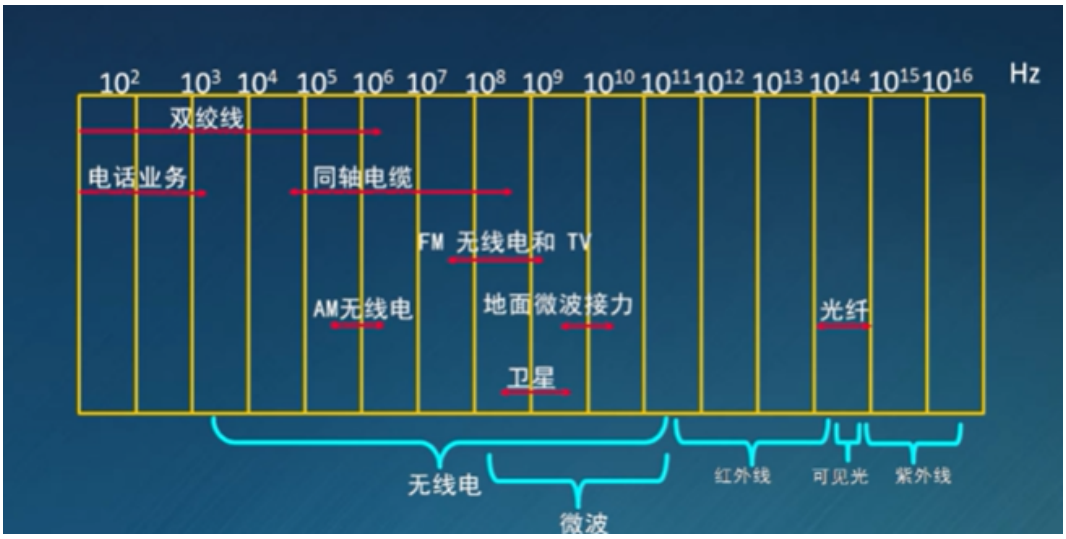


LED

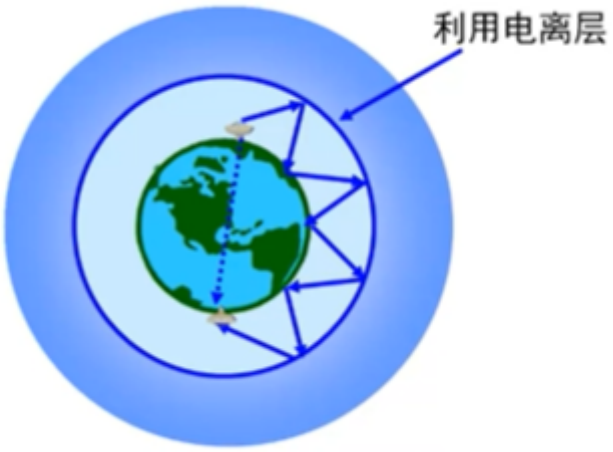
- 将点到点的链路串联起来构成一个环路，通过T形接头连接到计算机上
- T形接头分**无源**和**有源**两种，上面图片的是有源T形接头的内部结构
- 光信号经过**光电二极管**变为电信号，再生放大后，经过**发光二极管LED**变成光信号继续向前传输，相当于一个**转发器**，一旦**T形接头**出现故障，整个光纤环路就会断开

3. 非导向传输媒体

1. 无线传输使用的频带分布：



2. 短波，

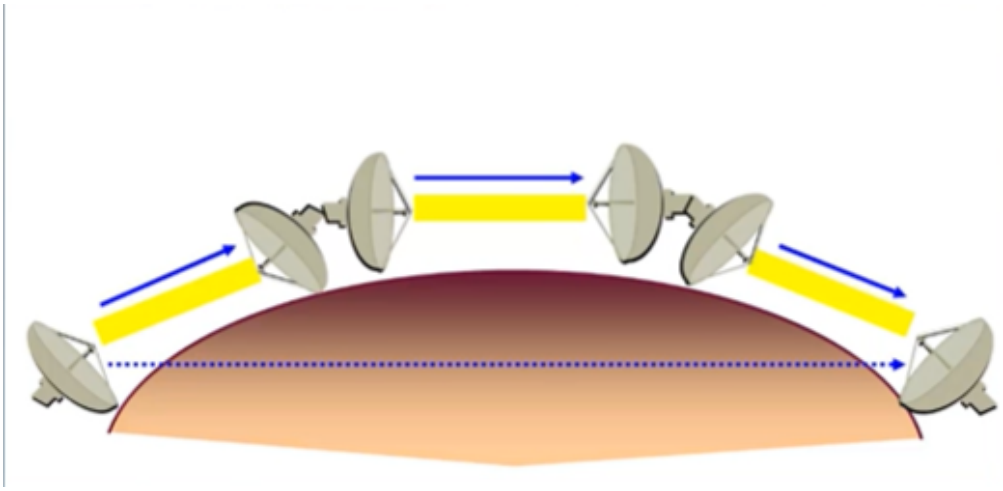


    1. 短波通信主要是利用大气中电离层对无线电波的反射进行传输，电离层的不稳定所产生的衰弱现象和电离层反射所产生的多径效应使得短波通信的质量比较差，因此必须使用短波无线电台进行数据传输时，一般都是低速传输

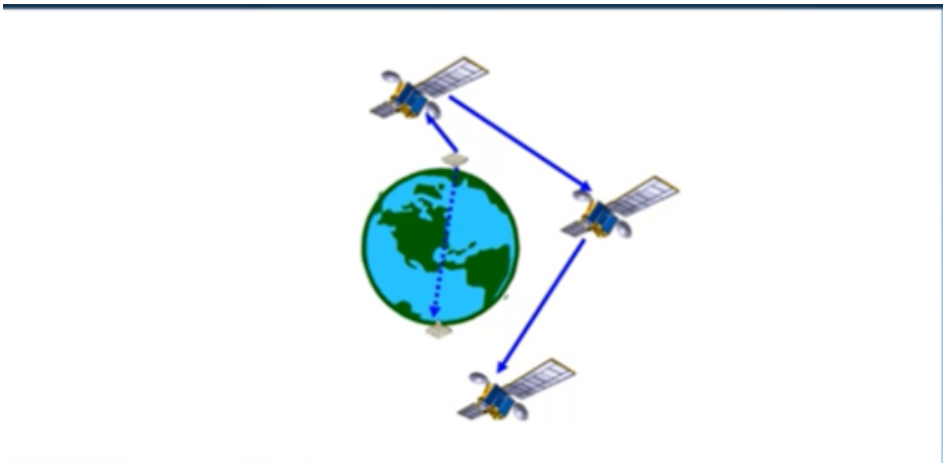
3. 微波

    1. 空间中直线传播，可以穿透电离层进入太空
    2. 微波通信方式主要有两种，店面微波接力通信和卫星通信

3. 地面微波接力通信，



    1. 由于地球表面是球面，微波是直线传输，为了实现远距离传输就在地球表面的高处建立天线塔作为中继站，把前一站发送过来的信号放大后发送给下一站这个过程就是微波接力
    2. 可以传输电话、电报、图像、数据等信息
    3. 主要特点
        - 频带宽，信道容量大，信号受干扰小
        - 建设投入少
        - 相邻站之间不能有障碍物，受天气影响大，保密性差
4. 卫星通信
    1. 利用高空中人造卫星作为中继站进行的微波接力通信，



    2. 特点
        - 通信距离远，适合光波通信，保密性差，与地面微波接力通信特点差不多
        - 在赤道上空一定高度放置三颗卫星可实现全球通信
4. 常用的还有蓝牙技术、红外通信、光波通信等

三、物理层标准举例

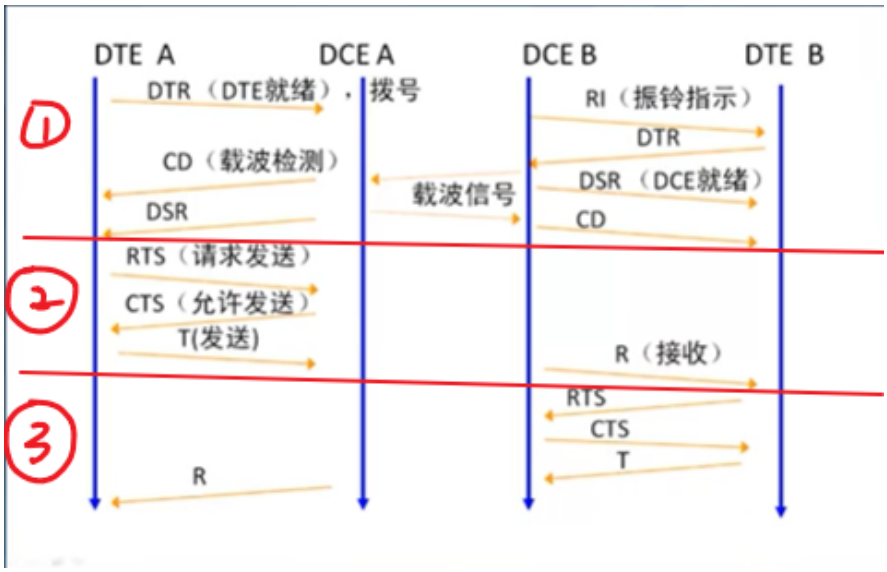
1. EIA-232-E标准
    1. 

    2. DTE（数据终端设备）：具有一定数据处理能力及数据收发能力的设别，如计算机
    3. DCE（数据电路端接设备）：在DTE和传输线路之间提供信号变换和编码功能，并负责建立、保持和释放数据链路的连接，如MODEM（调制解调器）
    4. 232标准是上图中计算机与调制解调器连接时的标准
    5. 232标准规定了物理层的4个特性
        1. 机械特性，规定插头有25针引脚
        2. 电气特性，规定逻辑1用-3V或更低电压来表示，逻辑0用+3V或更高电压表示。两台设备直接相连时最大距离不能超过15米，接口数据传输率不超过20K比特每秒
        3. 功能特性，规定DTE和DCE之间个信号线的功能及连接情况，
        

        4. 规程特性，规定DTE和DCE之间信号持续的应答关系和操作过程
            - 232标准将数据传输过程分为3个阶段
                1. 建立连接阶段
                2. 数据传输阶段

3. 释放连接阶段

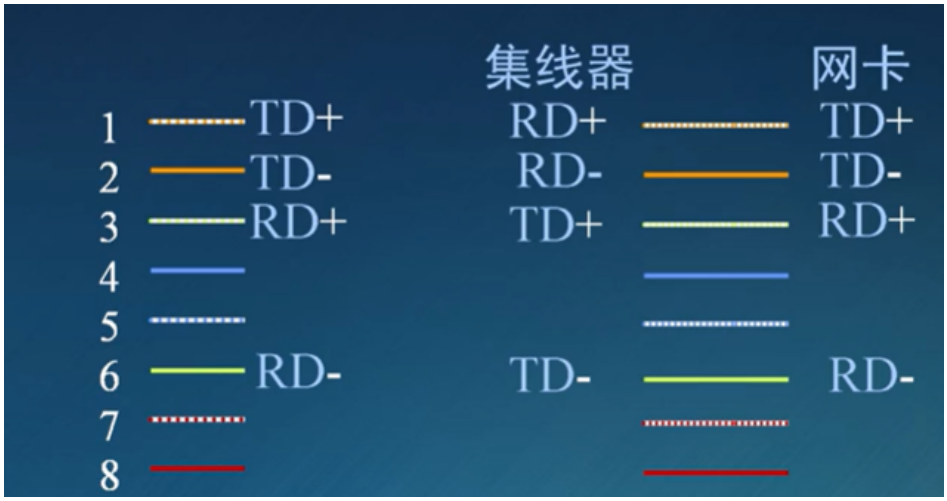
- 具体操作过程

1. 

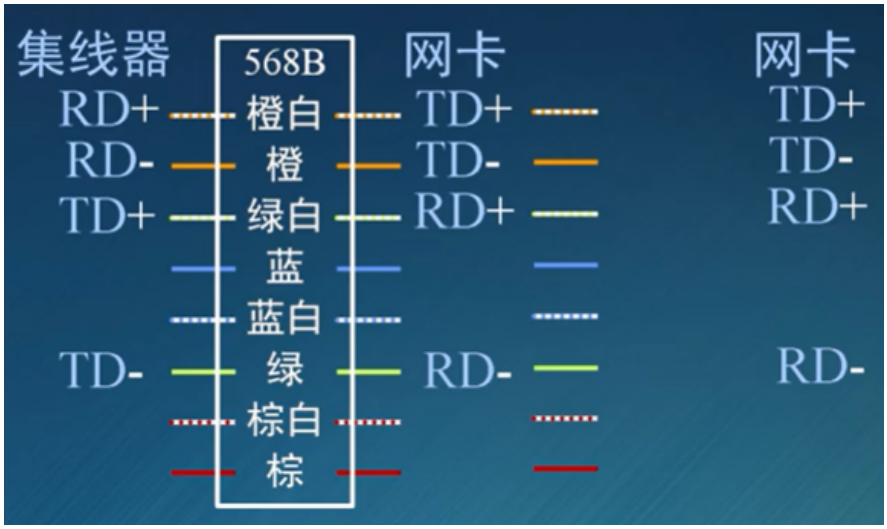
    1. 连接建立阶段：
        - 发送方A的DTE设备首先向DCE A发出DTE就绪信号
        - 然后向DCE B进行拨号
        - 接收方的DCE B以振铃的形式通知DTE B
        - 然后DTE B发出就绪信号通知DCE B
        - DCE B在发出就绪信号响应DTE B
        - 之后将接收方准备就绪的消息以载波的形式发送给发送方
        - 发送方通过载波检测获得了对方准备就绪的消息之后
        - DCE A向DTE A发送就绪信息，同时以载波的形式将发送方就绪的消息告知接收方
    2. 数据传输阶段：
        - A首先发送数据，DTE A向DCE A发出发送请求
        - DCE A回送允许发送信号
        - 数据从DTE A发送出来，被DTE B接收
        - 接下来B发送数据和上面一样

2. 568标准：美国电子工业协会和电信行业写回联合发布的一个关于双绞线的标准（下面重点为双绞线线序标准）
    1. 双绞线中一共有8根信号线，按照标准的线序插入到RJ45的水晶头中

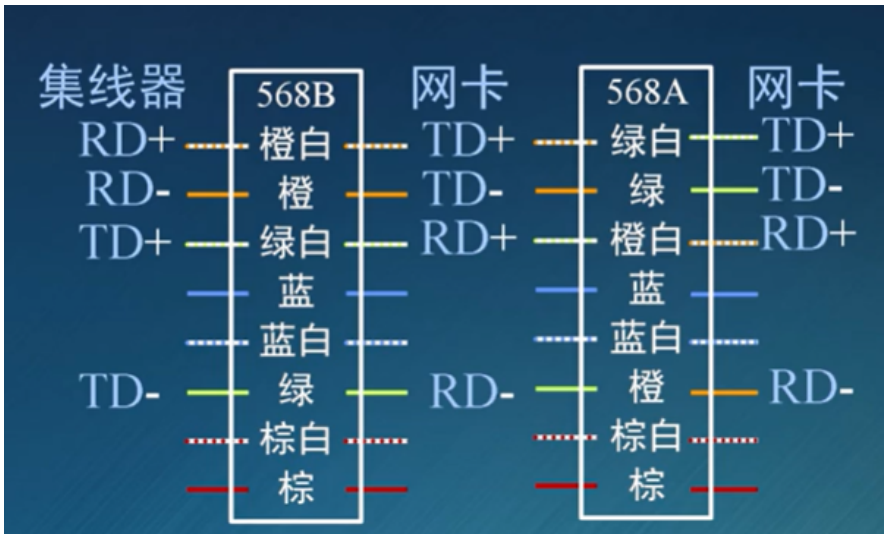
2. 568B标准：

1. 大多数情况下四个引脚传输数据，另外四个作为备用引脚
2. 1号引脚：发送高电平数据
3. 2号引脚：发送低电平数据
4. 3号引脚：接收高电平数据
5. 6号引脚：接收低电平数据
6. 使用双绞线的设备，其接口的功能特征和双绞线的功能特征一一对应
    1. 

    2. 要实现计算机与集线器之间的数据传输的话，集线器接口的功能特征就要与双绞线的功能特征在发送和接收上刚好相反的，这样才能左到计算机发送的数据被集线器接收，集线器发送的数据也能被计算机接收
    3. 考虑到兼容性，目前大多数网线都是按照568B的标准制作的

- 

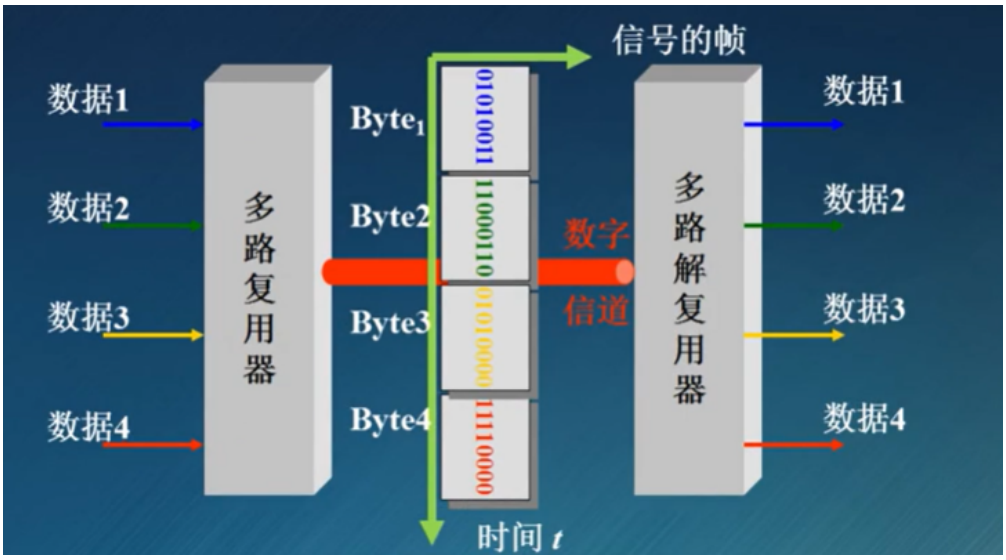
4. 但是如果我们要用网线将两个相同的设备连接起来（如两台计算机），由于同类设备接口功能特性相同，如果采用两端线序相同的网线直连的话，就没有办法将数据的发送引脚和对方的接收引脚对应上，也就不能实现数据传输了。因此根据网线两端连接设备的不同，网线分为**直通线**和**交叉线**
    1. 直通线：网线两端采用相同的线序
    2. 交叉线：网线两端的线序刚好相反，

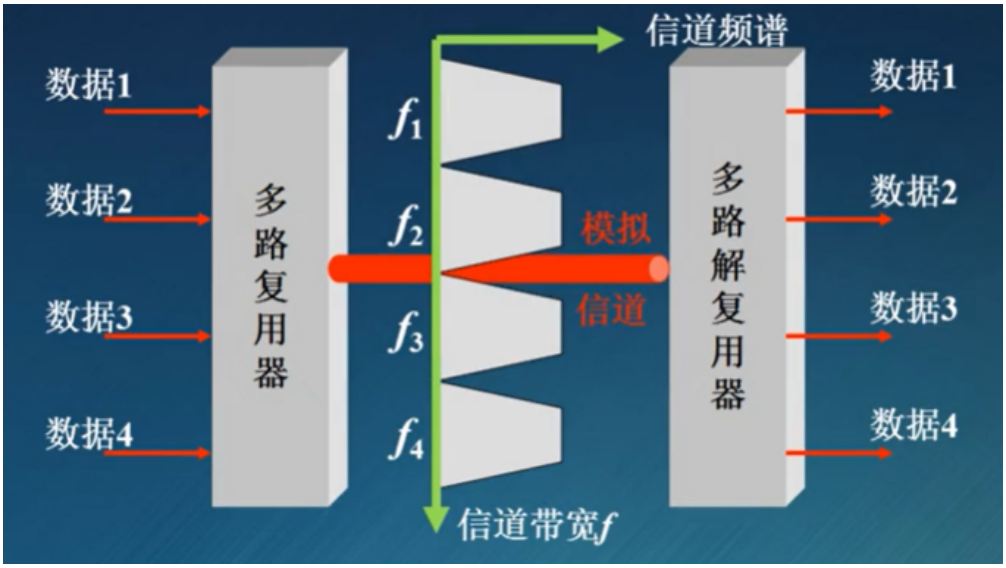


，后面新产生的标准也就是568A标准

    3. 实际应用中关于直通线和交叉线的使用：**不同类型的设备之间采用直通线，相同类型的设备采用交叉线**这个原则
        - 类型主要是按照DCE和DTE进行划分的，常见的网络设备中，计算机和路由器属于DTE设备，交换机和集线器属于DCE设备

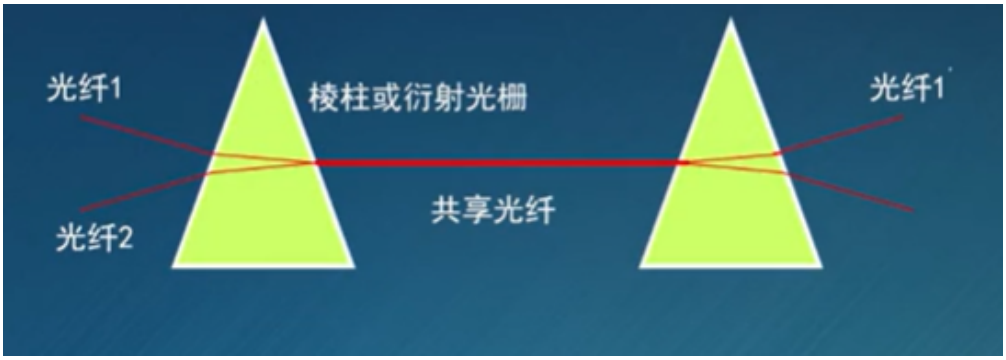
四、信道复用技术

1. 信道复用技术就是通过复用的手段让多个用户共享一个信道进行通信，这样可以充分利用信道资源，降低通信成本。
2. 按照复用实现的方式上可以划分四类
    1. 时分复用
        1. 

        2. 特点
            - 将时间划分成一个个等长的时分复用帧，每一个时分复用帧的用户在每个时分复用帧中占的固定序号的时系
            - 如图中有1，2，3，4四个用户，每个用户所占用的时系是周期性出现的，这个周期就是时分复用帧的长度。
            - 数据在发送方通过多路复用器实现对高速信道的复用，之后在接收方再复用多路解复用器对数据进行分用，分别送给相应的用户
            - 时分复用的所有用户是在不同的时间占用同样的频带资源
    2. 频分复用
        1. 

        2. 特点

- 将信道的带宽划分为小的频带，为每个用户分配一个固定的频带，之后这个用户就始终占用这个频带进行通信
- 可见频分复用的用户是在相同的时间占用不同的带宽资源

3. 波分复用
    1. 光纤通信中，在同一根光纤中同时传输具有不同波长的光载波信号技术
    2. 波分复用为每个用户分配相互隔离的光谱频带，所以也称为光的频分复用
    3. 由于光信号的频率非常高，所以人们习惯上用波长而不是频率来表示光信号，因此也就有了波分复用这个名称
    4. 

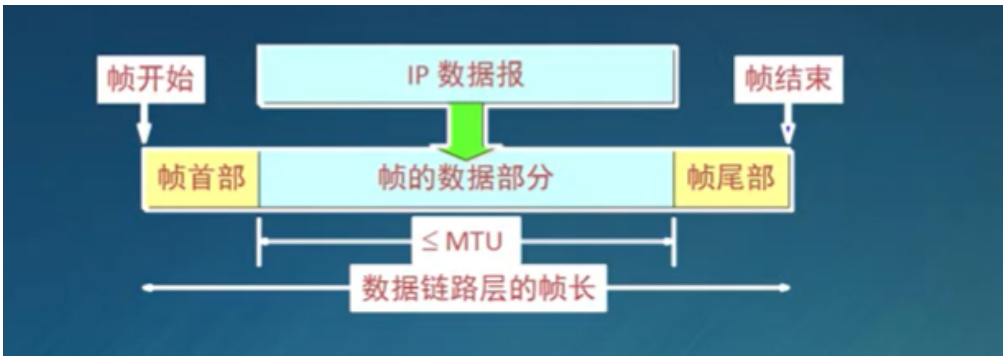
    5. 在发送端具有不同波长的光信号经过棱柱或衍射光栅汇总在一起，并耦合到一根光纤上进行传输，到了接收端再经过棱柱或衍射光栅分离出来
4. 码分复用
    1. 更常用的名称是码分多址，简称CDMA
    2. 特点
        - 每个用户可以在相同的时间使用同样的频带进行通信，由于个用户使用经过特殊挑选的不同码型对发送的数据进行编码，所以个用户之间不会造成干扰
        - 码分复用最初用于军事通信，因为信号具有很强的抗干扰能力，其频谱类似白噪声，不易被敌人发现，随着价格和体积的降低，码分复用逐渐普及
    3. CDMA工作原理
        1. CDMA系统中，每个用户共享全部的时间和带宽资源，所以数据必然会在信道中碰撞叠加，但在接收方通过利用发送方选择的特殊码型对数据进行内积计算就能过滤除掉除了发送站以外的其余各站的数据信息，这样就能将个用户的信息分离开来
            - 经过运算后，将其他站的信息过滤掉

- 利用内积运算进行过滤

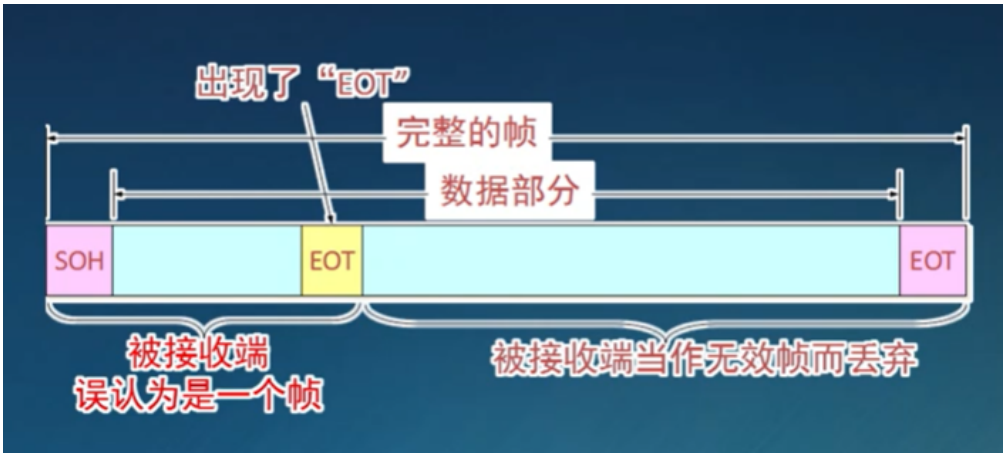
2. 具体实现
    1. 将每个比特时间再细分为m个短间隔，每个短间隔称作码片（chip），一般m=64或128，这里为了说明方便，令m=8
    2. 对每个站制定唯一的码片序列
        - 一个站要发送比特1就要发送它的m比特的码片序列，如果要发送比特0就要发送它的m比特的码片序列的反码序列
        - 例如给某站分配的码片序列是00011011，发送1时就发送00011011，发送0时就发送11100100
    3. 重要特点：系统中每个站的码片序列不同且正交
        1. 如果用S和T分别表示两个站的码片序列，那么有：
            - 正交：S与T向量内积为0，即 $\vec{S} \cdot \vec{T} = \frac{1}{m} \sum_{i=1}^{m} S_i \times T_i = 0$（这种情况下编码中的0是-1，1是1来表示）
        2. 任何一个码片向量和该码片向量自己的规格化内积都是1，任何一个码片向量和该码片向量的反码向量自己的规格化内积都是-1
    4. 在CDMA系统中，利用全球定位系统GPS可以使每个站发送的码片序列都是同步的。系统中的每个站必然会处在下面三种状态之一
        - 发送码片序列相当于发送比特1
        - 发送码片序列的反码序列就相当于发送比特0
        - 或者什么都不发
    5. 对于每个站的状态，在接收端可以利用该站的码片序列和收到的数据进行内积运算推断出来
        - 接收站使用S站的码片序列与接受到的向量做内积运算，必有：
            1. 所有其他站的信号都被过滤掉（内积为0）
            2. 运算结果为+1：S站发送1
            3. 运算结果为-1：S站发送0
            4. 运算结果为0：S站没有发送数据

三、数据链路层

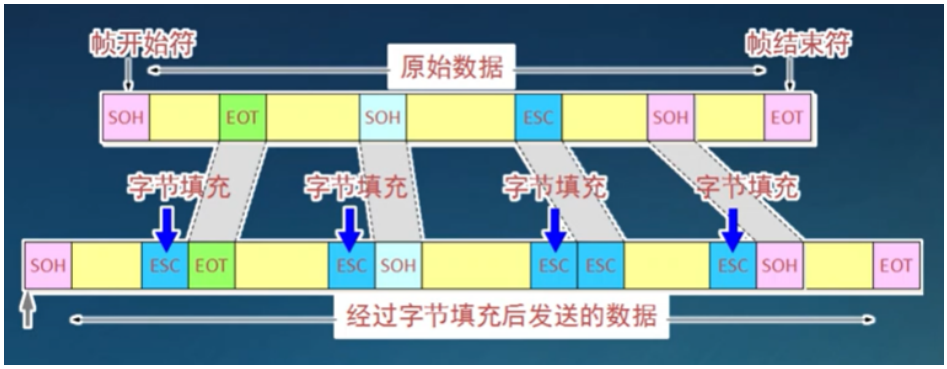
一、数据链路层的基本概念

1. 链路：一条无源的，点到点的物理线路段，中间没有任何其他的交换节点（链路是通信通路的一个组成部分）
2. 数据链路：物理链路+链路控制规程幸成的数据管道
3. 数据链路层的三个基本功能
    1. 分装成帧
        1. 封装成帧（framing）就是在一段数据前后分别添加首部和尾部，然后就构成了一个帧
        2. 首部和尾部的一个重要作用就是进行帧定界，此外首部和尾部还包括很多控制信息
        3. 

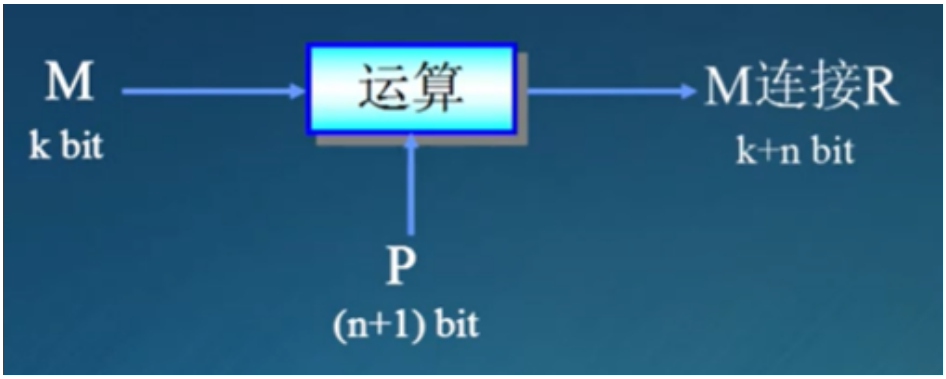
        4. 帧的发送是从首部开始的，到尾部结束。为了提高数据传输效率，应该使帧的数据部分长度尽可能地大于首部和尾部的长度，但是每种数据链路层协议都对帧的数据部分长度设置了上限，我们把这个值称为**最大传输单元**，简称MTU。
    2. 透明传输
        1. 由于帧的首部和尾部通常会采用一些特殊字符和比特组合来作为帧开始和结束的标记，所以在帧的数据部分就不允许再出现与这些定界符号相同的内容，否则就会出现帧定界的错误
        2.



3. 上图中，数据部分出现了与帧结束部分相同的内容，接收方就会误认为这是真的结尾而把它接受下来，然而真正的帧并没有传输结束，后续的数据部分就只能成为一个无效帧被丢弃。
4. 为解决上述这个问题，在**发送方的数据链路层**就需要对数据部分出现的与帧定界符相同的字符进行处理。比如我们可以在这些字符前面插入一个转义字符，接收方在接收时，再将这些转义字符去点，同时忽略其后面字符的含义。如果转义字符也出现在数据部分，那么就在转义字符前面也插入一个转义字符，接收方收到连续的两个转义字符后，删掉前面一个就好



3. 差错控制
    1. 数据在传输过程中受到噪声和外界环境的干扰，可能就会出现比特差错，也就是0变成了1，1变成了0。为了保证数据的可靠性，在计算机网络的传输过程中，必须采取各种差错检测的措施，其中**循环冗余校验（CRC）**是数据链路层上广泛使用的一种差错检验方法
    2. 循环冗余校验（CRC）的原理

- 

1. 在发送端，先把数据划分成组，假定每K个比特
2. M是要发送的数据（K bit），运算就是将M与一个n+1位的除数P进行除法运算，得到n位余数R
3. 将R连接再数据M后面，是用于差错检验的冗余码，它将和M一起构成一个帧发送出去
4. CRC进行差错检验的数学基本原理：
    1. 若`m / p = n 余 r`，则有`(m-r)能被p整除`。CRC中除法采用的是**模2除法**，即加法不产生进位，减法不产生借位，所以模2运算的加减法运算结果是相同的，由此可知，如果之前的除法采用模2除法一定有`(m+r)/p余数为0`
    2. 基于上述原理，CRC在进行计算式，先对原数据$M \times 2^n$，相当于在M的后面添加n个0，然后再与P进行模2除法运算，最后把运算得到的R附加在M的后面，相当于$M \times 2^n + R$

3. 在数据后面添加的冗余码，称为帧检验序列FCS（Frame Check Sequence），通常是数据帧当中用于差错检验的一个字段。
    - 帧检验序列和CRC的冗余码并不是同一个概念
    - 帧检验序列可以通过CRC计算得出，但这不是唯一的方法，还有其他方法也可以
4. 利用CRC进行检验的工作是在接收方完成的，接收方将收到的每一帧数据再与除数P进行模2除法运算
    - 如果余数为R = 0，则判定这个帧没有差错，接收
    - 如果R != 0，则判定这个帧有差错，就丢弃
    - CRC只能检测是否有错误，但不能确定错误出现的位置。同时也可能数据在传输过程中出错了，但凑巧的是，这个错误的帧经过计算余数为0，出现对错误的漏检，这种可能性在

理论上是存在的，但是只要经过严格的挑选，并使用位数足够多的除数P，这种概率就会很小很小

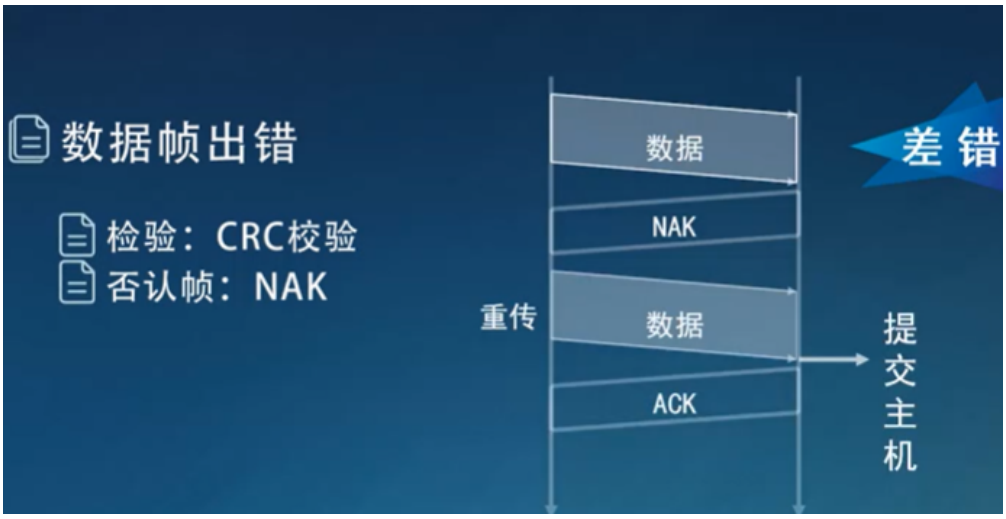
- 为了表示方便，除数P采用生成多项式表示
- 仅使用循环冗余检验技术只能做到无差错接收，但是“无差错接受 != 可靠传输”，因为数据传输过程中还可能出现帧丢失，帧重复，帧乱序等问题，所以还必须加上**确认和重传机制**才能保证可靠传输

二、数据链路层的可靠传输

1. 在OICI参考模型中，关于数据链路层功能的定义是：通过一些数据链路层协议，在不太可靠的物理链路上，实现可靠的数据传输
2. 停止等待协议
    1. 基本原理
        1. 正常情况，

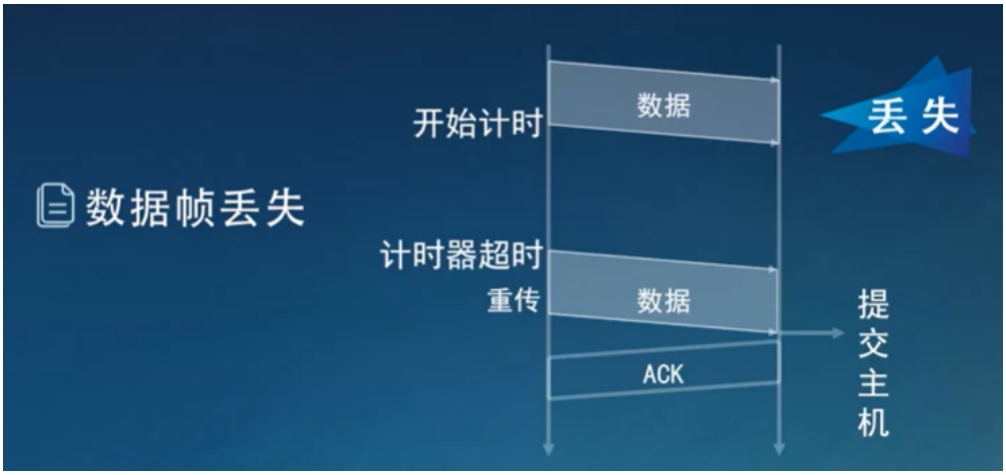
- 接收方接受到发送发发来的一帧正确的数据后，将其提交给主机
- 然后向发送方发送一个**确认帧ACK**
- 发送方收到确认帧之后，才能发送下一帧数据

2. 数据帧出错，



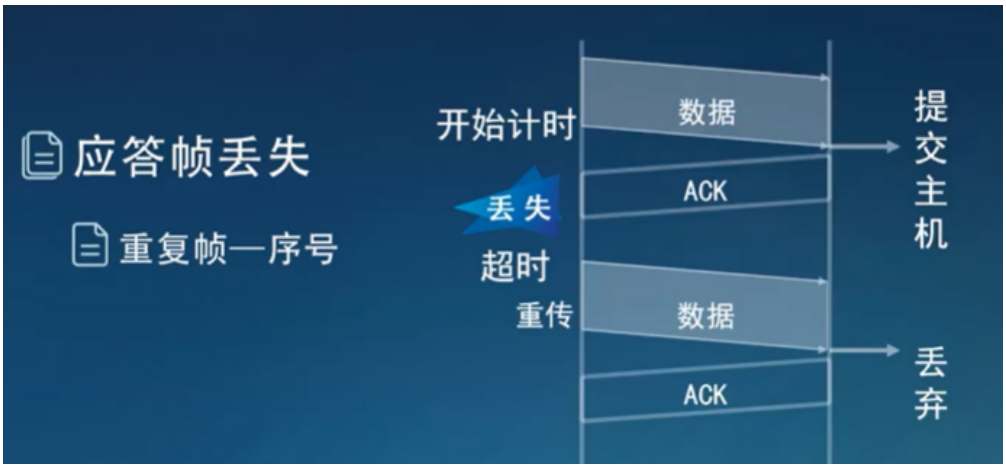
- 如果传输过程中数据出现差错，接收方收到数据后，通过循环冗余校验可以很容易的发现错误
- 此时接收方向发送方发出一个**否认帧NAK**
- 接下来发送方会重传这一帧数据，知道收到确认帧为止
- 这种处理方式交错**差错重传**

3. 数据帧丢失，



- 数据帧在传输过程中可能会丢失，由于接收方收不到数据，所以不会发出确认帧。而发送放如果一直等待下去就会出现**死锁**
- 为了防止这种情况的出现，发送方没法送出一帧数据，都要启动一个超时计时器
- 如果在计时器所设置的时间内，没有收到接收方发来的确认帧的话，就重传之前发出的这一帧数据，知道收到确认帧为止
- 这种处理方式称为**超时重传**

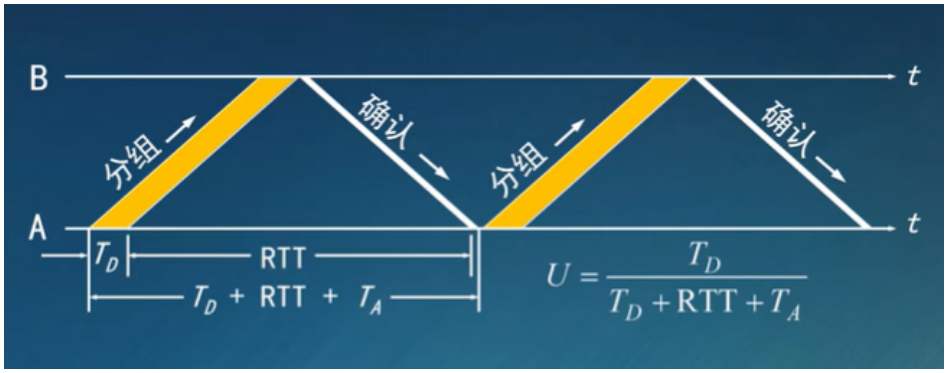
4. 数据帧被接收方接收了，但确认帧却丢失了，



   - 由于确认帧丢失，发送方也会进行超时重传，但**问题是接收方已经接受过这一帧数据了**，这样就会出现重复帧的问题
   - 为了解决重复帧的问题，我们可以给每一个发送的不同的数据帧都带上不同的序号，根据序号，接收方就可以很容易的判断重复帧
   - 当收到重复帧时，接收方将重复帧丢弃，同时向发送方发送一个确认帧以防止发送方在超时重传
5. 由于上述确认和重传机制就可以在不可靠的通信链路上实现可靠的数据传输。由于发送方对出错的数据帧的重传是自动完成的，所以这种可靠传输协议又称为ARQ（Automatic Repeat Request）

2. 信道利用率

   1. 停止等待协议的优点是简单易实现，但是缺点是信道利用率太低

      - 

        1. 假设发送方发送一帧数据的时间是T_D，数据在发送方和接收方之间的信道上往返的传输时间是RTT，发送方接收确认帧的时间是T_A，从上图中可以看出发送方在每一帧数据发送的过程中有效数据占用信道传输的时间仅为T_D，而整个过程需要的时间远大于T_D，所以停等协议的信道利用率很低，从图中发现这个问题出现的原

因是**发送方在等待接收方确认的过程中信道一直出于空闲状态**

2. 为了解决上述问题，出现了连续ARQ协议

3. 流水线传输（连续ARQ协议）

1. 采用流水线传输思想，即发送方可以连续发送多个分组，不必每发完一个分组就停下来等待对方的确认
2. 由于信道上一直有不间断的数据传送，所以这种传输方式可以获得很高的信道利用率
3. 连续ARQ原理
    1. 边发送边接收确认
    2. 帧编号
        1. 由于连续发送，则需要连续编号
        2. 确认帧也需要编号
    3. 接收端接受到有差错的数据帧的处理方式
        1. 向发送端发送否认帧
        2. 不响应，通常选用这种，实现起来更方便✔
    4. 接收端采用只按照序接收的工作方式------Go-back-N（回退N）
        1. **如果前面序号的数据帧出错或丢失了，那么其后面的数据帧只能被接收方丢弃**
        2. 因此如果发送方因计时器超时对某一帧数据进行重传的话，除了要重传这一帧数据外，还要把在超时之前发送的所有没有接受到确认帧的数据都重传一遍（称为回退N）
        3. 回退N会严重影响传输效率。在信道传输质量不好的情况下，甚至不如停等协议
    5. 接收端采用**累积确认**的方式
        1. 为了减少开销，接收方不必对每个接收到的分组都发送一个确认，而是对按序接收到的最后一个帧发送确认，这样就表示到这个分组为止的所有分组都已经正确收到了

6. 使用连续ARQ协议时，发送方在等待确认帧的过程中，实际上并不能无限制的发送数据帧。因为一旦有数据帧出错，那么发出去的数据帧越多，需要回退重传的数据帧也就越多，重传的开销也就越大；而且为了给发出去的大量数据帧编号，就需要占用较多的比特位，进而增加了更多的传输开销，因此引入**滑动窗口机制**，对发送发已经发送出去但没有确认的数据帧的数量加以限制很有必要

4. 滑动窗口机制
    1. 窗口：就是指在发送方和接收方分别设置的可以移动的发送窗口和接收窗口。通过对窗口的设置可以对已经确认的帧的序号进行循环利用，同时加入适当的控制机制来避免二义性
    2. 发送窗口的作用

        1. 对发送方的流量加以控制，只有在发送窗口内的帧才能被连续发送出去。
            - 比如连续ARQ中发送序号占3比特，就有8个不同的序号从0到7。
            - 又假设窗口大小为5，那么刚开始时只有0到4号帧可以被发送出去
            - 之后在没有收到任何确认的时候，窗口停止不动，发送方也停止发送
            - 当收到0号帧的确认后，窗口向前移动一个号，此时5号帧就落在了窗口里，可以发送出去了
            - **随着确认帧逐渐被收到**，发送窗口逐渐向前移动，更多的数据帧逐渐被发送出去
    3. 接收窗口的作用

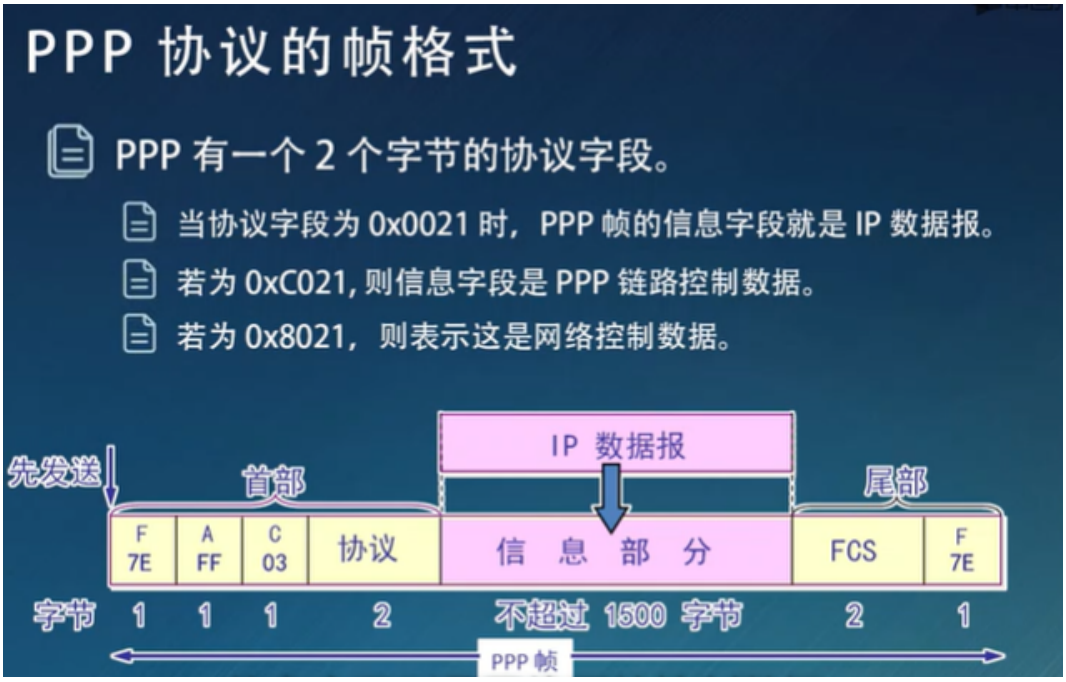
1. 控制哪些帧可以接收。接收方只有接收到发送序号在接收窗口里的数据帧时才能将该帧收下，否则一律丢弃
2. 在连续ARQ协议中接收窗口的大小为1
    - 当收到0号帧后，窗口向前滑动1个号，准备接收1号帧，同时发送对0号帧的确认
    - 随着数据帧逐渐被按序接收，接收窗口逐渐向前移动
    - **当发送窗口和接受窗口的大小都为1时，就是最初讨论的停等协议**
    - 为了提高对信道的利用率，我们可以设法只对出现的差错或丢失的数据帧进行重传，但是此必须加大接收窗口，先收下那些发送序号不连续但在接收窗口内的数据帧，等所有窗口内的帧都到达后，在一起提交，这就是**选择重传ARQ。**
        1. 可以避免重传已经确认的帧，但付出的代价是需要在接受方设置相当容量的缓冲区

三、PPP协议（一种点对点协议）

1. PPP协议应满足的需求
    1. 简单：首要要求
        - 因为因特网工程任务组在设计因特网体系结构时，把最复杂的部分放在了运输层的TCP协议中。而网络层的IP协议相对简单，只提供不可靠的传输服务。在这种情况下，数据链路层的协议就没有必要比IP协议更复杂
    2. 封装成帧
        1. PPP协议必须规定特殊的字符作为帧定界字符，以便于接收方从比特流中提取出完整的数据帧
    3. 透明性
        1. 如果数据中出现与帧定界符相同的比特组合，必须采取一定的措施来解决，保证数据的透明传输
    4. 多种网络层协议
        1. 要能够在一条物理链路上同时支持多种网络协议的运行
    5. 多种类型的链路

    1. 串/并行、低/高速、同/异步
    2. PPP协议必须能够在多种类型的链路上运行
6. 差错检测
    1. 必须能够对接收方收到的数据进行差错检测，并丢弃有差错的帧，以防止差错帧继续在网络中传输造成资源浪费
7. 检测连接状态
    1. 要能够及时自动检测出链路是否处于正常工作状态
8. 最大传送单元（MTU）
    1. 必须对每一种类型的**点对点链路**设置最大传输单元MTU的默认值
9. 网络层地址协商
    1. 必须提供一种机制使通信的两个网络层实体能够通过协商知道或配置彼此的网络层地址
10. 数据压缩协商
    1. 要提供用来协商数据压缩算法的办法
11. 此外，在标准中还明确了PPP协议不需要的功能
    - 纠错（只检错，不纠错）
    - 流量控制
    - 序号（不需要使用帧的序号）
    - 多点线路（不支持多点线路）
    - 半双工或单工链路（只支持全双工链路）

2. PPP协议的三个组成部分
    1. 一个将IP数据报封装到串行链路的方法
    2. 一个用来建立配置和测试数据链路连接的链路控制协议LCP（Line Control Protocol）
    3. 一套用来支持不同网络层协议的网络控制协议NCP（Network Control Protocol）
3. PPP协议的帧格式
    -

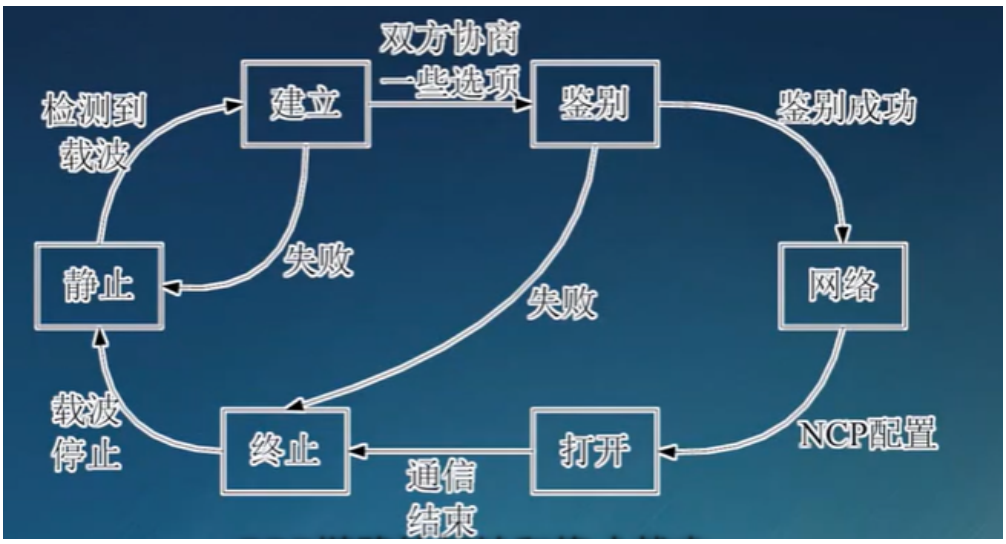


- 首部和尾部部分分别为4个字段和2个字段。首部的第一个字段和尾部的最后一个字段都是标志字段，规定值是十六进制的`7E`，二进制形式`01111110`，表示一个帧的开始和结束，也就是帧的定界符。首部中的地址字段A设置为十六进制的`FF`，控制字段设置为十六进制的`03`，事实上这两个字段并没有真正的意义
- PPP首部的第四个字段是一个协议字段，占2个字节
  - 当协议字段为0x0021时，PPP帧的信息字段就是IP数据报
  - 当协议字段为0xC021时，PPP帧的信息字段就是PPP链路控制数据
  - 当协议字段为0x8021时，PPP帧的信息字段就是网络控制数据
- 信息段的内容就是PPP协议要封装的主体部分，通常是网络层交下来的IP数据报，长度可变，但最长不能超过1500个字节
- 在PPP帧的尾部，第一个字段是使用循环冗余校验的**帧校验序列**，占2个字节

4. PPP协议对透明传输问题的处理
   1. PPP协议有两种实现透明传输的办法
      1. PPP用在**异步传输**时，就使用一种特殊的字符填充法
      2. PPP用在同步传输时，规定采用硬件来完成0比特的填充
   2. 字符填充法
      1. PPP协议将十六进制的`7D`定义为转义符，使用它来进行字符填充

2. 将信息字段中出现的每一个0x7E字节转变成2字节序列（0x7D，0x5E）
3. 若信息字段中出现一个0x7D的字节，则将其转变为2字节序列（0x7D，0x5D）
4. 若信息字段中出现ASCII码的控制字符（即数值小于0x20的字符），则在该字段前面要加入一个0x7D字节，同时将该字符的编码加以改变
5. 接收端收到数据帧后进行反向操作即可

3. 0比特填充法
    1. 在发送端先扫描整个信息字段，**只要发现5个连续的1**则立即填入一个0。这样就能保证在信息字段中不会出现6个连续的1，也就不会出现与标志字段相同的比特组合。
        - `10110111110100` -> `101101111100100`
    2. 接收端在收到数据帧后，先找到标志字段确定帧的边界，然后再用硬件对其中的比特流进行扫描，每当发现5个连续的1，就将其后面的一个0删掉，这样就能还原原来的比特流
        - `101101111100100` -> `10110111110100`

5. PPP协议的工作过程
    1. PPP协议最典型的应用就是用户通过向ISP拨号介入Internet的过程。在这个过程中PPP的工作可以用下面状态图来描述：
        - 

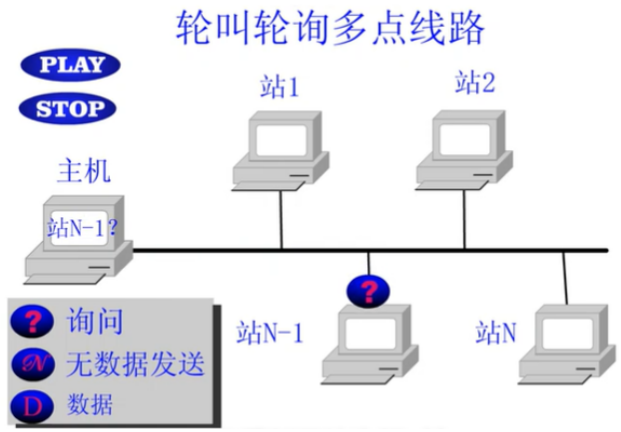
        - PPP链路的起始和终止状态永远是图中的静止状态，此时在用户的PC机和ISP的路由器之间是不存在物理连接的
        - 当用户向ISP的路由器发出呼叫时，路由器就能检测到呼叫的载波信号
        - 在双方建立了物理层连接后，PPP就进入链路建立状态。它的目的时建立链路层的LCP连接。这时LCP通过请求配置帧，开始

协商一些配置选项，包括链路上最大帧长、所使用的鉴别协议等等
    - 协商结束后，双反就建立了LCP链路
    - 接着就进入鉴别状态，这一状态主要是对双方的身份进行鉴别。如果鉴别失败，就转到静止状态；如果鉴别成功就转到网络状态
    - 在网络状态下，PPP的双方通过网络控制协议NCP，根据网络层的不同协议交换特定的网络控制分组，用来协商网络层的相关配置
    - 配置完毕后，链路层就进入了可以进行数据通信的打开状态，链路两端的用户就可以相互发送数据了
    - 数据传输结束后，一端发出终止请求，在收到对方的终止确认后，就进入到链路终止状态
    - 当线路上的载波停止后，就回到了静止状态

四、信道共享技术

一、信道共享技术之受控接入

1. 在多点接入共享信道上，有不同用户同时发出的信号在信道上会出现叠加和碰撞的情况，这就导致接收方无法从接受到的信号中提取出正确的信息，因此对信道共享技术**最基本的要求**---*就是在某一时刻只有唯一有效的信息在信道上传递*。另外作为一个公共的信道，信道共享技术还必须保证平等的对待信道上面的每一个用户，也就是要让他们有平等的发送机会、接收时会，对它们的请求做出实时的反应
2. 信道共享技术特点与要求
    1. 信号的碰撞和叠加
    2. 要求某一时刻只有唯一信息有效传递
    3. 要求平等对待用户
        - 平等发送
        - 平等接收
        - 实时反应
    4. 用户好像独享网络
3. 信道共享技术也称**多点接入技术**，更具技术的控制复杂度和可维护性可分为**受控接入**和**随机接入**

4. 受控接入
    1. 特点
        1. 各个用户不能任意的接入信道而必须服从一定的控制（控制分为两种）
            - 集中式控制：轮询
                1. 网络中的接入点被分为主机和站两种类型。
                    1. 主机：主要负责接入的管理。按一定顺序询问各站有没有信息要发送。如果有，被询问的站立即将信息发送给主机
                    2. 站：主要参与数据传输
                2. 具体来讲轮询分为：
                    - 轮叫轮询
                        1. 工作原理：每个站只能接收主机的信息，也只能向主机发送信息
                        2. 机制：主机从1站开始，逐个询问各站是否有数据要发送。
                    - 

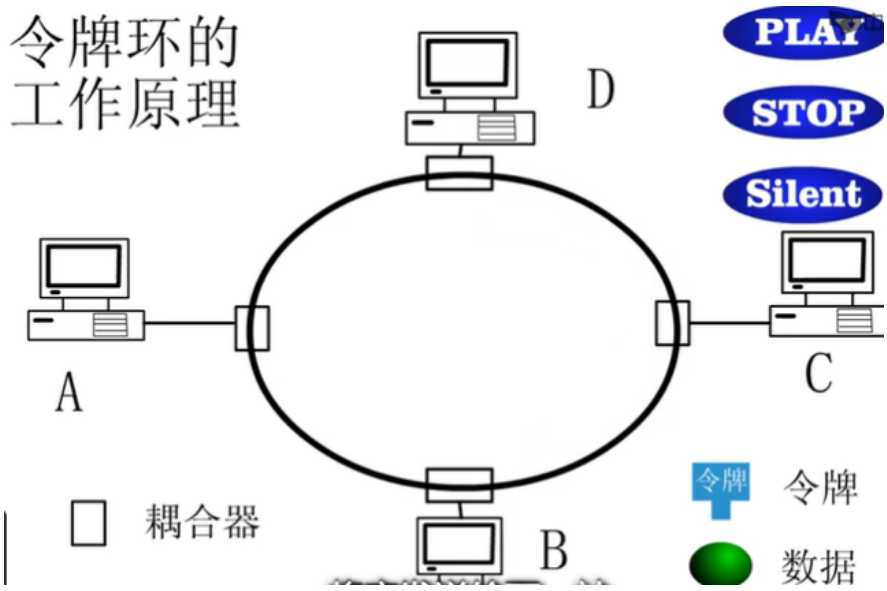
                    - 如果1站没有数据要发送，就发送一个空指帧给主机表示没有数据
                    - 然后主机在询问2站，2站如果有数据要发，就可以立即发送给主机
                    - 然后主机在询问下一站，知道询问完N站后，又重新从1开始
                    - 当然在这个过程中，主机也可以主动将数据发送给各站

3. 轮叫轮询最大的缺点是：**轮询帧在多点线路上不停的循环往返形成了很大的开销，增加了帧的等待时延。**为了克服上述缺点可以采用下述传递轮询的方法

- 传递轮询

1. 工作原理：每个循环从主机向N站发送轮询帧开始，当N站发送数据完毕，会以控制帧的形式告诉主机；没有数据发送时，会将N-1站的地址附加在控制帧中，这样控制帧就会被N-1站接收，也就是说，不是再由主机向N-1站发轮询帧，而是由N站向N-1站发送轮询帧……直到最后再由1站把主机地址附加控制帧中，询问全就又回到了主机手里，完成一个循环
2. 传递轮询的帧时延总是小于同样条件下的轮叫轮询，而且站间的距离越大，传递轮询的效果就越好。但由于传递轮询的协议比较复杂，实现成本高，所以在站间距离比较小且通信量较大的环境下，它的优势就不是那么明显

- 分散式控制：令牌环网



1. 工作原理：

1. 令牌环网的**拓扑结构**是一个闭合的环。所谓的令牌实际上是一种特殊的帧，它平时不停的在换上流动，当有一个站有数据要发送时，必须先截获这个令牌，然后再将数据发送出来。

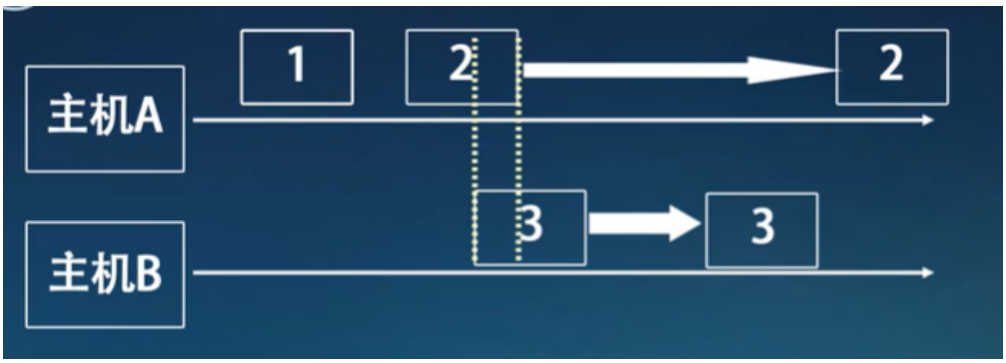
2. 当发出的数据在环路上经过目的站时，目的站一方面要复制这个帧，表示手下这个数据；另一方面还要将这个数据帧转发给下一站，让它在换路上继续流动。
3. 转了一圈之后这个帧一定会回到发送站，发送站对返回的数据帧进行检查，以判断数据是否发送成功
4. 数据帧接收完之后，发送站在生成一个新的令牌，将它发送给下一站。这样换路上就又有了令牌去等待其他站去截获它

2. 令牌环的工作原理体现了公平的接入原则，而且由于网络中不会出现几个站同时发送数据的情况，所以在重载环境下也不会因为冲突而影响效率。另外令牌环网还规定了每个站占用等待时间的上限，以保证一些实时性的应用。但是由于令牌环网的拓扑结构是闭合的环路，而且环路中使用了很多干线耦合器，一旦这些器件出现了故障，就会导致整个网络瘫痪。由于断点的检测比较困难，所以修复成本高（特点如下）
    - 公平原则，适合重载环境
    - 确定每个站占用信道的等待时间上限
    - 闭合环路一点断多点瘫，不易检查出断点
3. 针对上述令牌环网的缺点，利用总线型局域网的优势，令牌总线网就产生了
    - 

        1. 令牌总现网在物理上是一个总线型的网络，但在逻辑上却构成了一个环，使令牌可以像在令牌环网中那样在网络中流动。

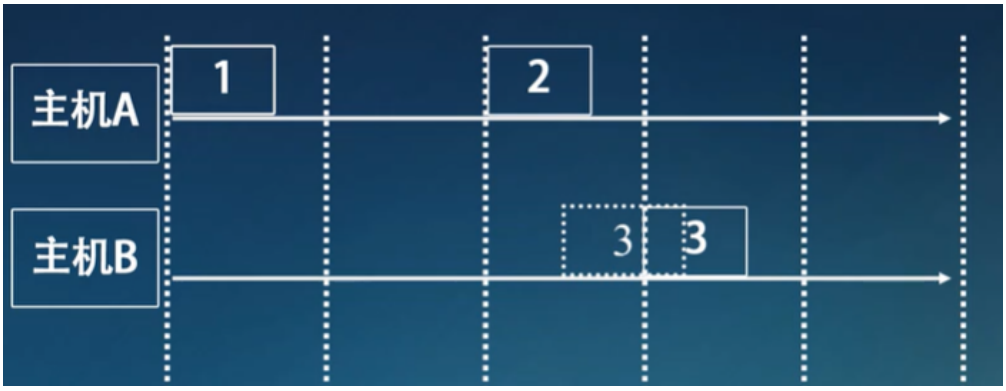
2. 与环形的网络相比，令牌总现网具有更高的可靠性。比如在上述的7个站中，如果C站出现了故障，而G站未工作，剩下的5个站在逻辑上依旧可以组成一个令牌环网。这里令牌传递的顺序与站的物理位置无关
3. 由此可见，令牌总线网既具有总现网的接入方便和可靠性高的有点，也具有令牌环网的无冲突和发送时延有确定上限值得有点。但是由于令牌在总线网中得传递顺序与站的物理位置无关，因此必须要有一个有效的**麦克层协议**来管理令牌，这就使得令牌总线网的协议非常复杂，所以推广应用比较差（特点如下）
    - 同时具有总现网和令牌环网的有点
    - 协议复杂、推广应用困难

二、信道共享技术之随机接入

1. 纯ALOHA
    1. 工作原理：想发就发。规定时间内若收到应答，表示发送成功；否则重发
    2. 重发策略
        1. 若立即重发，则显然要再次冲突
        2. 等待一段随机时间，然后重发；若再次冲突，则再等待下一段随机的时间，知道重发成功为止
    3. 工作过程
        - 

        1. 上图中主机A和主机B共享同一个信道。主机A先发送数据帧1，没有冲突
        2. 接下来A发送数据帧2，在数据帧2发送的同时，B发送数据帧3。显然数据帧2和数据帧3发生冲突，因此这两帧数据都要重传

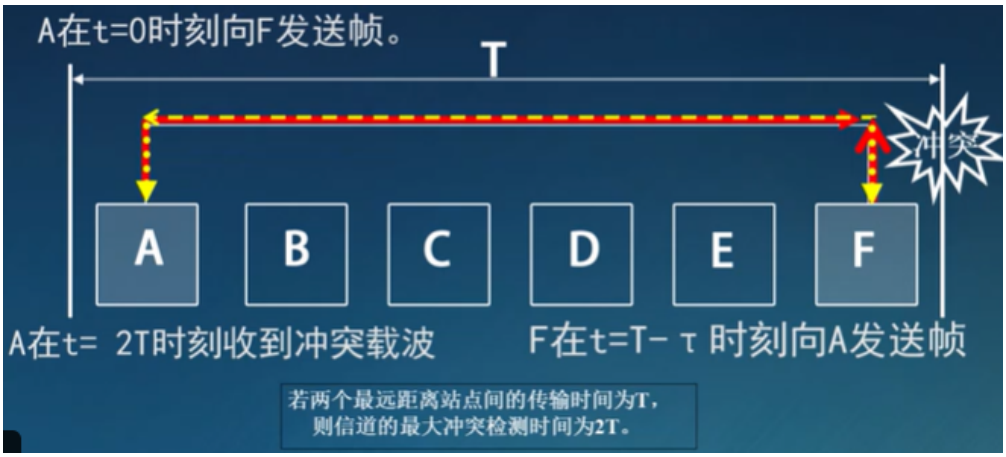
3. 主机A和主机B都随机等待一段时间后重传数据
4. 显然，在ALOHA机制下，信道上接入的站点越多，负载越重，发生冲突的概率就越大，因此性能也就越差

2. 时隙ALOHA
    1. 工作原理：为了提高ALOHA的性能，将各站的时间都同步起来，并且将时间划分为一段段等长的时隙，一个时隙长度正好发完一帧。同时规定，无论帧何时产生，都只能在每个时隙开始的时候发送到信道上，这就是**时隙ALOHA**
    2. 重发策略：同纯ALOHA
    3. 工作过程
        - 

        - 以上面情况为例，主机A在发送数据帧2时，主机B产生了数据帧3，但按照时隙ALOHA的要求，数据帧3不能立即发送，而是要等到下一个时隙开始时才能发送。这样就避免了冲突
        - 由此可见，与纯ALOHA相比，时隙ALOHA可以提高信道的吞吐量，改善性能
3. 从ALOHA演变而来的CSMA（载波侦听多点接入---Carrier Sense Multiple Access）：
    1. 工作原理：与ALOHA相比最大的差比就是增加了一个载波监听装置，每个站在发送数据之前都先监听一下信道上的其他站是否正在发送数据，如果有数据正在发送，那么该站就暂时不发送数据，这样就可以降低冲突发生的概率，把这个过程描述为“先听后发”
    2. CSMA的分类（根据监听的方式可分为）
        1. 非坚持CSMA：一旦监听到信道忙，就不再监听；延迟一个随机事件后再次监听
            1. 一个缺点：随机一段时间后，再次监听前，信道就已经空闲了，也就是说无法在信道空闲的第一时间把数据发送出去，

影响了信道利用率的提高，针对这个问题可以采用下面的坚持CSMA来解决

2. 坚持CSMA：监听到信道忙时，仍继续监听，直到信道空闲
    1. 两种不同的发送策略
        1. 1-坚持CSMA：一旦听到信道空闲就立刻发送数据（以概率1发送）
            - 问题：如果两个或更多的站同时采用这种方式监听信道，一旦信道空闲就会有多个站同时发送数据，这样就会发生冲突。
        2. p-坚持CSMA：因此就有了另外一种这种的策略，就是当听到信道空闲时，以概率p来发送数据，而以概率1-p来延迟一段时间重新监听信道
    3. CSMA的缺点
        1. 传播时延
        2. 仍然存在冲突的可能。比如：两个距离很远的站，其中一个站发送的数据在传输到另外一个站的附近之前是不会被监听到的。如果此时另外一个站也发送了数据，那么一段时间之后它们会碰撞在一起，如果两站没有发现这种碰撞而继续发送数据，那么后续的发送就是在拜拜浪费时间和资源
        3. 在冲突发生时，站不知道是否出现冲突，这样，发送数据的站将一直把数据发出，但显然这些数据是有错的，因此这段时间是浪费的
4. CSMA/CD（Carrier Sense Multiple Access/Collision Detection）
    1. 工作原理：边发送边监听（冲突检测）。若监听到冲突，则冲突双方都立即停止发送。信道很快空闲，从而提高效率。“先听后发，边发边听”
    2. 冲突检测的方法
        1. 比较接受到的信号电压的大小：一旦信号经过叠加，其电压的摆动值就要比正常值大一倍
        2. 检测曼彻斯特编码的过零点：采用曼彻斯特编码时，电压的过零点是在每个比特的正中央。如果发生冲突，过零点的位置会发生偏移，所以根据过零点的位置也可以判断冲突
        3. 发送的同时也接受，将收到的信号逐个比特的与发送的信号进行比较，如果有不同就是发生了冲突。

3. 检测到冲突后，停止发送数据，并且发送人为干扰信号，强化冲突。**目的**：让信道上的每个站都能知道冲突的发生，这样信道就能很快地空闲下来
4. 从前面的介绍中可以发现，由于电磁波的传播时延，在每个站刚刚发送数据开始的一段时间间隔内，仍然有发生冲突的可能，因此把这段时间称为冲突检测时间
    - 

    - 总线一端的A站在$t=0$时刻向F发送数据帧
    - 假设数据帧在A和F两站之间单程的传播时间为T.
    - F站在A站发出数据帧的某一时刻也想发送数据，于是就监听信道
    - 由于A站发送的数据还没有到达F站附近，所以F站发现信道是空闲的，于是在$t=T-\tau$时刻向A发送数据帧（τ是非常小的一个值），可见这两个数据帧一定会发生冲突，而且冲突会发生在F站附近
    - 冲突首先会被F站发现，之后F站立即停止数据的发送，并发送人为干扰信号强化冲突
    - 大约在$t=2T$的时刻，冲突信号会传到A站
    - **通过上述分析可知，如果两个最远站点间传输时间为T，那么信道的最大冲突检测时间为$2T$。**
5. 碰撞槽时间（Slot time）---最大冲突检测时间：
    1. 指的是从发出数据开始，到检测到信号冲突的时间间隔的上限
    2. 假设最远主机之间的传输媒体长度为S，帧在媒体上的传输速度为$0.7C$（C为光速），物理时延为t_P，那么两个站点间单程的传输时间为$T=\frac{S}{0.7C}+t_p$，$Slottime=2T=\frac{2S}{0.7C}+2t_p$.
    3. 显然碰撞槽时间越是小于一个帧的发送时间，CSMA/CD的优越性就越明显。相反，如果一个帧的发送是时间小于碰撞槽时间的话，那么即便冲突发生也无法检测出。因此在采用CSMA/CD的信道上，对链路上的最小帧长进行了限制

4. 假设L_{min}是最小帧长，R是数据传输率，则必须满足$\frac{L_{min}}{R} \geq Slot\ time$；由上知$L_{min} = (\frac{2S}{0.7C} + 2t_p) \times R.$

五、局域网技术

> tips：这一块的内容，我看的这个老师讲解的稀巴烂，感觉他说话罗里吧嗦的，还没有结构，然后下面的笔记就贼乱！！！不过我过段时间会看书直接给它补充完整。
>
> 真tm恶心，遇到讲课的老师，真tm吐了，连个英文单词都给你读一遍，三岁小孩？？？
>
> 有这个时间把课内容讲好不好吗？？？

一、局域网体系结构

1. **IEEE802参考模型**
    1. 有别于传统的OSI模型以及5层体系结构
    2. IEEE802参考模型对应于OSI参考模型的物理层和数据链路层。数据链路层又分为逻辑链路控制子层和介质访问控制子层
    3. 将数据链路层分为两个子层
        1. MAC：Media Access Control
        2. LLC：Logical Link Control
        - 原因：媒体介入方法太多，分成两层后便于简化系统的实现
2. 局域网802参考模型和OSI/RM的对比
    1. 在5层体系结构中，把数据链路层分为两个子层
    - 

    1. 两个子层

    1. 偏上的**逻辑链路控制LLC子层**：主要向上层提供连接的环境
    2. 偏下的是**媒体访问控制MAC子层**：对下层提供媒体访问的具体方法
    3. 两者通过统一的物理层来传输数据
2. 最下面的是物理层：作为传输介质
3. 在数据链路层中需要提供SAP（service access point---服务访问点），服务访问点应该出于偏上层的LLC子层上
4. 各层的功能
    1. 物理层：负责信号的编码译码、同步码的产生和去除、比特的传输和接收
        - 物理层的功能和传统的五层体系结构是一致的
    2. MAC层：成帧和拆帧、实现和维护MAC协议、比特差错检验、寻址
        - 注意，在差错检验中，一旦检验出差错进行后处理时，差错控制机制是交给LLC子层来处理

          的
    3. LLC层：建立和释放逻辑链路、与高层的接口、差错的控制、给帧加（减）序号
        - 由于MAC子层负责媒体传输访问的细节，所以局域网的传输介质访问细节对于LLC子层是透明的
5. 这样分成两个不同的子层，在IEEE802参考模型中存在两个不同的PDU
    - 

    1. LLC-PDU：高层数据首先被交给LLC子层，加上LLC首部幸成LLC的PDU
    2. 向下会交给MAC子层，将LLC-PDU当成MAC子层的数据部分，然后分别加上MAC的首部和尾部，封装成帧，这是就会出现两个不同的PDU
6. IEEE802参考模型的架构

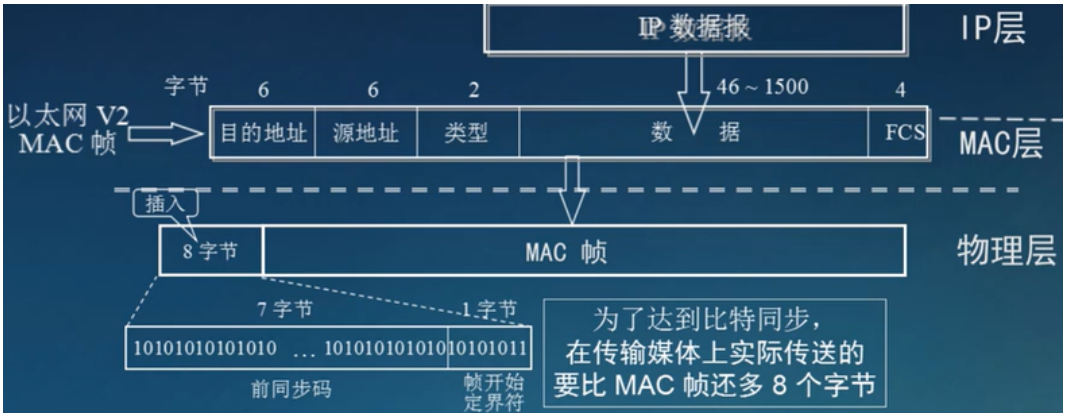
- 

  - 将802参考模型分成若干个子标准
  - 802.1是一个总述，用来描述OSI模型与网络管理在IEEE802参考模型中发挥的作用
  - 802.2用来描述LLC子层。在IEEE802参考模型中只有一个802.2文档是用来描述LLC子层的
    - 其他802.3之后的是描述MAC子层的
  - 802.3是用来描述CSMA/CD---以太网
  - 802.4用来描述令牌总线网
  - 802.5用来描述令牌环网
  - 还有802.6用来描述城域网，802.11用来描述无线网--WIFI的

7. 在802体系结构定义下局域网的地址结构
   - 

1. 局域网中，地址表示采用二进制来进行描述。在IEEE802的体系结构中，局域网的地址使用的是MAC地址，使用的是48比特的二进制数来进行定义的，实际记录采用的是十六进制
2. MAC地址是网卡地址或者说只要是网络接口上的都有MAC地址的信息。在实际的记录和表示时，6组十六进制数也采用点进行分割
3. MAC地址作为网络设备接口的标识，它在记录地址信息的时候需要保证全球的唯一性
    - MAC地址分为两部分。前24比特叫制造商ID，有IEEE统一分配；后24比特由公司企业内部自己确定它的唯一性

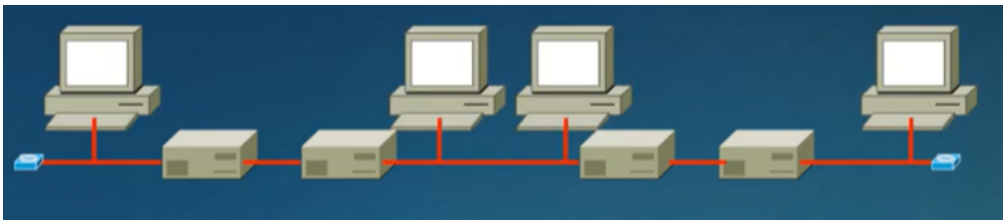
二、以太网

1. 以太网V2的MAC帧格式
    1. 802.3采用CSMA/CD（先听后发，边听边发的策略），称为以太网Ethernet，重载下性能差。分类如下：
        - 10 BASE 5、10 BASE 2、10 BASE T、10 BASE F
        - BASE：表示采用基带的信号进行传递。BASE前面的数值描述当前技术的网络带宽的大小，BASE后面的字符表示所使用的细节。如10 BASE 5，10MBPS的基带粗缆以太网
    - 

    2. 上图就是以太网的MAC帧格式。第一个字段6字节是目的地址，第二个字段6字节是源地址，第三字段2字节用来标识这个帧所携带的数据部分它封装的高层协议，最后一个字段4字节是它的校验字段，合到一起就成为了一个以太网的帧格式。然后被物理层发送出去
    3. 以太网的帧被物理层发送出去前，需要在前面增加8字节的比特同步。在实际传输要比MAC帧多8个字节，起到同步和前挡的作用

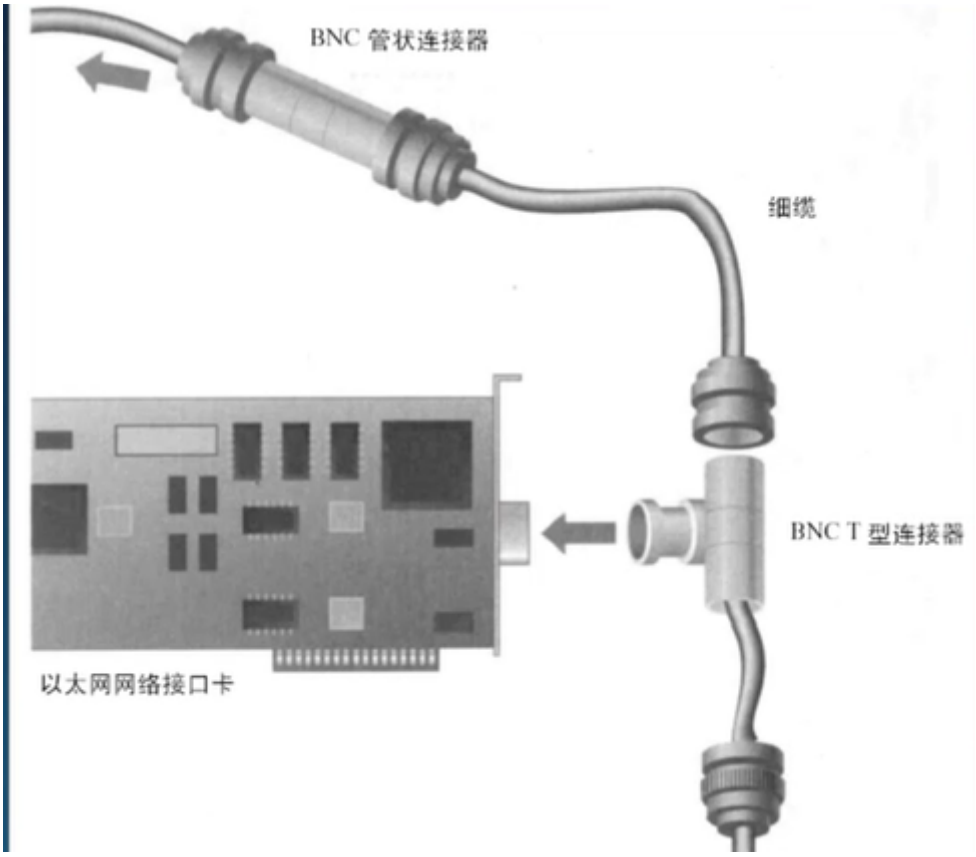
        - 根据以太网的帧格式就能很好的分析以太网所携带的数据的含义

2. 标准以太网（10 BASE 5）
    - 传输媒体10mm粗缆（铜缆）
    - 特性抗阻50Ω.
    - 曼彻斯特编码
    - 总线拓扑
    - 10Mbps
    - 单段最长500m
        - 方波在传输时随着距离的增加会有信号的衰减和变形，当达到500m时信号就可能衰减的不可识别

    1. Repeater（中继器）：在物理层上实现局域网互联的设备，负责连接各个电缆段，对信号进行放大和整形，驱动长线电缆连接。
        - 不能无线增加长度，**“543”规则**
        - 
        - 使用中继器续连时，500m最多5段；4个中继器；最多三段500米可以连接计算机

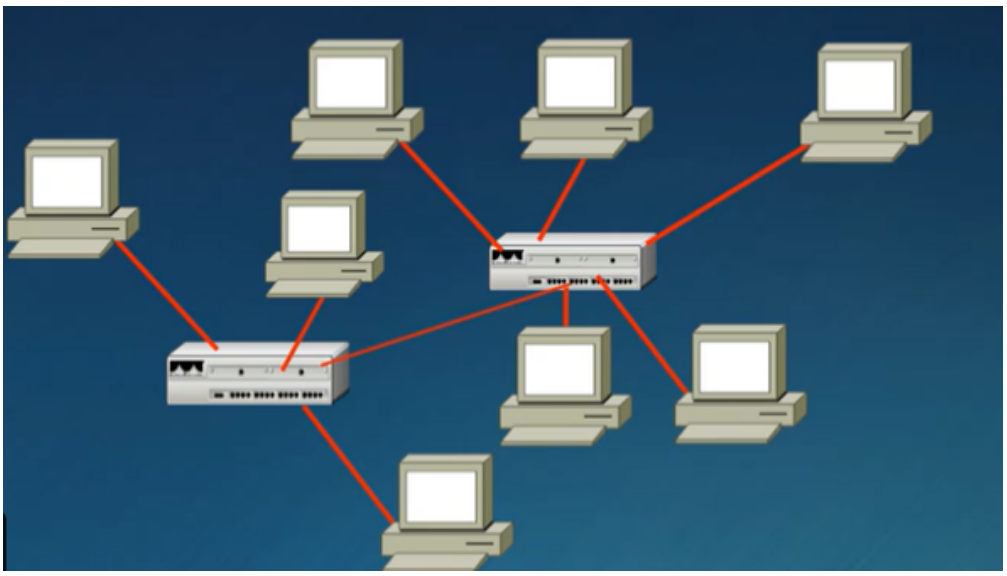
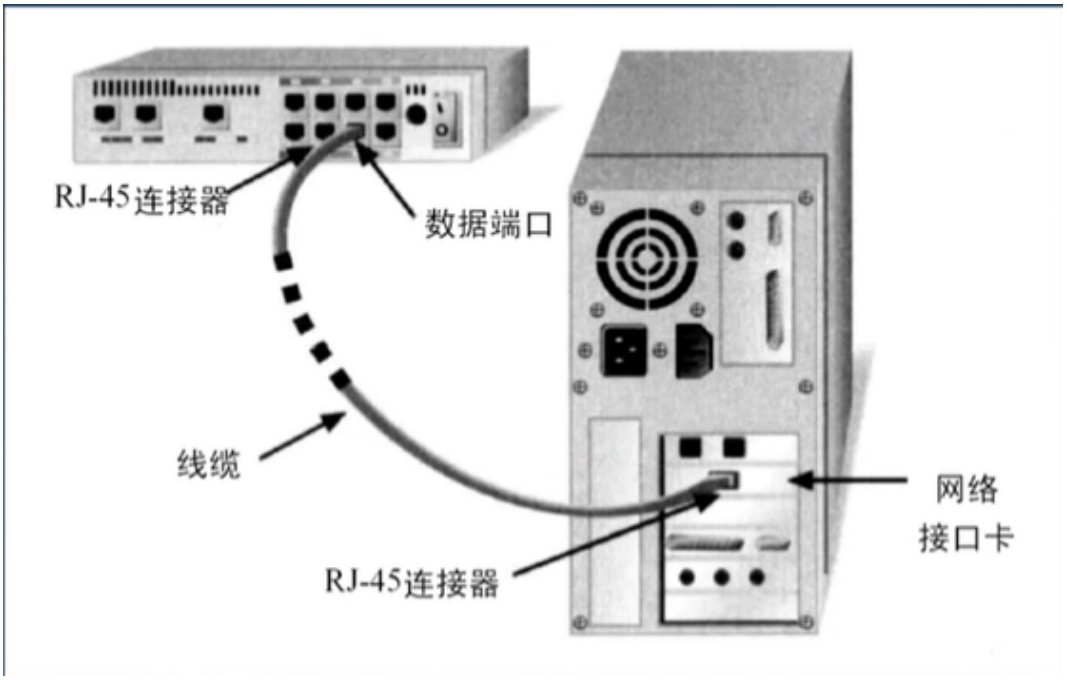
3. 细缆以太网10 BASE 2
    - 传输媒体：细缆
    - 特征抗阻：50Ω.
    - 曼彻斯特编码
    - 总线拓扑
    - 单端最长185m
    - 185*5=925m，网络跨距最大长度

    1. 细缆和网卡的连接方式

- 

    - 通过BNC的T型头直接与网卡连接，省去了连接的电缆

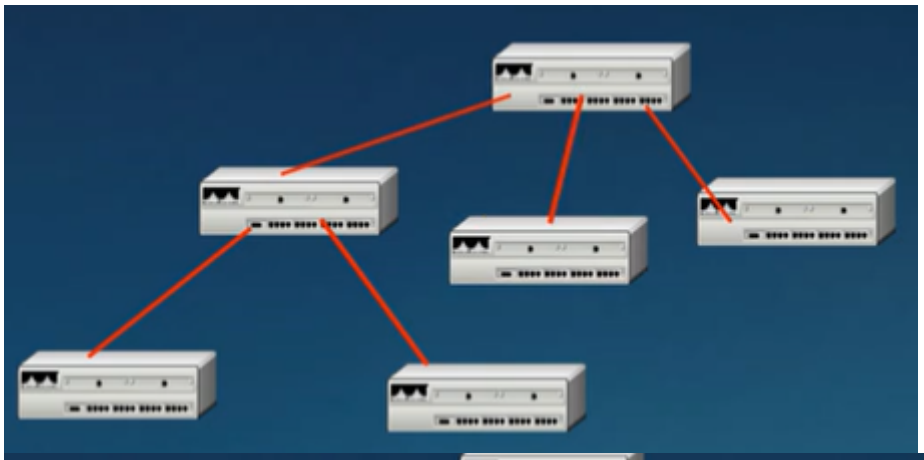
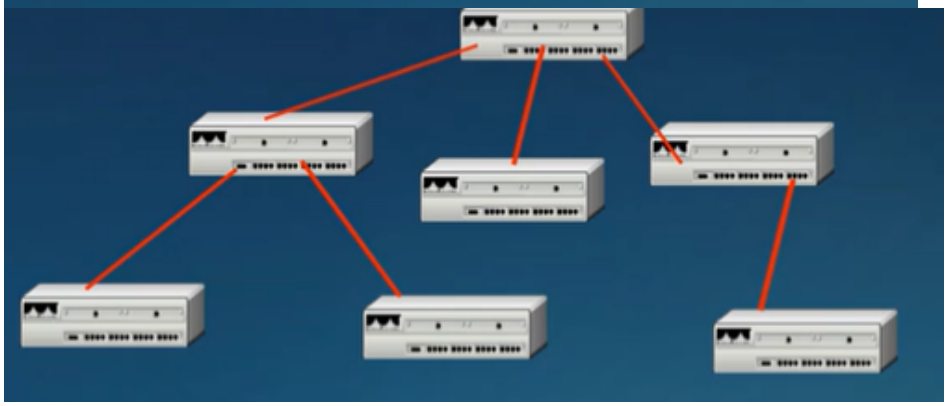
4. 10 BASE T（T（twisted pair）---双绞线）
    - 传输媒体：UTP
    - RJ-45连接器
    - 
    - 星形拓扑结构

- 
    - 扩展网络距离，两个集线器连接一条级连线
- 单段最长100m
- 曼彻斯特编码
- 使用HUB互连
- 
- 网络跨距500m
- 

1. 在使用双绞线的网络中，如果要扩展网络距离，就要使用级连的方式

2. 级连的规则（10 BASE T 要求）

    1. 距离最远的两排主机之间最多只能间隔4个集线器

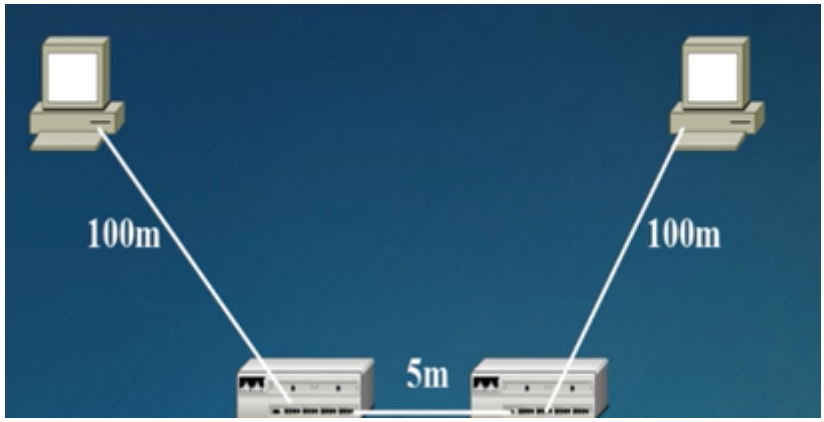
        - 
        - 
        - 很显然第一张图可以，第二张图不行

三、快速以太网

1. 采用的是100Mbps的带宽。100 BASE TX

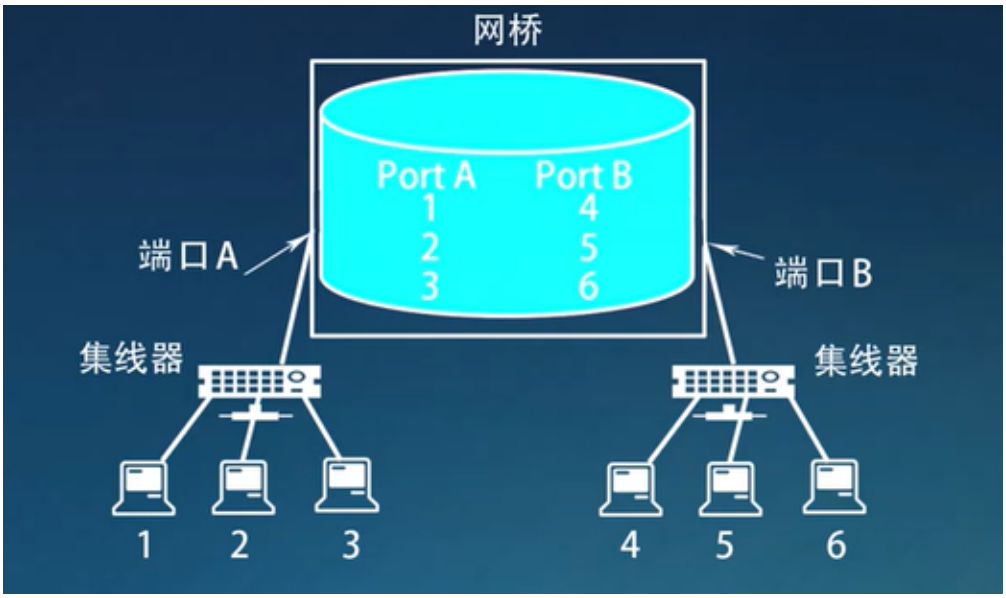
    - 传输介质：5类UTP
    - 传输速率：100Mbps
    - CSMA/CD
    - 单段线路最长100m
    - 使用HUB连接

    1. 网络跨距问题

        - 

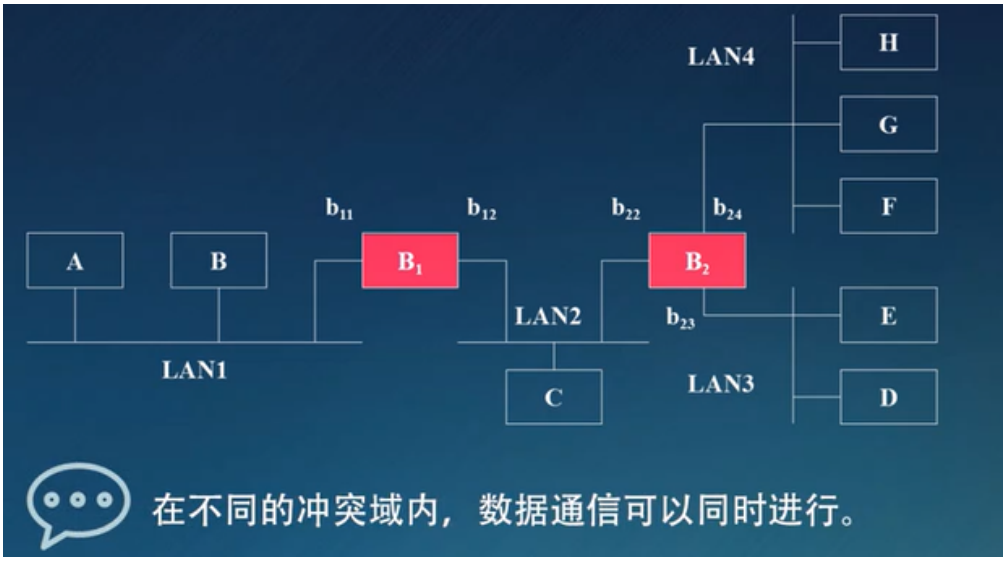
        - 系统中最多两个集线器，且距离不大于5m，两台主机之间最大距离205m
2. 除了是用双绞线作为传输介质，也可以使用光纤作为传输介质。100 BASE FX
    - 传输介质：光纤
    - 传输速率：100Mbps
    - CSMA/CD
    - 使用**一对**光纤进行连接传输
        - 因为光纤只能单向传输数据，而网络通信需要双向传输数据

四、局域网的扩展

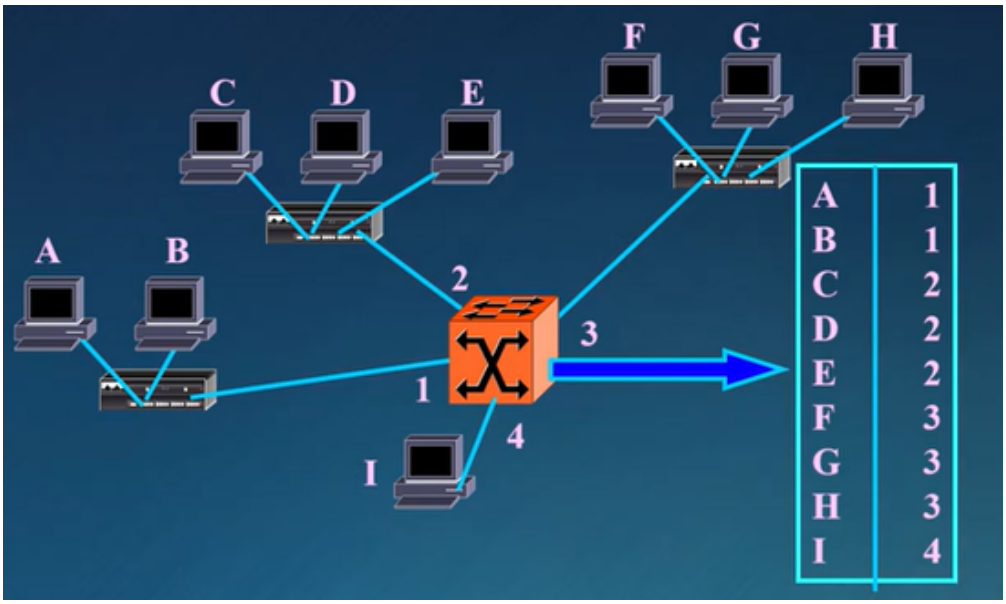
1. 网桥
    1. 网桥基本原理
        - 

        - 网桥可以连接多段不同的网络。上图中的网络有2个口（端口A和端口B）。端口A连接了一个集线器所连接的网络，端口B连接了另一个集线器所连接的网络将这个网络分成了两个部分。一个是1，2，3三台主机，另外一个是4，5，6三台主机
        - 网桥内部和中继器不同的是，网桥有一个表，这个表就是MAC地址表。网桥内部的MAC地址表记录的是它的端口和主机的对应关系。比如上图中，记录下来端口A对应主机1，2，3，端口B对应主机4，5，6.
        - 网桥可以依据上述表记录的消息来决策是否转发数据

- 比如透明网桥
    1. 帧的转发策略（先查看帧的地址DA/SA（目的地址/源地址））
        1. 相同网络数据交换，丢弃帧，不转发
        2. 不同网络数据交换，接收处理帧，转发
    2. 帧转发规则（查看MAC地址表）
        1. DA在表内，按表的指定端口转发帧
        2. DA不在表内，用扩散Flooding方式转发帧（向所有其他端口转发帧）

2. 在上图网桥进行工作的时候
    1. 如果1向2发送数据时。这个信号将通过端口A到达网桥时，网桥通过查表，发现原MAC地址和目的MAC地址是相同网络的，这时候网桥将不会把这个数据通过B口转发出去。
    2. 而如果是当1向4发送数据时，网桥通过查找MAC地址表，发现这是不同网络之间的数据传输，才会转发这个数据。
    3. 如果1向2发送数据的同时，4也向5发送数据时，由于网桥的隔离效果，使得1，2发送数据的同时，4，5也能同时发送数据，**这样就使得网桥有效的隔离了传输的数据流量。**
3. 网桥在工作时能有效的对各个不同的网络区域进行分隔
    - 

        1. 上图中，B1和B2是两个网桥，它们不同口上所连接的局域网都可以在独立的进行数据传输，数据通信可以同时进行，大大的提高了数据传输效率。
        2. 而如果中间不是网桥而是中继器，显然只能同时有两台主机之间进行通信，这样网桥就可以有效的隔离冲突域
4. 网桥的优点

    1. 过滤通信量
    2. 扩大物理传输范围
    3. 可以连接不同的物理层
    4. 提高传输的可靠性
5. 网桥的缺点
    1. 增加了传输延迟
    2. 没有流量控制功能，可能产生溢出
    3. 帧的处理耗费时间
    4. 只适合比较小的局域网，大规模的局域网中可能产生广播风暴
2. 最常用的**“局域网交换机”**
    1. 局域网交换机与网桥的关系类似HUB和中继器的关系，HUB/集线器是多端口的中继器，而局域网交换机可以说是多端口的网桥。
    2. 局域网交换机的基本工作原理和网桥是类似的，只不过在工作过程中采用了交换技术。一般来说是目前局域网的常用阻抗设备
    3. 局域网交换机的原理
        - 

        - 如上图，与网桥类似，局域网交换机的内部也有一个MAC地址表来记录每个端口与对应主机之间的关系。与网桥也类似的是当进行决策转发时，交换机也可以去根据源地址和目的地址是否在同一网络内来决策是否转发这个数据。与此同时交换机还采用了交换技术
    4. 局域网交换机的主要特点
        1. 所有端口平时都不连通
        2. 当接入交换机的设备之间需要通信时，交换机能够同时连通许多对端口

3. 每一对相互通信的设备都能像独占通信介质那样进行无冲突的数据传输
4. 双方完成通信后，会断开这种连接
- 这样就使得网络中不同端口对之间可以同时进行数据传输，这就是所谓的**交换**。

5. 局域网交换机在进行帧的转发过程中，需要首先去决定到底如何缓存这个帧。因为只有当局域网交换机先缓存好帧之后，才有可能根据帧里面的字段来决策是否转发这个数据。具体缓存策略分为3中

    1. Store & Forward（存储转发式）

        - 最早期的局域网缓存设备是需要将整个的帧完全缓存后，去识别目的MAC地址的信息与原MAC地址的信息进行比对，然后决策是否进行转发。
        - 当然，如果将全部的帧尽心缓存后，可以通过CRC校验的方式来确保所转发帧的正确性，不会转发废弃的数据和错误的数据。可以提高数据转发的效率
        - 反过来说，又会降低数据转发的效率。因为缓存本身所花费的延迟时间比较长

    2. Cut Through（直通式）

        - 考虑存储转发式需要缓存全部的数据帧，代价太大。因此考虑缓存比较少的数据量，比方说根据前面的以太网帧的结构，实际上交换机只需要去识别所接到帧的目的MAC地址信息就可以了
        - 

        - 而目的MAC地址正好出现在以太网帧的前6个字节，因此没有必要缓存全部的数据帧，而只需要缓存前6个字节就可以获取所需要的目的MAC地址。
        - 由于上面缓存的数据量少，缓存所花费的时间少
        - 但也存在问题，无法进行CRC校验，可能会转发报文碎片

        - [x] 报文碎片：局域网中小于64字节的数据帧一定是报文碎片。前面有对以太网数据帧的最小长度限定为64字节，所以当某个数据帧字节数小于64那么就一定是一个错误的数据帧

    3. Fragment Free

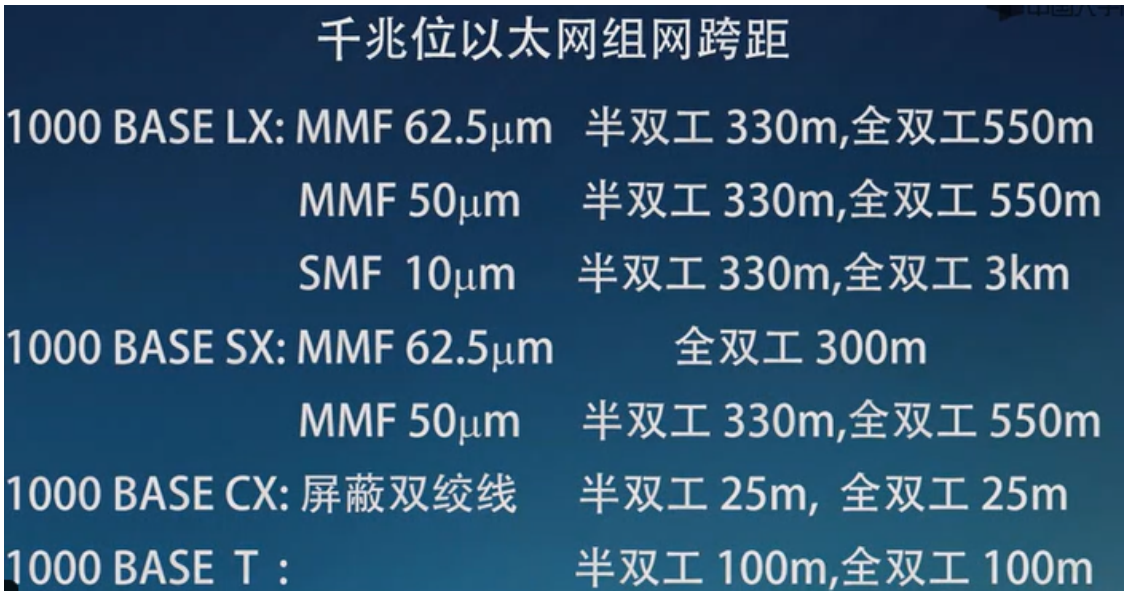
        - 为了确保以太网交换机不转发报文碎片，所以出现了这种转发策略
        - 仅缓存前面64个字节的数据
        - 缓存前64字节数据时，一方面可以获取目的MAC地址的信息，另一方面也可以尽量减少所缓存的总的数据量，因此性能在一定程度上是提高的

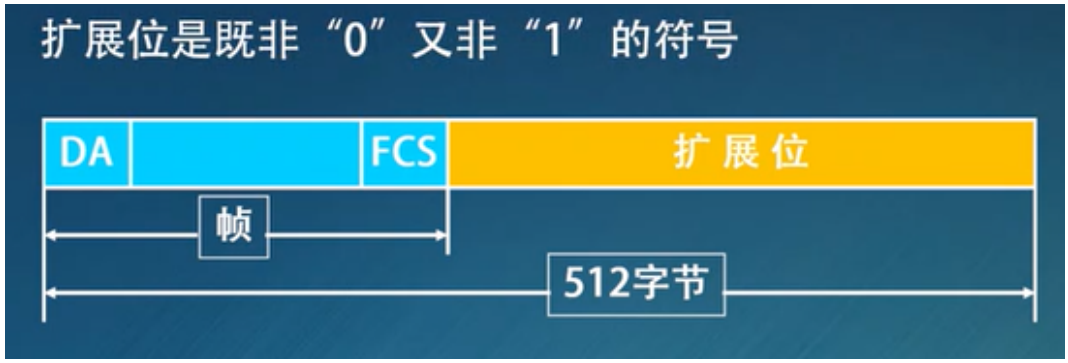
五、千兆位以太网

1. 最早采用的技术是1000 BASE CX
    - 使用特殊标准的短距离屏蔽铜缆来进行连接，最长25m
    - 使用9芯的D型连接器
    - 

    - 但是与百兆和10兆的技术不兼容，只是一个临时的过度技术
2. 采用光纤连接的千兆位以太网包括1000 BASE LX和1000 BASE SX
3. 1000 BASE LX（L->long，长波激光光源）
    - 传输媒体为光纤
    - 长波激光光源（LWL）
    - 波长1270nm---1355nm
    - 可以驱动的单模或者多模的光纤
    - 使用SC型光纤连接器，与100 BASE FX相同（可以支持由百兆到千兆的平滑升级）
4. 1000 BASE SX（S->Short，短波激光光源）
    - 传输媒体为光纤
    - 短波激光光源（SWL）
    - 波长770nm---860nm
    - 只支持多模的光纤
    - 使用SC型光纤连接器，与100 BASE FX相同（可以支持由百兆到千兆的平滑升级）
5. 1000 BASE T（T，双绞线）
    - 传输媒体为双绞线

- IEEE802.3ab
- RJ-45连接器，与100 BASE T相同（可以支持由百兆到千兆的平滑升级）
- 跨距100m，兼容其他双绞线连接方式

6. 

7. 帧扩展
   - 千兆位以太网的挑战---它的最小帧长度需要做一个扩充
   - 把最小帧长度一直扩展为512字节
   - 对于小于512字节的小帧，添加扩展位
   - 

8. 帧突发
   - 作为千兆以太网，仅仅做了帧扩展后是不够的，因为帧扩展使用的扩展位是无效数据，会降低网络传输的效率。这样会造成总体的有效数据量比例下降，因此造成总的带宽浪费。因此帧扩展需要另外一个技术来进行支持，以提高数据传输的效率，这就是所谓的帧突发
     - 帧扩展在大量的短帧环境中容易造成宽带浪费
   - 如果一个站点需要连续发送短帧，则第一个帧需要增加扩展位到512字节后再发出去。一旦第一个帧成功发送，那么后续短帧就可以连续发送，直到1500字节

- 

  - 允许站点连续发送多个短帧，直到1500字节
  - 在后续短帧进行发送的时候，短帧与短帧之间要通过帧间隙来进行分隔，帧间隙是既非0又非1的符号，这就是帧突发技术

六、网络层与网络互连

一、互联网

1. 因特网（Internet）：全球最大的、开放的，由众多网络相互连接而成的特定计算机网络，采用TCP/IP协议簇，前身位ARPANET
2. 互联网（Internet）：泛指多个计算机网络互连形成的计算机网络
3. 

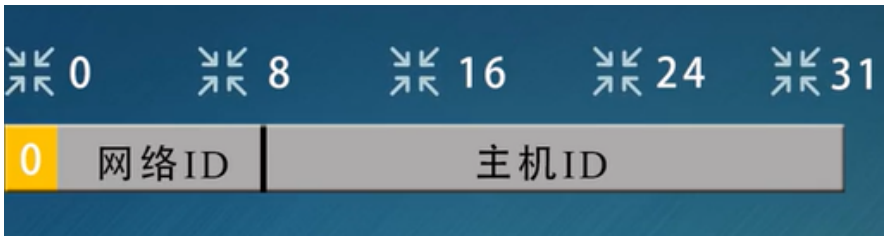
二、IP地址和子网掩码

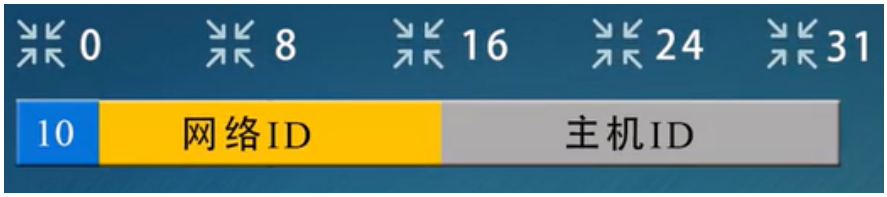
1. IP地址（IP协议的地址），IP协议----Internet Protocol
2. 在网络层当中，可以提供两种不同的服务模式
   - 可靠的面向连接的服务

    - 不可靠的无连接服务
3. IP协议是Internet中不可靠的无连接服务
4. IP协议的特点
    - 采用“尽力传递”的设计思想：因为IP协议无法提供可靠的数据传输，主要原因在于IP协议采用无连接的分组传输机制
        - 面向连接：指的是在通信之前需要建立连接，在通信当中需要维护连接，在通信后需要释放连接，如下图的打电话
        - 

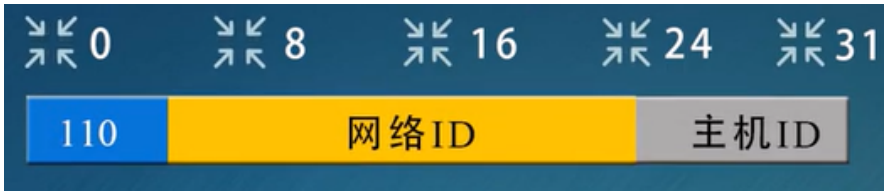
        - 无连接就是无需去建立维护和断开连接，最典型的例子就是发短信
        - 无连接方式的通信双方无法知道对方的状态，所以不可靠（没有差错恢复，同时可能有数据丢失）
        - 纠错重传问题交给传输层解决
    - 快速、简单、效率高（牺牲数据传输的可靠性来获取数据传输的快速简单和高效性，IP协议的高层具有其他协议来帮助进行纠错和重传）
    - 实现IP的路由
5. IP层的主要协议
    - 应用层、运输层、网络层、链路层、物理层
    1. TCP/IP协议簇的网络层，可以称为IP层。在IP层中，最核心的协议是IP协议。还有一些辅助的其他协议，比如偏下层的ARP和RARP协议，偏上层的ICMP和IGMP协议
6. IP地址

1. IP地址的组成：
    - IP地址典型的是由：网络（ID）地址和网络上计算机地址（ID）两部分组成
    - IP地址是使用4个8位（32位）二进制字节表示。实际记录用4个点分的十进制数表示如`202.118.100.196`，其中每个十进制数都会对应一个8位的2进制数，所以范围是`0~255`。
    - 32比特的二进制数被人为的分为两部分
        - 

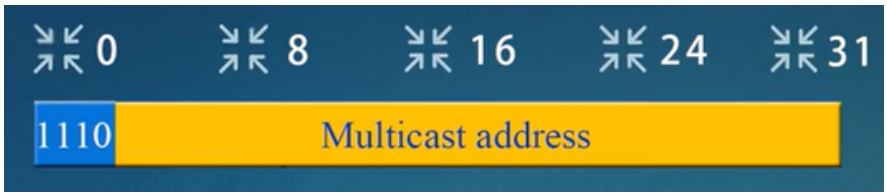
        1. 网络ID
        2. 主机ID
        - 原因是：这和IP地址的寻址方式相关。IP地址寻址时首先会根据网络号寻找网络，找到网络后，再在网络当中根据主机号来寻找主机，因此需要把IP地址分为网络号和主机号
2. IP地址的分类
    1. 原因：网络号和主机号占用的比特数会影响网络数和网络中主机数的变化
    2. IP地址一般被分为A、B、C、D、E，共5类，且常用的有A，B，C类（可以分配主机号）
        1. A类地址
            - 

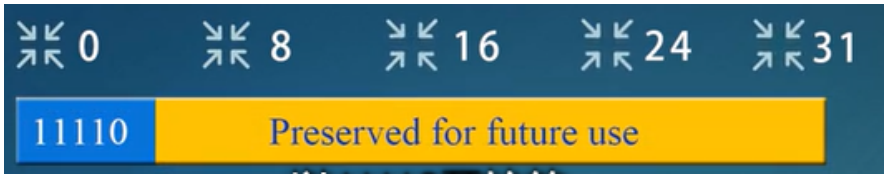
            - 前8个比特为网络号，后24个比特为主机号
            - 0作为第一个比特
        2. B类地址
            - 

            - 前16比特为网络号，后16比特为主机号
            - 前两个比特位10
        3. C类地址

- 

  - 前24比特为网络号，后8比特为主机号
  - 前三比特为110

4. D类地址
   - 

     - 1110开始的为D类地址，为多个地址
5. E类地址
   - 

     - 以11110开始的是E类地址，E类地址是保留做未来使用的

3. IP 地址的分类

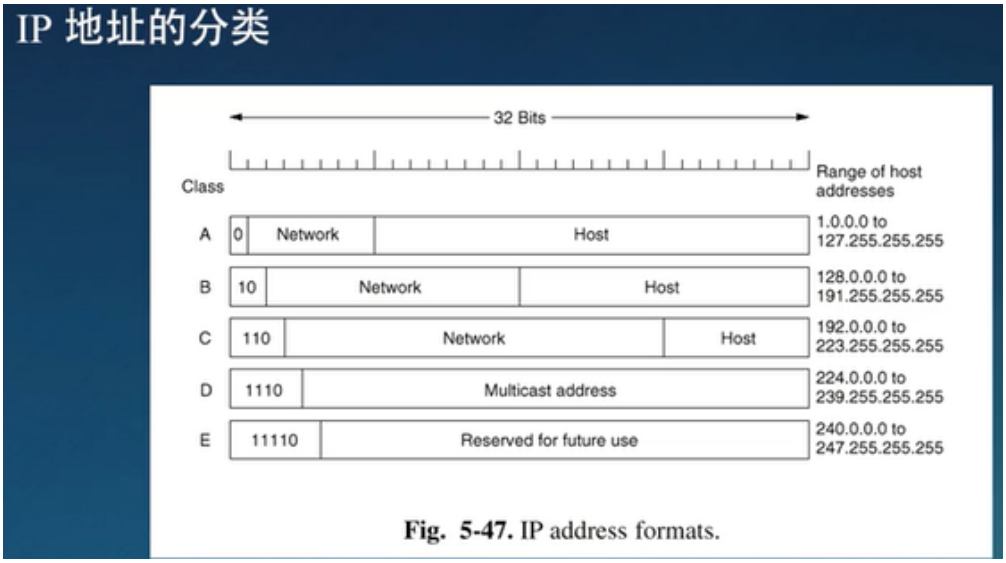


Fig. 5-47. IP address formats.

- 通过第一个十进制数就可以很清楚的分别属于哪类IP地址

4. 特殊的 IP 地址

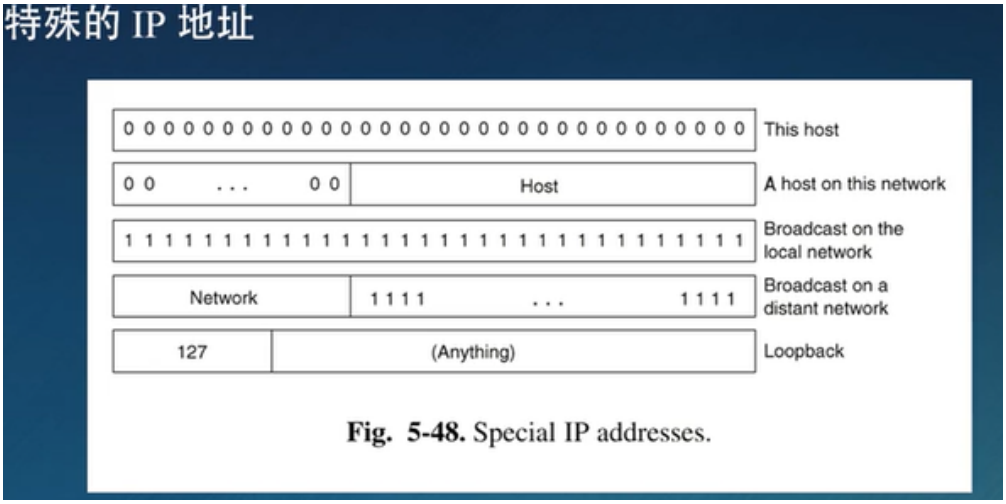
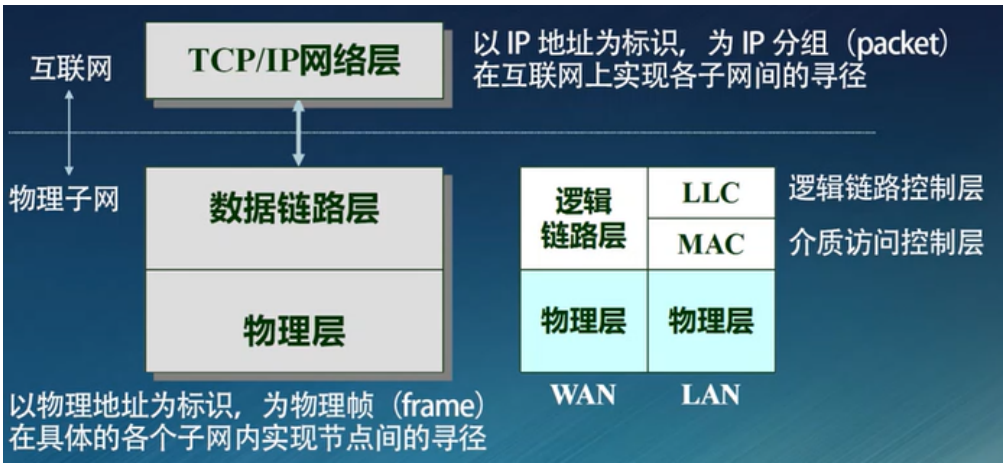
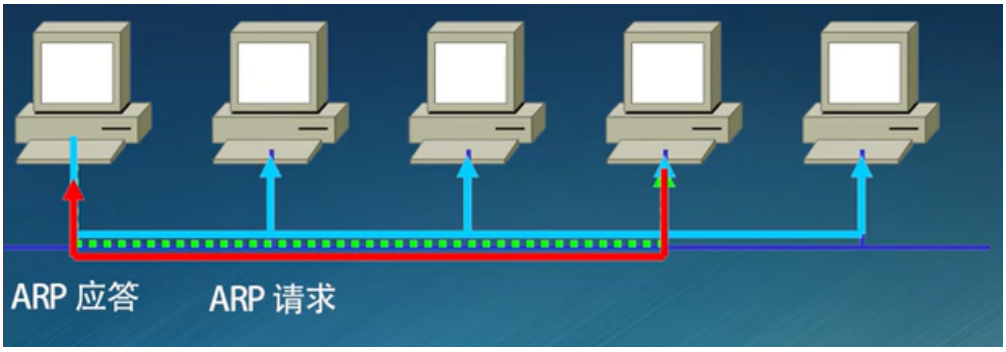


Fig. 5-48. Special IP addresses.

        - 全0表示自己，全1表示所有
        - Host：本机（不能参与通信，仅在主机未获得自身IP地址时临时使用的一个IP地址）
        - A host on this network：本网络的某个主机号
        - Broadcast on the local network：本网络内的广播（只能向本网络内的所有主机发送信息）
        - Broadcast on a distant network：某网络的广播地址
            - 如果`network|0000...0000`：表示某网络的网络号
        - Loopback address：简单的说表示本机（可以向这个发送数据，但是向这个地址发送数据时，这个数据包不会通过数据链路层和物理层的封装和转发，直接会在网络层作为一个Loopbacks的转发，直接发给接收方的模块）
    5. 总结
        - 分配给主机的IP地址，其主机号不能全为‘0’
        - 分配给主机的IP地址，其主机号不能全为‘1’
        - 主机号为全‘0’的，表示某网络的网络号
        - 主机号为全‘1’的，表示某网络的广播地址
        - 如果主机号为n位，则能容纳$2^n - 2$台主机
7. 子网掩码
    - 新问题，作为主机，如何判断一个IP地址它的网络号到底是多少？（计算机不擅长逻辑，擅长计算）
    - **子网掩码**是由连续的若干个二进制“1”组成的代码，而组成子网掩码的“1”的个数与网络号比特数相同
    - 确定网络ID和主机ID的方法：`子网掩码 and IP地址 = 网络地址（ID）`。
    - 主机使用子网掩码判断目的IP地址是否与主机处在同一个网络中
    - 主机在发送数据前使用目的IP地址与子网掩码进行与运算，再把本机的IP地址和子网掩码进行一次与运算，比较两次结果
        1. 结果相同，则是在本网络中进行数据传输
        2. 结果不同，不是在本网络中进行数据传输，则需要将发送的数据发送到网络的出口，发到其他网络中去

三、ARP协议

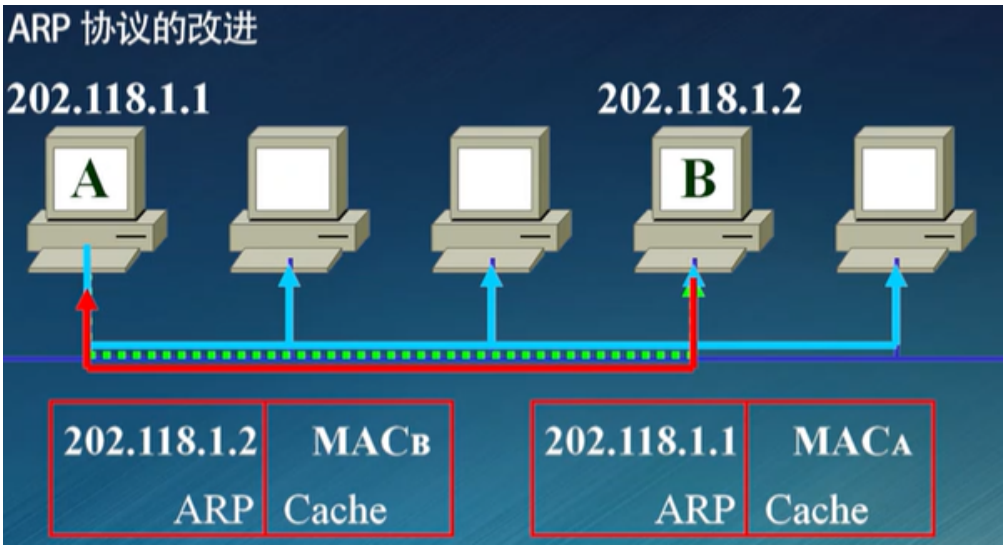
1. ARP协议（Address Resolution Protocol---地址解析协议）
    1. 功能：MAC地址和IP地址之间的地址转换功能
    2. 网络中两个层都有自己的地址，数据链路层有自己的MAC地址，网络层TCP/IP协议中有自己的IP地址。面临一个问题，这两个地址如何进行映射？
        - 

        - 换句话说，在传输过程中，很可能我已经知道对方的IP地址，却不知道MAC地址，这时候就需要做一个转换。为什么IP地址和MAC地址需要进行转换？形象点：IP地址相当于人名，MAC地址相当于人本身
    3. 基本原理
        - 

        - 比如上图中，绿色箭头两端的电脑需要从尾端向箭头端的主机发送数据，发送主机已经知道对方主机的IP地址，却不知道MAC地址
        - 由于局域网是一个易于传播的环境，因此，请求主机只需要向网络发送一个广播的ARP请求，并且等待对方的这个请求主机的返回一个ARP应答即可
        - 其中有个小问题：为何要广播发送请求，得到应答后，然后再单播的通信呢？直接广播发送数据就行了。
            1. 原因：人的大脑是有记忆的，也就是说只要问过一次张三是谁后，就可以记住张三的名字和这个人对应的关系，下此就

可以直接找张三这个人就行，不需要再次广播了。网络中和这是类似的，尤其是在局域网中一般来说要尽量避免广播通信的产生，要尽量减少广播通信量。因此作为ARP协议，基于这样的思想，做了些改进。

4. ARP协议的改进

- 

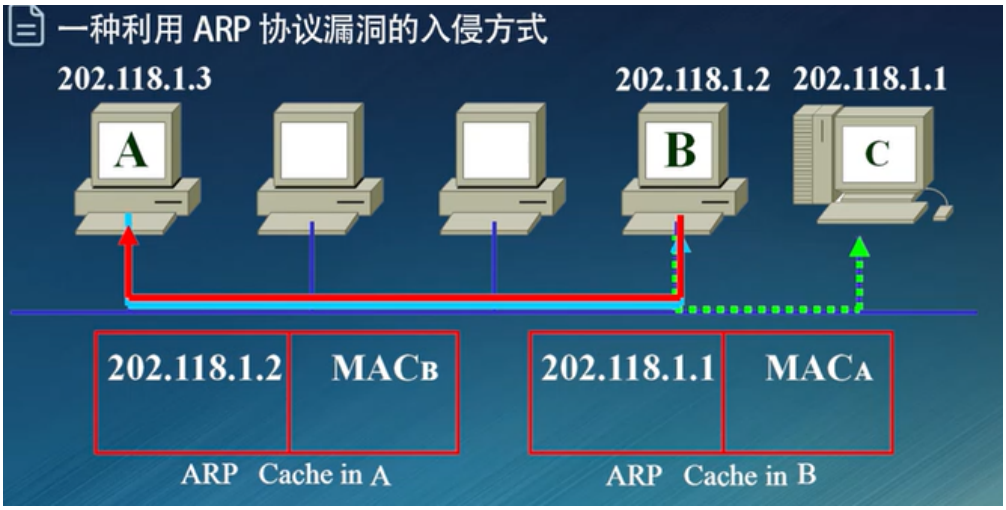
1. 首先给参与通信的每台主机都分配了一个ARP Cache，这个ARP Cache实际上就是一个IP地址和MAC地址的映射表，它就相当于进行人的记忆。
2. 同时，ARP协议还采用了**捎带技术**，比方说A向B发送数据的时候，往往意味着B过一段时间也要向A发送数据。这时当A向整个网络去发送广播ARP请求的时候，它就简单的捎带着本机的IP地址和MAC地址的对应关系也发送出去就可以了。这样，A主机当它捎带发送本机的IP地址和MAC地址的对应关系后，B主机接收到这个ARP请求，B主机就可以先将A主机的IP地址和MAC地址的对应关系存放在本机的ARP Cache中，然后B主机再向A主机返回自己得IP地址和MAC地址对应得关系，由A主机存放在本机得ARP Cache当中。这样一次请求和应答A，B两台主机都知道了对方得IP地址和MAC地址得对应关系。接下来，两台主机就可以再继续通信了，并且在通信过程当中，就可以避免再次发送广播ARP请求了。

5. ARP协议得工作原理小结

- ARP协议进行得是IP地址和MAC地址得转换
- 首先查找本机得ARP Cache，如果存在记录则直接使用本地得记录
- 如果ARP Cache当中没有记录，则发送ARP请求报文进行询问
- 被询问得主机由于捎带技术得原因，会记录下请求方得IP地址和MAC地址得对应关系，并返回应答

- 询问得计算机收到应答后，修改本机得Cache内容
- ARP工作原理是基于**"信任机制"**而工作的

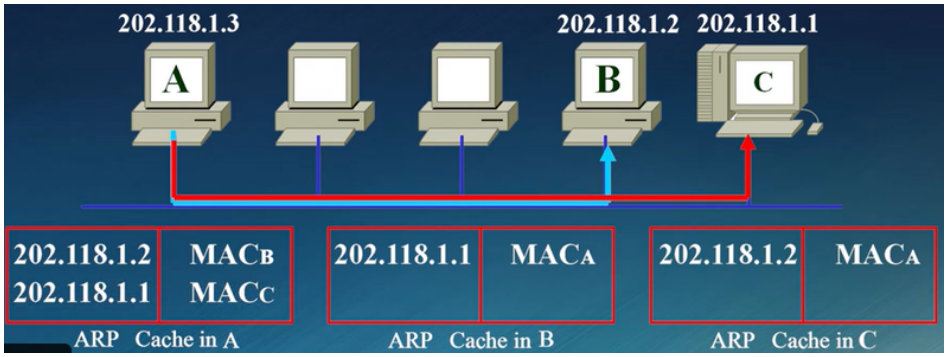
6. 基于**信任机制**/利用ARP协议漏洞的入侵方式



1. B，C之间进行敏感数据的操作，这时A主机想要冒充C来与B进行通信。A主机IP地址为`202.118.1.3`，它就可以主动的向B主机发送一个虚假的ARP请求，告诉B主机`202.118.1.1`所对应的MAC地址是我A的MAC地址。当然B主机比较老实，高速A主机`202.118.1.2`是我B主机的MAC地址，这时候B主机试图向C主机发送数据，也就是向`202.118.1.1`发送数据，实际上这个数据会发送给A。因为B主机会把`202.118.1.1`所对应的A的MAC地址进行使用，最终数据就发送给了A主机

2. 上述还存在一些其他情况，B，C主机的管理员可能会在平时进行沟通。很可能B主机向C主机发送数据后，管理员会高速C主机说，我把这个数据已经发过去了，这时候C主机可能发现没有接受到数据从而发现A主机所进行的攻击.

3. 上面的攻击模式可以更进一步：A主机先向B主机发送一个假得ARP请求，告诉B主机`202.118.1.1`所对应的MAC地址是A的MAC地址；接下来A主机再向C主机发送一个假得ARP请求告诉C主机`202.118.1.2`所对应的MAC地址是我A的MAC地址。这时候可以发现A主机出于一个镶嵌再B，C主机之间的位置。如果A主机上运行一个非常良好的转发程序---也就是说当B把数据给A的时候，A把数据传给C；当C把数据传给A时，A再把数据传给B。这样B，C之间的所有数据都会被A主机截获，甚至于A主机可以篡改B，C之间的所有数据通信，这样所造成的危害及其严重。------这就是ARP协议所面临的非常严重的攻击漏洞



7. Arp命令
    1. 从上面可以发现ARP Cache非常关键，那么如何获取ARP Cache的内容呢?
    2. 任何一个操作系统都有一个ARP指令，比如Windows下，运行`arp -a`时，就可以获取本机的ARP Cache里面的值。那么这个ARP Cache里面所显示的值是什么值呢?

```
C:\>arp -a

Interface: 192.168.56.1 --- 0x5
Internet Address      Physical Address
   Type
192.168.56.255        ff-ff-ff-ff-ff-ff
  static
224.0.0.22            01-00-5e-00-00-16
  static
224.0.0.251           01-00-5e-00-00-fb
  static
224.0.0.252           01-00-5e-00-00-fc
  static
239.255.255.250       01-00-5e-7f-ff-fa
  static

Interface: 192.168.2.190 --- 0xf
Internet Address      Physical Address
   Type
192.168.2.1           d8-c8-e9-80-b6-d9
  dynamic
192.168.2.255         ff-ff-ff-ff-ff-ff
  static
224.0.0.22            01-00-5e-00-00-16
  static
```

```
16  224.0.0.251           01-00-5e-00-00-fb
      static
17  224.0.0.252           01-00-5e-00-00-fc
      static
18  239.255.255.250       01-00-5e-7f-ff-fa
      static
19  255.255.255.255       ff-ff-ff-ff-ff-ff
      static
```

3. 它显示的是近期与本机通信过的对方主机的IP地址和MAC 地址的对应关系

8. ARP协议的漏洞如何防范

   - ARP协议的漏洞很大一部分原因是ARP协议支持的IP地址到MAC地址的动态映射。要想在一定程度上避免这种问题，就需要使用**静态地址映射**。
   - 实现静态地址映射的方案也很简单：
     1. 我们可以使用ARP指令的-S参数来实现IP地址到MAC地址的静态映射关系。`C:\>arp -s 157.55.85.212 00-aa-00-62-c6-09`。
     2. -s:static
   - 这种方案并没由完全解决ARP协议的漏洞，是通过牺牲ARP协议的方便性来获取安全性。

四、IP协议

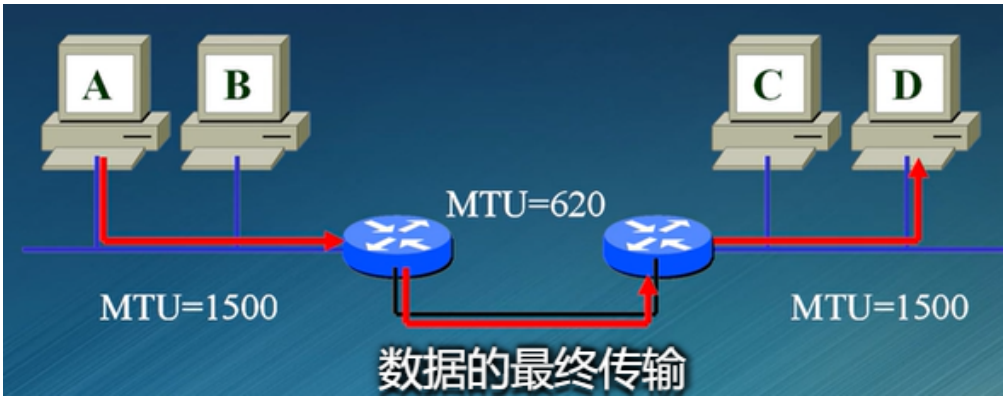
1. IP数据报（datagram）：是IP协议的PDU（协议数据单元），是Internet上数据传输的基本单元。
2. 路由器在不同的网络间转发IP分组，实现IP数据报的“寻路”功能。

- 

3. 上图是**IP数据报文的格式**
    1. Version：版本号，目前为4（IPV4）
    2. HLen：Header length---首部长度，以4字节为单位记录（比如20个字节的首部，记录5）
    3. TOS：Type of service---服务类型字段。这个字段8比特中，前3个比特是优先级比特（也就是可以给出IP数据报的8个优先级），其余的5个比特中，有4个比特是目前有意义的，分别为D T R C，最后一个比特是保留比特
        - 

        - D为1：低时延服务
        - T为1：高吞吐率的服务
        - R为1：高可靠性服务
        - C为1：低费用服务
        - 特定情况下，只能有一个为1
    4. TotalLength：总长度，以字节为单位

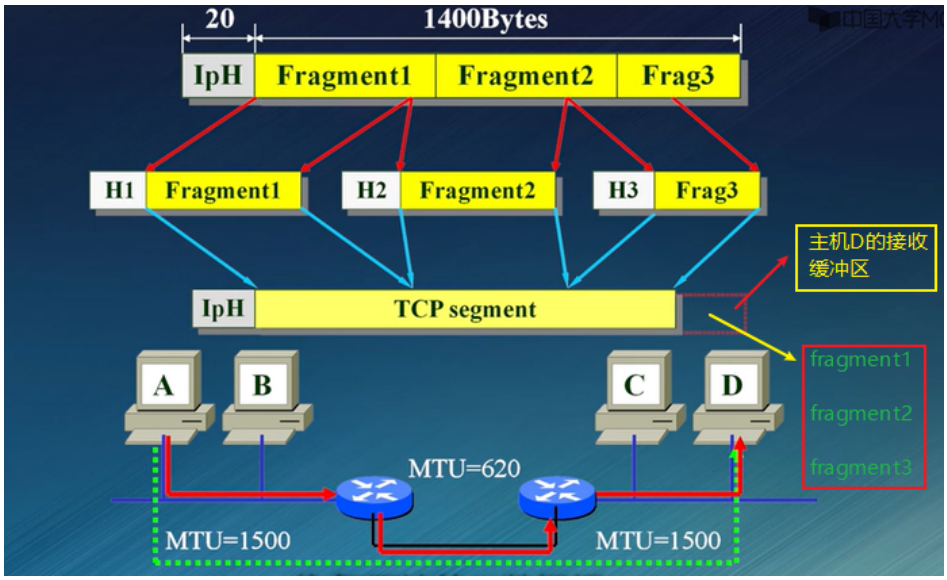
5. TTL：Time To Live----网络数据传输中，可能会出现异常情况，使得IP数据报误入歧途，走到某一个未知的网络中，并且找不到出口，这个时候的IP数据报即废弃的IP数据报，那么在Internet当中，会产生大量的废弃IP数据报，如果这些废弃的IP数据报不及时处理掉，就会导致在Internet当中塞满这种废弃的IP数据报，这时候TTL值会发生作用。在初始时，当一个主机发出IP数据报时TTL=255，每经过一个路由器，TTL的值就减1，当路由器知道TTL=0时，IP数据报就会被处理掉。这样就能避免在Internet当中出现大量的废弃的IP数据报。
6. 协议：指这个IP数据报它的数据部分所封装的高层的协议类型
7. checksum：首部校验和字段。主要功能是对IP数据报的首部进行校验和的计算
8. option：选项/填充字段。一般情况，IP数据报的首部不携带选项字段，除非某些个在特殊应用的场合。这时候IP数据报它的首部长度会大于20字节；如果IP数据报首部不携带option字段，IP数据报首部长度默认为20字节。
9. 还有三个字段没有包括进来：identification、flags、fragment offset----标识、标志、分片偏移量。这三个字段与IP数据报的分片相关的字段

4. IP数据报文的分片
    1. 分片的问题：IP数据报是被链路层的帧所封装的，因此IP数据报的长度实际上受限于数据链路层的帧的长度。而数据链路层的帧的最大长度一定不是无限长的，上限值即MTU。这个最大长度最终限制了IP数据报所能携带的数据的数据量。因此在传输过程中可能发生----由于不同部分的网络它的MTU不同，可能会导致网络传输当中出现分片的问题。
        - MTU：最大传输单元，指局域网中的最长帧。
        - 

- 如上图，A，B所在网络和C，D所在网络它们的两端网络MTU=1500，但是这两个网络之间，通过路由器连接的这段中转网络它的MTU是620字节。某一时刻A主机发送一个IP数据报时，长度很可能会超出中间网络的MTU。这时候，通过中转的MTU=620的网络进行中转时，就会面临中转网络的MTU小于要传输的IP数据报的长度，此时要想完成数据的传输，就要在**路由器**上对IP数据报进行分片。

2. 小结

    1. 不同局域网的MTU是不同的
    2. 当较大的报文经过MTU小的局域网之前，应该对该报文进行分片
    3. 分片由路由器完成
    4. 分片时，应该对IP数据报的数据段进行操作，IP数据报的首部要进行微调，并不参与IP数据报的分片过程

    - 

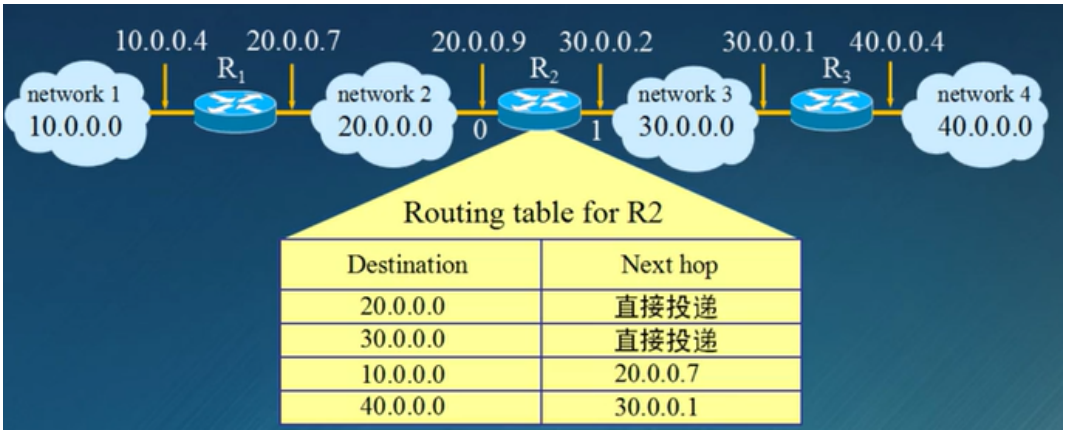
    - 如上图，A主机想要把一个数据报发给D主机。初始的时候，A发出的数据报由于它要最大限度地使用网络地MTU，因此可能发出这样一个IP数据报---首部长度20字节，数据部分为1400字节，总长度1420字节。这个数据报就会被A主机发到中转路由器，路由器接受到这个IP数据报时，下一段网络地MTU=620小于要传输地数据报地长度。
    - 此时，路由器会将IP数据报的数据部分（**注意，不含首部**）分成若干份。让前两份数据都是600字节，最后一个200字节，整成若干个小数据段发送。**但是，显然是不能直接发送的，因为最后还要进行排序和连接。**

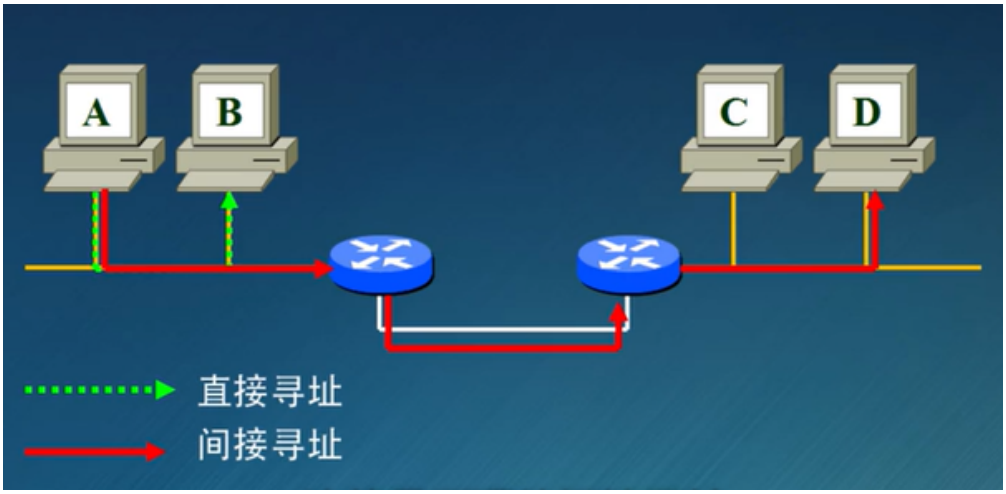
- 给每个数据段前面加上一个分片的首部，通过分片首部的数据来描述和记录当前分片在原始IP数据报中的位置等相关信息。一旦组合后，原来分段后的数据前面加上20字节的首部就形成了一个IP数据报，恰好能在下一段网络中转发出去
- 接下来，当这个数据通过重复的转发，到达接收方的D主机之后，D主机需要将接受到的分片进行重组，恢复为原来的IP数据报。对于D主机来说，它需要一个**接收方的缓冲区**。
- 接受方缓冲去区中，它一旦接到一个分片，会分析首部的信息，并且把对应的分片的数据放到主机D的接收缓冲区的对应位置中，把这个数据重新组合起来，然后加上原始的首部，恢复原始的IP数据报。这样D主机就可以接到分片重组后的数据。
- 从上面可以知道，每一个分片需要记录它的分片偏移量
- 

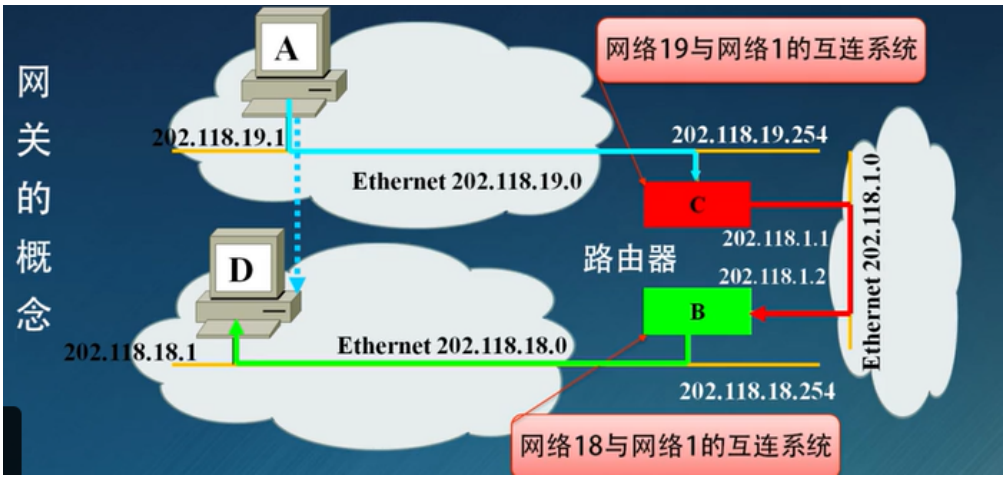
- 分片的重组
    1. 位置：目的主机
    2. 依据：IP对数据报中的字段（标识、标志、分片偏移量）
- 标识（identifier）：重组时，同一原始数据报的标识。当分片的时候，每一个分片的identifier都和原始IP数据报的标识字段相同
- 标志（flag）：`[][DF][MF]`，总共三个比特，第一个比特保留。MF比特为1时表示“还有分片，当前分片并不是最后一个分片”，MF=0表示本分片是最后一个分片。DF为1时表示“当前IP数据报不能进行分片”。

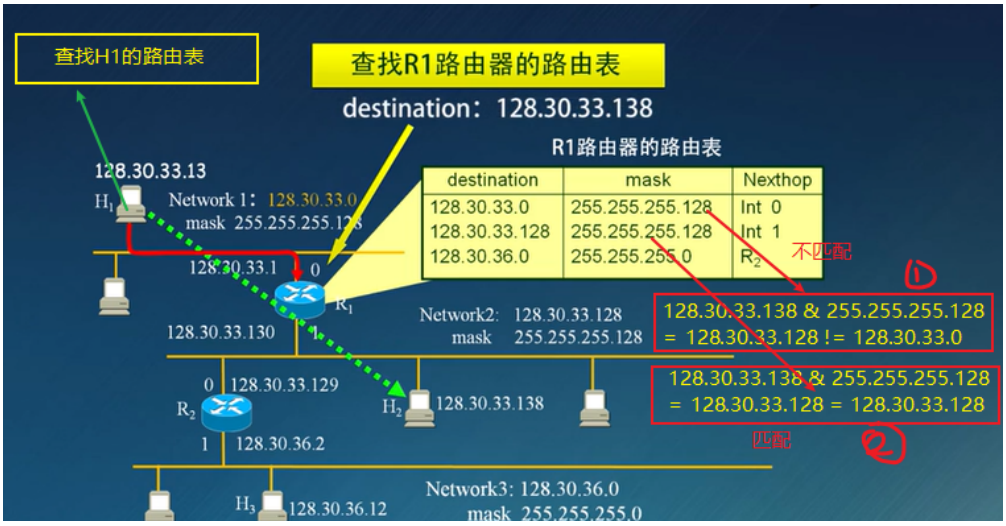
- 片偏移：某片在原分组数据中的相对位置。注意：在IP数据报首部当中记录的时候是以8字节为单位进行记录的---真正记录在分片偏移量的值需要把真实值除以8才能获得---意味着每一个分片的长度为8字节的整数倍。

五、路由与寻址

1. IP路由主要是要解决在网络中传输的**最佳路径**的问题。
2. IP路由
   - 

     1. 路由器转发数据的时候，以**路由表**为依据。（路由表类似带有距离的路标）
     2. 直接寻址（源与宿在相同的网络）：在物理网络内部确定主机---主机的数据传输路径。
     3. 间接寻址：源与宿不在相同网络
     4. 直接寻址和间接寻址的关系
        - 确定到达目的的网络的数据传输路径
        - 在目的网络中用直接寻址方法到达信宿
   - 

5. 如果需要进行间接寻址，就需要经过中转的路由设备的转发。很有必要的在中转的每一站中都要去设置它的下一站的转发地址---（下一站的转发地址：网关）。尤其是在主机的相关配置中，网关的概念需要明确。因为在给主机配置IP地址的时候，都要指定这个主机的网关的IP地址。
6. 网关
   - 

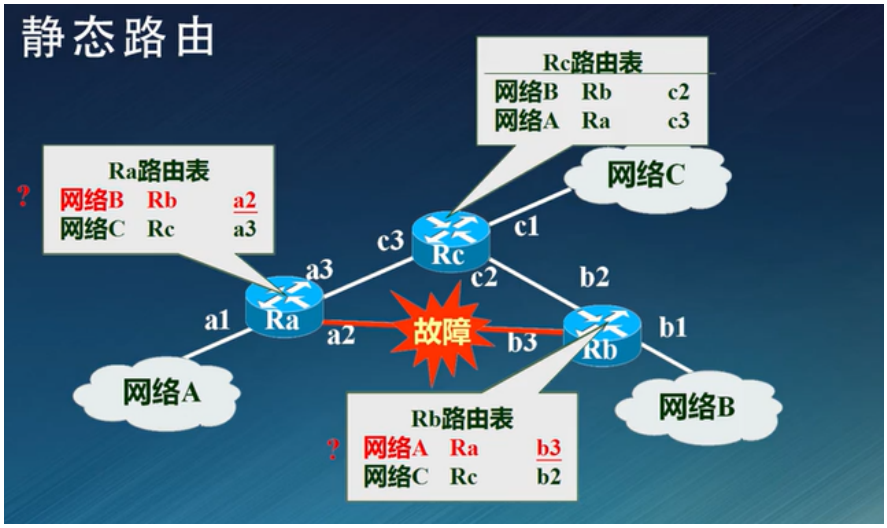
     - 直白点：指的是这个设备所在网络的默认出口地址
7. 一个IP数据报在一个网络中，经过中转路由器转发的时候的转发策略以及如何工作
   - 

     - 比方说在当前的网络当中，H1要向H2发送数据。
     - 作为H1这个主机，首先需要查找本机的路由表，把这个数据发给他所在的默认网关---就是它出口路由器R1的IP地址
     - 接下来，R1路由器需要去查找本机的路由表。首先对第一行的掩码与目的IP地址进行与运算，发现结果与路由表对应的目的IP地址不匹配；然后进行第二行的同样运算，发现匹配。所以将数据从接口1（路由表右边）转发给H2这个主机。
8. 实际传输过程中，IP数据报经过路由表时，并不知道网络的整体状况，只能看到一个路由表，这是决策过程如下：

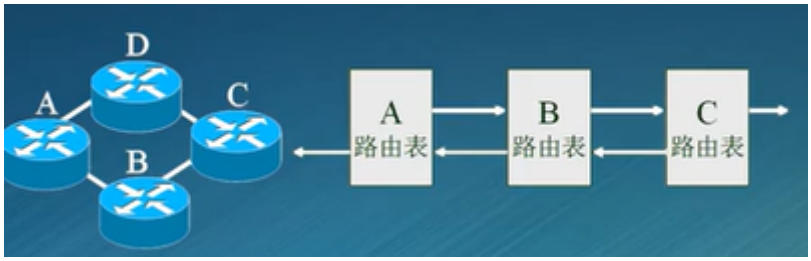
- 

  - 路由器会将目的IP地址同子网掩码从第一行开始，一行一行进行与运算，并与表格的目的网络列地址进行比对，找到相同的一行，然后将数据从对应的从下一站转发出去。
  - 注意，最后一行是默认路由，在任何情况下都会匹配。即前面所有的信息都没匹配时，最后一个一定会匹配。

3. 路由表的维护
    1. 选择最佳路径
        1. 静态路由
            1. 由网络管理员来设置路由表
            2. 需要管理员经常手工维护
            3. 简单、有效，适用于结构简单的网络
            4. 不适合于拓扑结构和传输流量经常改变的复杂网络
            5. 最大的问题：当网络状态发生变化时，路由表的信息仍然保持不变，这时候会导致网络运行出现异常

- 

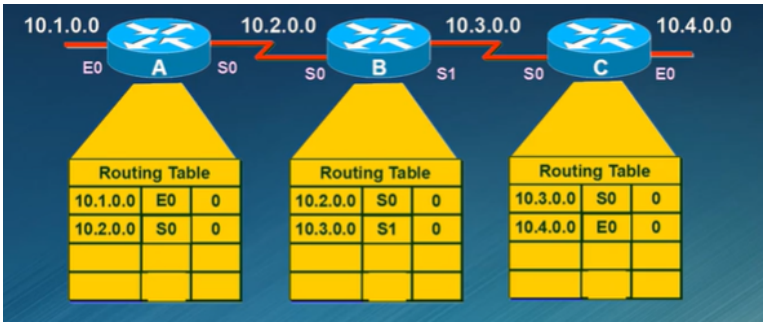
- 上图中，三台路由器以环形方式连接三个网络。所构成的网络分别配置路由之后，网络是完全通畅的，即A，B，C之间可以相互传输数据。但是网络可能会出现故障，比如Ra和Rb这两台路由器所连接的网络出现了故障，网线断了。这时会导致这个网线所对应的两台路由器的两个路有消息出现失效。这时候对于Ra和Rb这两台路由器来说，它们并不知道实际上它们可以通过Rc来转发来完成数据的传输。这时候如果管理员不进一步的去手工维护路由表的话，那么对于网络A到网络B之间的所有数据传输都是无法进行的。只有当管理员进行手工维护后，修改对应路由信息，这个网络才能保证畅通。
- 当网络很复杂是，网络出现故障时，网络管理员进行手工维护是困难极大的。

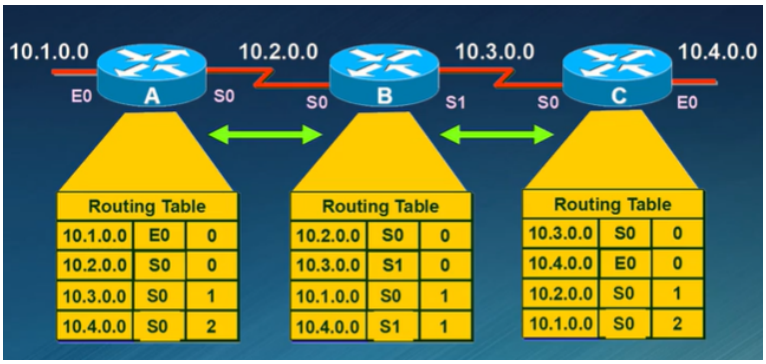
2. 通过算法实现动态路由
    1. 距离向量算法---Distance-Vector
        1. 基本概念
        - 

            1. 每个路由器都会周期性的向它相邻的路由器发送一个它到所有其他路由器的距离的向量
            2. 距离向量：方向和对应的路径的代价
            3. 每一台路由器接收到它相邻路由器发来的**距离--向量**的数据后，将会启动对应的算法，来更新路由表。
        2. 更新路由表的原则
            1. 发现新路由，则更新路由表
            2. 发现更短路由，则更新路由表
            3. 发现必经之路上的距离有变化，则更新路由表
    2. Router Information Protocol（RIP协议）：
        - 使用D-V算法。这个所谓的距离在RIP协议中是以站点数即跳数HOP为度量的D-V算法。默认一个路由器就是1跳，并且在RIP协议HOP=16时为距离无穷大，目的不可达。即RIP协议所支持的网络它间隔最多15台路由器。

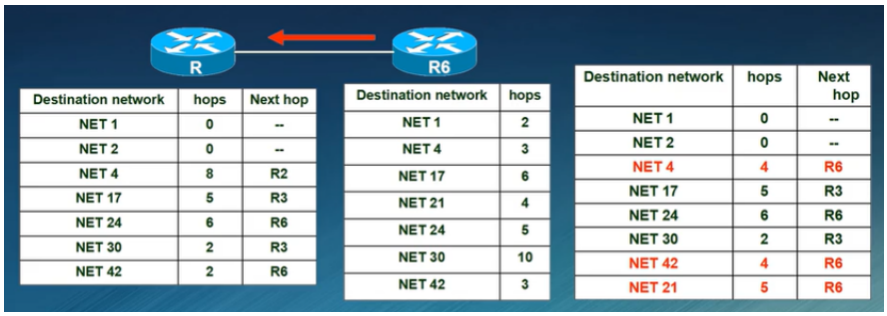
刷新周期为30s----相邻路由器传输的路由表的间隔时间为周期性的30s。

- 1988年RFC1058正式定义
- 适用于小型网络的内部路由协议

1. 跳数---HOP
    1. HOP数可以认为时路由器的个数
    2. IP路由系统中使用HOP来计量站间距离，即间隔的HOP数越小，站间距离就越短
    3. 特殊情况下，可以定义1个路由器的HOP数大于1
2. RIP协议的动态特性
    1. RIP协议建立路由表的初始过程
        - 

            - 上图有A，B，C三台路由器，连接了四个网络。对于每台路由器来说，在刚刚启动时，都会在自己的路由表中记录下来本路由器所直接连接的网络信息。
            - 初始化后，对应网络之间想要进行数据传输是不可能的，因为路由信息并不完整
    2. 当三台路由器的初始化阶段结束后，接下来会启动RIP协议的距离向量算法来传递自己的路由表。
        - 

            - 比如，A，B之间交换路由表，当A接到B路由器的路由表后，发现B路由表中它到10.3的这个网络的路由信息A是不知道的，遵循新路由则更新，所以A路由器需要把10.3.0.0这个路由信息添加到本机的路由表中。它接到B到10.3.0.0是0

跳，那么A就需要先把数据发给B，然后再到目的的10.3.0.0.这时就需要增加1跳，后续则同理。
        - 经过几个更新周期后，路由表稳定下来（受列）
    3. 具体例子描述更新三原则
        - 

Destination network	hops	Next hop
NET 1	0	--
NET 2	0	--
NET 4	8	R2
NET 17	5	R3
NET 24	6	R6
NET 30	2	R3
NET 42	2	R6

Destination network	hops
NET 1	2
NET 4	3
NET 17	6
NET 21	4
NET 24	5
NET 30	10
NET 42	3

Destination network	hops	Next hop
NET 1	0	--
NET 2	0	--
NET 4	4	R6
NET 17	5	R3
NET 24	6	R6
NET 30	2	R3
NET 42	4	R6
NET 21	5	R6

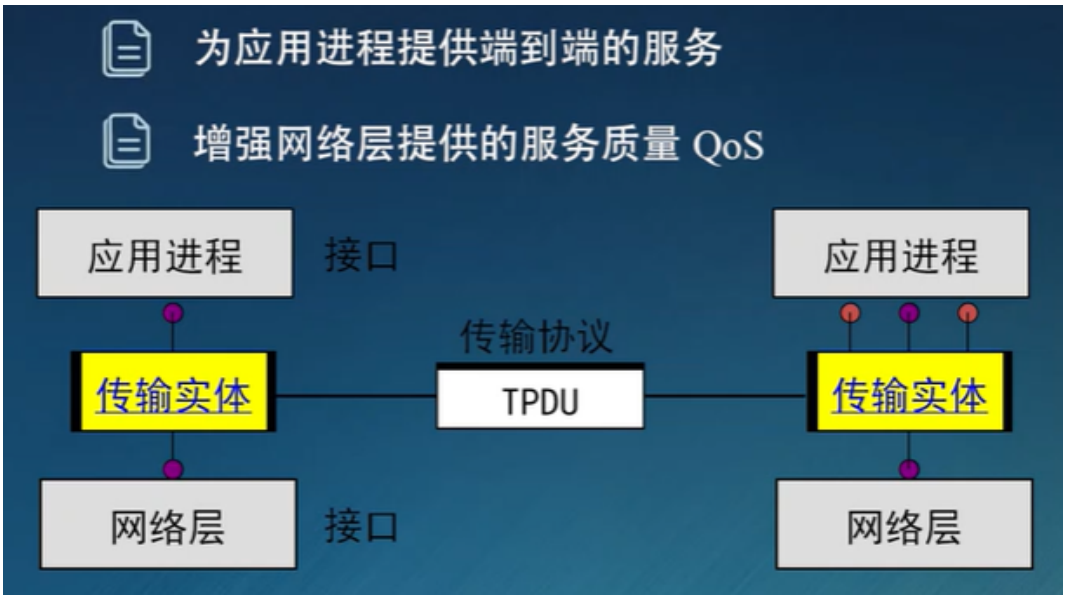
        - 某一时刻，R路由器的路由表如左图所示，那么它的邻居R6向自己发送了一个路由表。
        - 根据R6的路由表NET1的信息，可以发现不需要R的NET1 hops=0，所以不需要更新✕.
        - R6的NET4的hops=3，而R的NET4的hops=8，所以需要更新✔.
        - R6的NET17的hops=6，而R的NET17的hops=5，不需要更新✕.
        - R6的NET21的hops=4，而R没有NET21的信息，所以需要更新✔.
        - R6的NET24的hops=5，而R的NET24的hops=6，所以需要更新✔.
        - R6的NET30的hops=10，而R没有NET21的hops=2，不需要更新✕.
        - R6的NET42的hops=3，而R的NET42的hops=2，不需要更新✕.

七、传输层

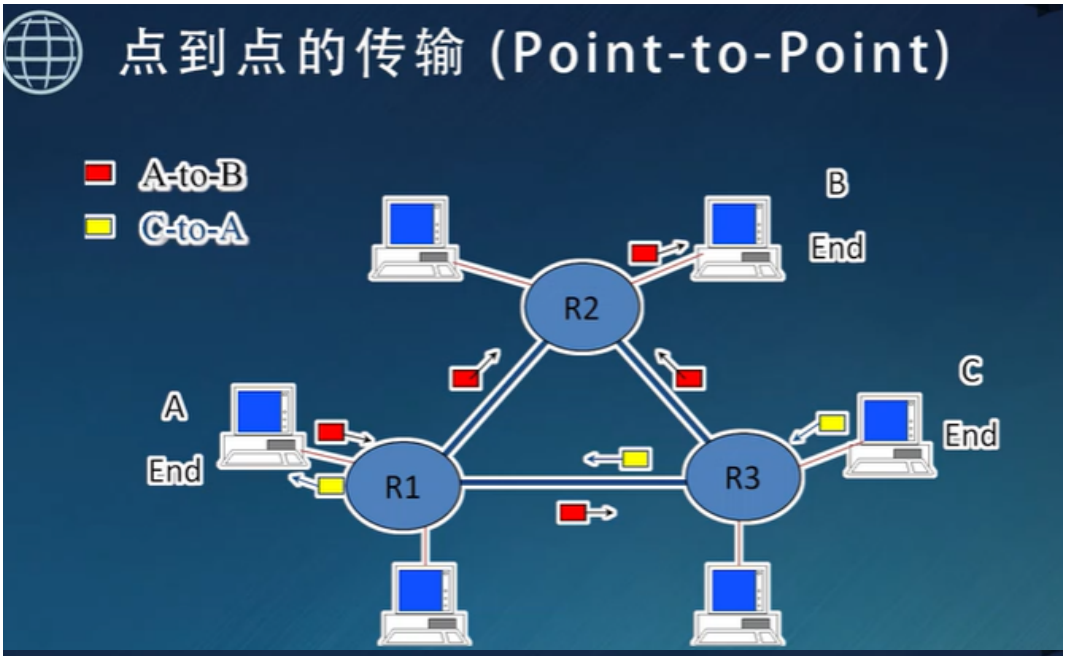
一、传输层的作用

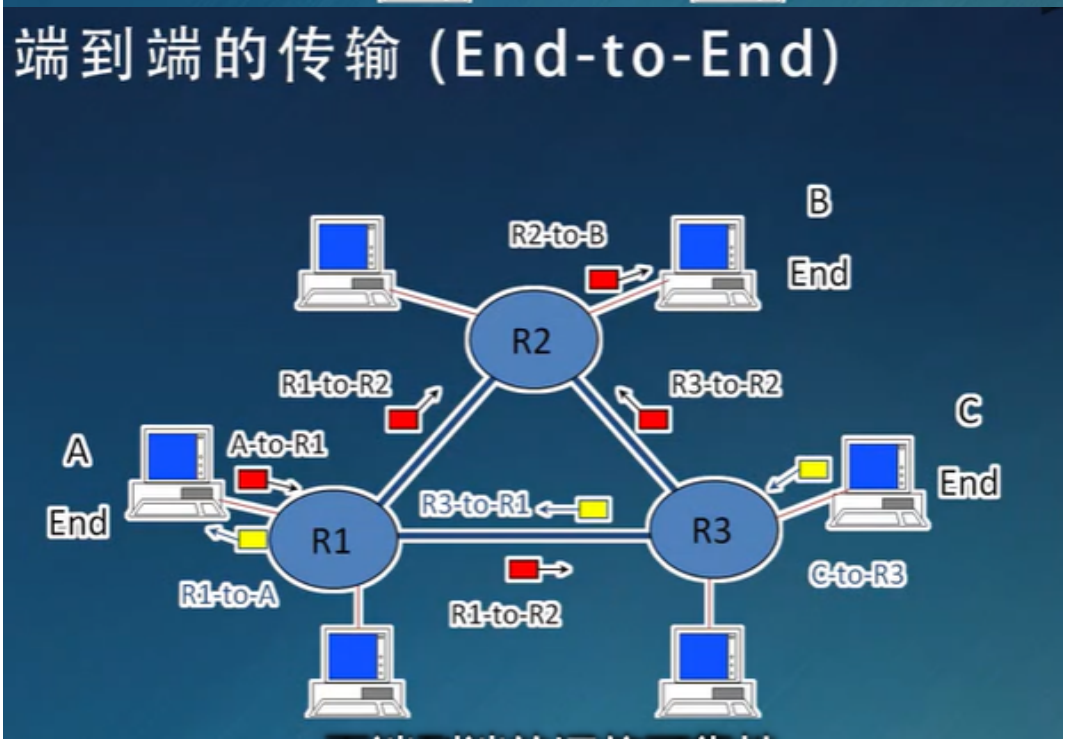
1. 上述网络层的问题解决后，并不是万事大吉，还涉及两个问题：
    1. 当前计算机安装的操作系统大多是多任务操作系统，也就是说同一台主机上允许同时运行多个具有网络通信功能的进程，那么通过网络层寻址到达目的主机的报文是如何被目的主机精准的交付给目的进程的呢?

2. 网络层为用户提供了传输服务并不都是可靠的，由物理层知网络层提供的两种服务为可靠的蓄电路服务，另一种是不可靠的数据报服务。若网络层的协议采用了数据报服务，那么用户交给这个协议实体的数据是否就没有服务质量的保证了呢？

2. 传输层在网络体系结构中的重要位置
   - 

   - 传输层位于应用层的下端，为上层的应用进程提供传输层的实体传输服务，同时被下一层的网络层调用它的功能来实现点到点之间的主机之间的通信。
   - 传输层之间传输的协议数据单元称为TPDU，这里用message代替
3. 传输实体（Transport Entity）：在收/发两端的传输层实现对等实体通信的硬件或软件。
   - 利用网络层提供的点到点的分组传输服务
   - 向高层提供端到端的TPDU（传输协议数据单元）传输服务（报文的可靠投递服务）
4. 为什么会出现网络层的数据丢失和乱序呢？网络层的下一层，也就是数据链路层不是已经保证数据真的可靠传输了嘛？而网络层的协议数据单元也就是分组时封装在数据帧当中传输的，而帧的传输的可靠性已经由数据链路层保证了，为什么还要考虑分组的丢失和乱序的情况呢？必须要先明确两个概念---点到点的传输、端到端的传输

- 

- 

- 主机A要向主机B发送一个报文，从上图可以看出，可以选择的路径很多。比如R1到R2，或者R1，R3，R2；同时如果主机C要向主机A发送一个报文的话，它也可以选择多条路径。那么在这些路径里，假设主机A和主机B之间选择了R1到R2这条路径，那么很显然A要向B投递一个报文时需要有四段这样的路程。首先A要形成一个报文；然后A将报文从A传递到R1；再从R1传递到R2；再从R2投递到B。那么在这个过程中我们称A到B之间的通信是端到端的，而A到R1，R1到R2，R2到B的通信是点到点的通信。
- 我们知道链路层可以保证一定的通信的可靠性，那么链路层保证的是点对点的通信可靠性。而端到端的通信可靠性是由传输层来保证的。

5. 所以，传输层的存在使应用进程无需关心网络层的服务质量，而可以成功的将应用程序从源端进程可靠的投递到目的进程。那么传输层的协议应该如何设计才能实现不可靠的分组投递基础上完成可靠的报文传输的呢？又是如何实现进程寻址的呢？

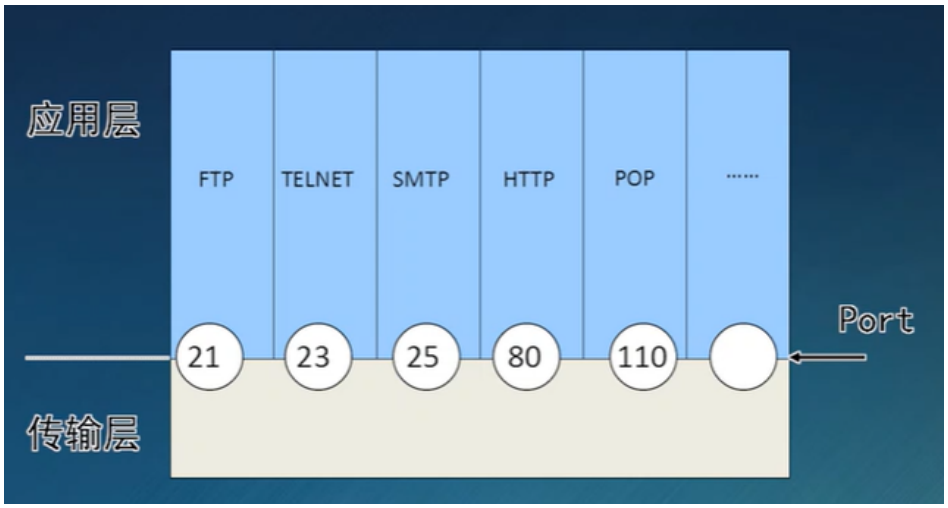
二、面向连接的TCP协议

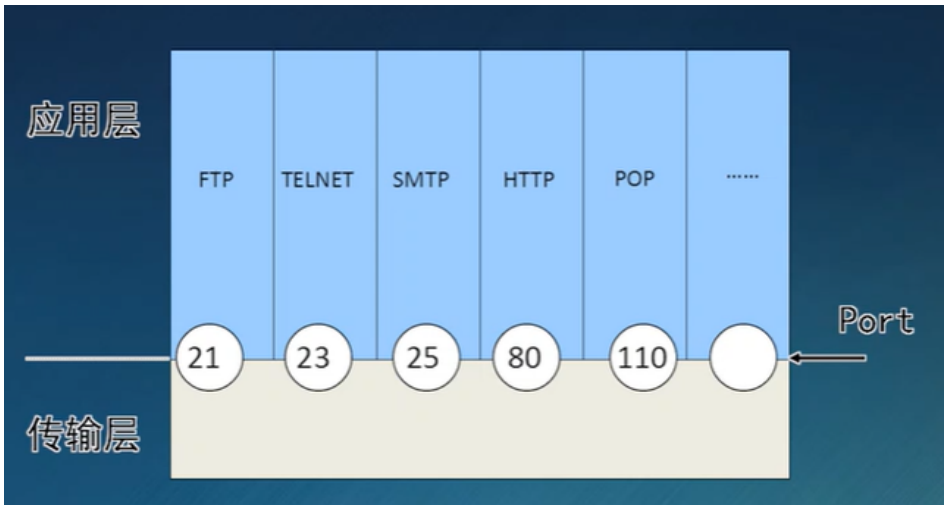
1. 传输层的TCP协议（Transmission Control Protocol），它为应用进程提供了可靠的端到端的面向连接的字节流通性，解决分组的重传和排序问题，由FC793正式定义。为Internet中许多著名的应用提供服务。
2. TCP协议的基本概念
    1. 为应用进程提供可靠的、端到端的、面向连接的字节流通信的协议
    2. 它是利用网络层IP协议提供的不可靠的分组传输服务，在此基础上解决分组的重传和排序问题
    3. 为Internet的许多著名应用提供传输服务
3. TCP连接的性质
    1. 面向连接：就是在传输数据之前要明确知道接收方的存在，以及要与接收方商议传输的一些性质比如缓冲区的大小，发送序号的初始值等等。
    2. 全双工：在TCP连接建立好后，发送方和接收方是可以同时双方向进行数据传输的
    3. Unicast：也叫单波通信协议，只支持两个端点间的一对一的通信，不支持多波（multicast）和广波（broadcast）的通信
    4. 面向字节流：TCP协议本身是面向字节流的流逝协议
4. TCP协议的传输实体
    - 

    - 可以是一个软件：也就是一个用户进程

- 也可以是操作系统中的某一部分，用来管理TCP字节流，实现与IP层的接口

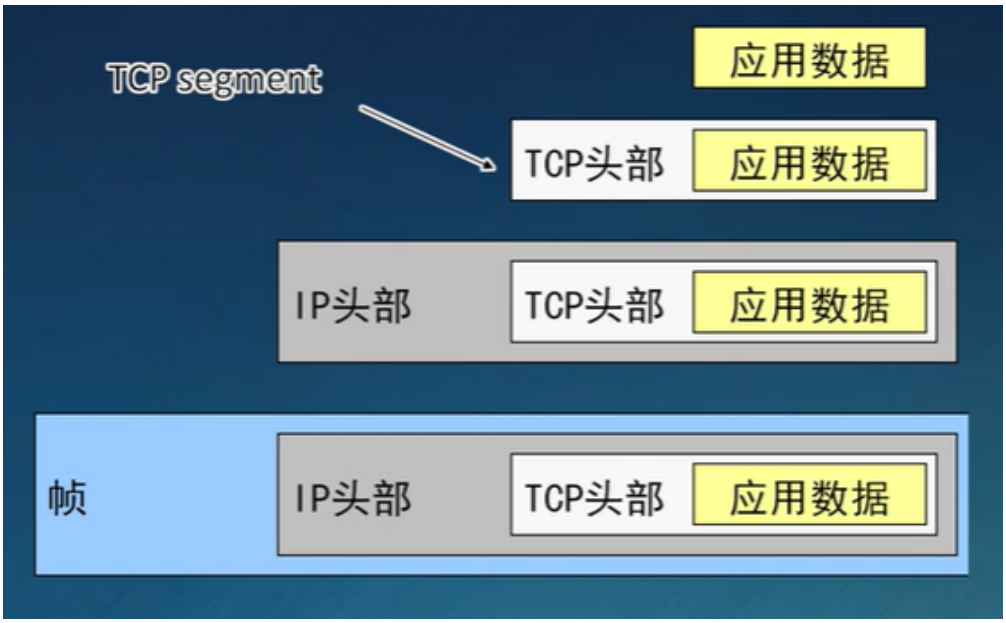
5. TCP的端口
    1. 端口的定义
        1. 端口是用来标识一个进程的，在一台主机当中，我们必须能够区分多个进程，而必须给进程进行编号，这个编号我们称之为端口（port）
        2. 在Internet的TCP协议当中用16bit区分2^{16}个端口
    2. 端口的分配
        1. 采用众所周知的端口：全局分配，用于标准服务器，取值小于1024
            1. 优点：能简单知道与你通信的其他进程的端口号，这样比较有利于进行点到点的通信
            2. 缺点：没有在国际组织注册的进程不会获得端口号，这样新的进程的出现它的端口号的获得会变得非常复杂。
        2. 临时分配端口：由本地的操作系统在启动进程时临时为这个进程分配一个端口（本地分配，主机建立连接时为用户进程动态分配端口号），等这个进程结束使用后会释放这个端口号，取值大于1024。
            1. 优点：可以将端口号资源充分利用
            2. 缺点：当临时启动一个端口号时，与你通信的其他主机并不知道你的端口号是什么，所以你没有办法将你临时获得的端口号通知给对方
        3. 解决方法：将临时端口号和众所周知的端口号进行搭配分配。对于标准服务器的服务进程，为它分配固定端口号；对于临时启动的客户进程，由操作系统为它临时分配端口号。当前的TCP/IP协议中，小于1024的端口号已经全部分配给标准服务器了。大于等于1024的端口号可以由操作系统动态的分给客户进程来使用。
    3. 用套接口创建通信端点
        1. 什么是套接口socket：有了端口号后，我们可以知道在internet当中访问唯一进程的方法就变成了两部分：第一部分就是要通过对方的ip地址访问到对方主机。第二部分是然后再根据端口号访问到对方主机的某个进程。这种分两步走的动态模式称为**socket通信**。socket通信是由Unix操作系统首先提出来的。

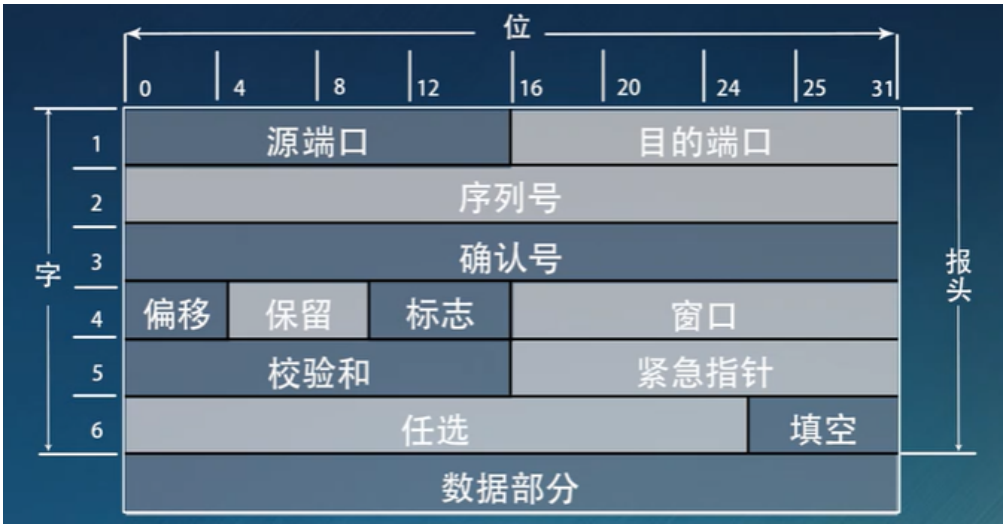
2. socket构成就是由端口号和主机ip地址动态绑定形成的。
3. socket通信当中，端口号是一个重要指标，用于进程寻址的。
    - 

4. FTP应用连接端口举例说明进程间如何利用端口号实现进程寻址
    - 

    - 假设这里有个FTP服务器，我们知道FTP服务器打开的服务端口是21号端口，也就是一个众所周知的端口。假设它的ip地址是18.22.67.7。客户端假设它的ip地址为128.8.6.4.194，那么它打开的端口应该是一个临时端口，也就是操作系统为它动态分配的一个端口。当客户端连接服务器端时，作为初始的连接请求，客户端连接请求会将21号端口作为自己的目的端口号，这样的话它就会将自己的源端口号也就是1234捎带给对方主机。也就是ftp服务器，这样ftp服务器就能够获得客户端的通信端口号了，从而与对方建立连接，实现1234端口与21号端口之间的连接的数据通信。
4. TCP协议的报文格式：tcp协议的报文格式分为两部分：
    1. TCP头部

    - 固定长度20字节（有些特殊的可以超过20字节（加上tcp头部的可选部分））
2. 数据：承载的是用户的数据报文
    - 适应IP的载荷能力（小于65535Byte）
    - 适应网络的MTU

5. TCP的封装
    - 

    - TCP报文格式在这里应该说是它的数据部分承载了用户的数据，并且在应用数据的基础上增加了一个TCP的头部，我们称它为一个报文/报文段。整个tcp报文/报文段再向下调用网络层功能，封装成ip分组，分别加上ip的头部。再继续向下交给链路层，这就是tcp报文的封装过程。
    - 那tcp头部包含那些字段，这些字段有什么作用呢?
6. TCP首部
    - 

1. 源端口号和目的端口号：是整个TCP报文段来自于哪一个应用进程，要到目的主机的哪一个进程去，由源端口号和目的端口号来标识出来，分别占用16个bit。

2. 序列号和确认号：TCP的报文段是要被封装在IP分组当中投递的，而IP分组在投递的过程中是有可能乱序和丢失的，为了能够使乱序和丢失的报文重传重排TCP报文段，必须自己给自己编号才行。所以这个序列号就是TCP报文段自己给自己的编号；而确认号是接收端的TCP实体。用于向对方确认已经收到了哪一个序列号的报文，而希望收到的下一个报文段的编号。
3. 偏移：偏移字段占用4个bit，这四个bit实际上是TCP报文的头部长度，单位是4字节。如果偏移字段是10，那么整个TCP报文段的长度是40字节。
4. 保留：保留字段6bit，目前没有具体应用功能。
5. 标志字段：标志字段字段也是6bit。`[URG][ACK][PSH][RST][SYN][FIN]`
    - URG----urgent：为1时，整个报文段头部中的**紧急指针字段**有效；为0，紧急指针字段无效。紧急指针字段是一个16bit的字段，里面存放的是数据部分需要被紧急处理的那一部分的最后一个字节的字节编号。
    - ACK----acknowledge：为1时，**确认字段**有效；为0，确认字段无效，即当前发送的TCP报文段只有序列号没有确认号。
    - PSH----push：指的是在整个TCP报文段中，接收方在接收了多个TCP报文段时会将TCP报文段进行缓冲，缓冲到一定长度后才会向上交给应用层去处理。当某一个TCP报文段需要被整段的紧急处理时，很显然它不希望被缓冲在接收方的缓冲区里，这时候它就将它的PSH比特设置为1，这是TCP接收方在接受到这个TCP报文段时就会不用等缓冲区满而将他上传给它的上一层应用程序来处理。
    - RST----reset：适用于当连接崩溃或出现错误连接时，用于连接复位的
    - SYN----synchronous：用于建立连接时使用。TCP协议是一个面向连接的协议，很显然，发送方在发送数据之前要与对方建立连接，这时候它就希望接收方要给它一些回应。首先它就给对方发送一个SYN=1的TCP报文，如果对方同意建立连接就向它返回一个SYN=1且ACK=1的TCP报文；如果对方不同意建立连接就返回一个SYN=1且ACK=0的TCP报文段；这样接收方就能区分发送方是否请求与它建立了连接，而发送方也可以区分接收方是否同意与它建立连接。

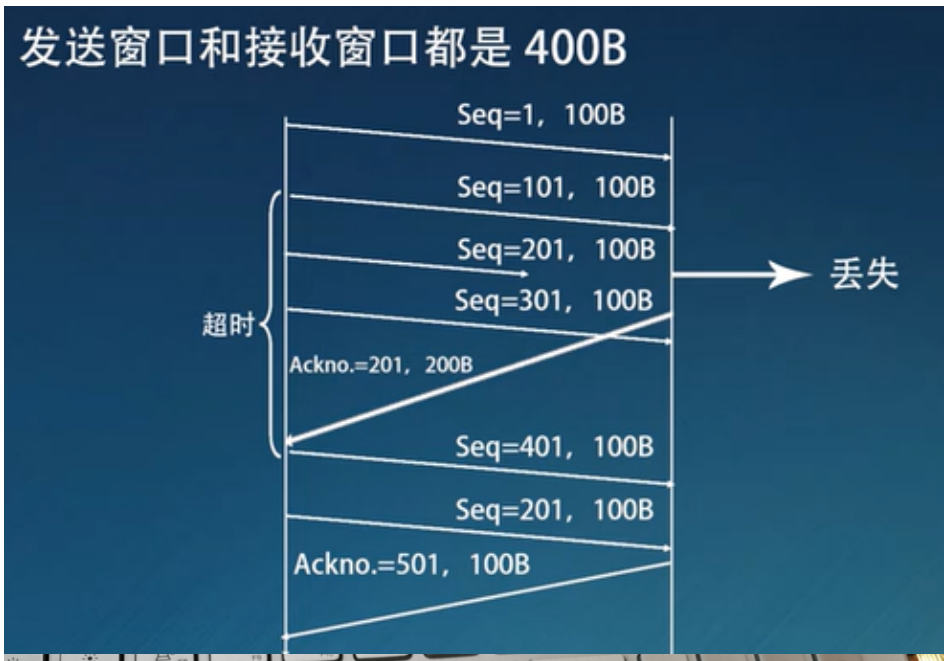
- FIN----finally：适用于断开连接时使用。当发送方没有数据要向对方发送时，它就要请求与对方断开这个连接，这时候它就向对方发送一个FIN=1的TCP报文段，如果对方也同意断开连接就返回一个FIN=1的TCP报文段，不同意则返回一个FIN=0的TCP报文段。

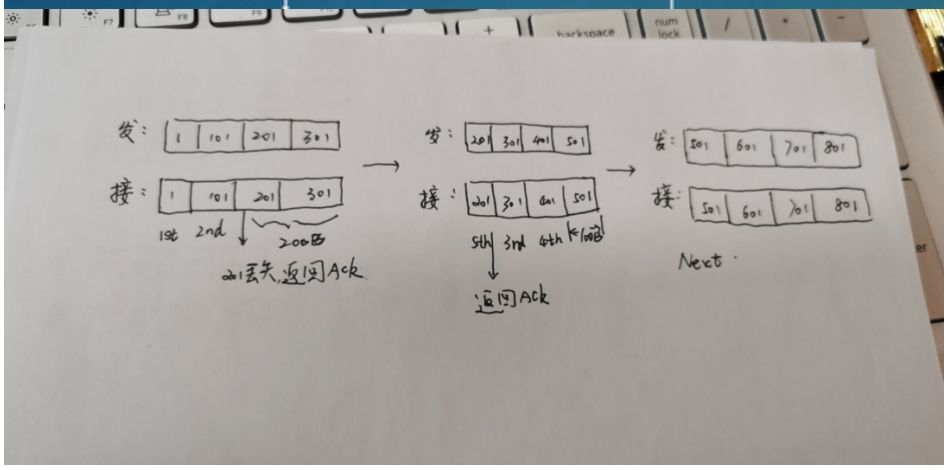
6. 窗口：窗口字段是窗口的尺寸，占16bit，是接收端向发送端发送的用于控制发送端的发送的报文段的尺寸，防止接收端的缓冲区溢出。
7. 校验和：校验和字段的用途类似于IP数据报的头部当中的校验和字段的用途。IP头部的校验和字段只校验头部，而TCP头部当中的校验和字段是对整个的TCP报文段的校验，运算原理相同。
8. 到紧急指针字段一共20字节，这是TCP头部的固定长度，也就是说任何一个TCP报文段起码要包含这20个字节的头部，其他还有任选的部分。

7. TCP的流量控制--滑动窗口

- 

1. 使用选择重传的ARQ
2. 发送段可以连续发送多个TCP报文段，只要是在发送窗口中的都可以发送走。而接收端也有一个接收窗口，只有在接收窗口中的TCP报文段接收方才会接收成功，并且返回ACK给发送方，这样发送方就可以确认这一报文段已经被成功接收。那么已经被ACK过的报文段就可以出发送端口，使得发送端口继续向下滑动。
3. 窗口字段如何实现流量控制的呢?

- 

- 

1. 发送端发送了一个Seq=1也就是发送序号等于1的TCP报文段，假设长度为100B。由于第一个报文段和第二个报文段都在发送窗口，显然可以不用等待应答而继续发送第二个报文段，假设发送序号是101，长度100B。
    - 序号为什么不是1，2，3...，？因为发送序号是字节编号，TCP是面向字节流的流逝协议，那么在每一个报文段中，它的发送序号都是这个报文段当中第一个字节的字节编号。第一个报文段的第一个字节是1的话，且长度为100B，显然这个报文段的序列号是1，下一个报文段的序列号是101。**这个第一个序列号是0还是1看文档怎么给出它的定义的。**
2. 第三个报文段Seq=201，100B。假设在传输时第三个报文段发生了丢失。显然接收方只能对前两个报文段进行确认，那么它发回的ACKno.=201，200B，也就是希望收到的下一个报文段的发送序号。

3. 这里接收端就会向发送段发回一个窗口尺寸，这个窗口尺寸就是接收端的缓冲区当中还有的剩余空间的量。有了这个发送尺寸后，发送段可以连续发送TCP报文段但是长度不能超过这200个字节。这很显然就实现了对发送端流量的控制。

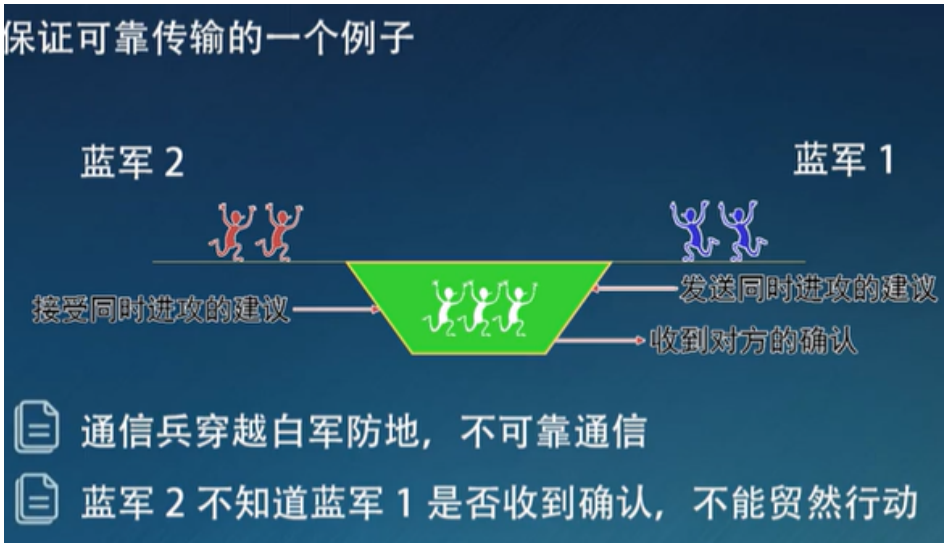
6. TCP运行原理

    1. 在TCP的运行中，还有两个问题必须考虑：一个是超时计时器的初值设定问题、第二是TCP如何利用滑动窗口原理来实现拥塞控制的
    2. 超时计时器的设定
        1. 计时器设置的过短时，当计时器超时了，这时候可能正常的一个TCP报文段还没有传到接收端，而在发送端就已经超时了，这样会导致额外的重复发送。而超时计时器设置的过长时，就是会产生已经丢失了，但是由于计时器还没有超时这样还会多等待一些时间造成时间的浪费。
        2. 加权平均进行设定
            - 至于权数多大，不同系统有不同设定，具体情况具体分析。
            1. $RTO = RTT_s + 4 \times RTT_D$.
            2. $RTT_s = (1-a) \times RTT_{s_{old}} + a \times RTT_{new}$.
            3. $RTT_D = (1-\beta) \times RTT_{D_{old}} + \beta \times |RTT_S - RTT_{new}|$.
    3. TCP拥塞控制
        1. 当TCP报文段发送过程中产生拥塞时，我们希望发送端减少发送或减缓发送，使得发送的压力不是那么大。
        2. 一般情况：在发送端设定一个拥塞窗口，拥塞窗口初值cwnd=发送窗口大小swnd。
        3. 在一个超时计时器时间内最多能发送的报文段的多少就由这个拥塞窗口来决定。
        4. 除此之外还要设定阈值，用于控制拥塞窗口的大小，拥塞窗口的大小应该小于等于这个阈值。$cwnd \leq ssthresh.$
        5. TCP报文段传输的是否顺利看：如果顺利，在规定的RTT时间里，发送段就会收到一个应答；如果在RTT时间内没有收到应答，我们可以判定线路上在传输TCP报文段时一定出现了问题，这时候认为是拥塞问题，这是拥塞窗口就要产生变化。cwnd是变化的，由小变大，初值为1.
        6. cwnd变化规则：

1. 首先拥塞窗口的初值为1。
2. 如果拥塞窗口小于所给的阈值时---$cwnd < ssthresh$，若在每一个RTT时间内，都能接受到应答报文，说明这个报文段已经这却被接受了，cwnd加倍；若超时---没有在RTT时间内收到应答，说明出现了拥塞，启动拥塞避免算法。
3. 若拥塞窗口大于所给的阈值时---$cwnd > ssthresh$，若在每一个RTT时间内，都能接受到应答报文，则cwnd加1.否则，启动拥塞避免算法。
4. 若$cwnd = ssthresh$，加倍加1都可以。

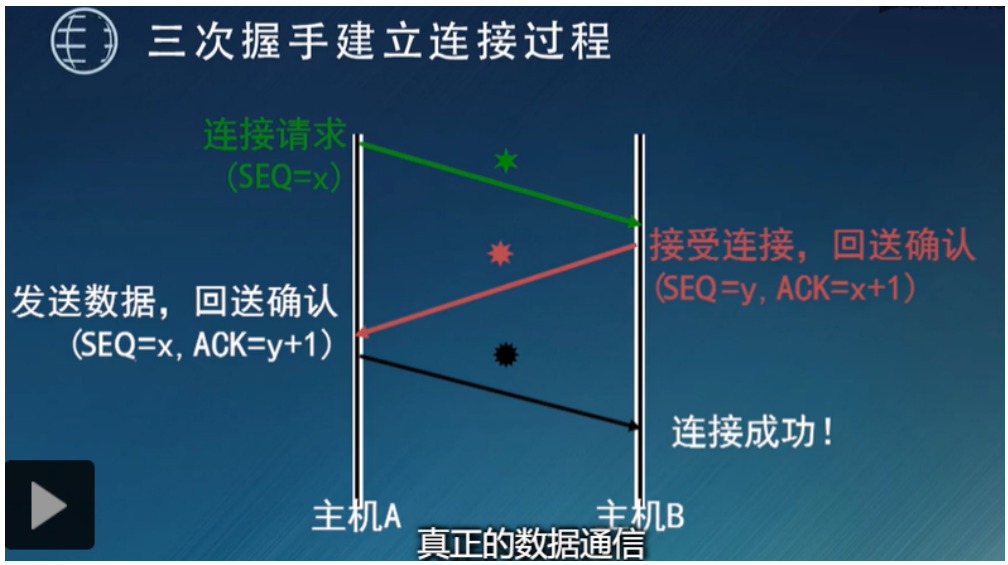
7. 拥塞避免算法（慢启动算法）
    1. 只要没有在规定时间内收到ACK，ssthresh立刻减半，但不小于2.
    2. 并且将cwnd重新置为1.
    3. 执行上述cwnd变化（拥塞控制）算法。

7. TCP的连接和释放
    1. 为了保证TCP协议运行过程中双方为这次通信预留足够多的资源，TCP采用面向连接的方法---连接之前，有一方发起连接请求，在双方明确同意的情况下，才开始数据的通信；数据传输结束后亦然。
    2. 之所以讲解TCP的连接和释放的原理，因为TCP提供的是可靠的端到端传输服务，为了能够保证一次数据传输的可靠性，必须要明确知道通信双方为这次通信准备了什么资源，这就需要在真正的数据通讯之前，双方要与对方打好招呼，为这次通信预留资源，这个过程就叫做建立连接。
    3. 如何能在网络层提供不可靠的IP分组投递的基础上，能够使对方明确的知道自己的存在，并且能够明确的为这次通信分配预定的资源呢?
        1. 典型例子----两军问题

- 

    1. 假设有两伙蓝军分别位于两个山峰之上，而它们的敌人---一伙白军位于山谷之中。两伙蓝军分别向白军发起进攻时，它们都不能取胜。如果它们协同作战则可以战胜白军。
    2. 假设蓝军派出通信兵穿越白军的阵地，去到另一伙蓝军的阵地通风报信。那么如果这个通信是不可靠的，如何找出一种方法使得两伙蓝军能够同时发起进攻，确保这次战争的胜利?
2. 上述例子就是可靠的建立连接的TCP传输所要面临的问题---也就是说当服务器和客户机即TCP连接的两端想要知道对方的状态和发起这一次连接请求时，他们的目的是确保对方知道自己的存在，并且能够确保为这次通信预留它们所需要的传输资源。

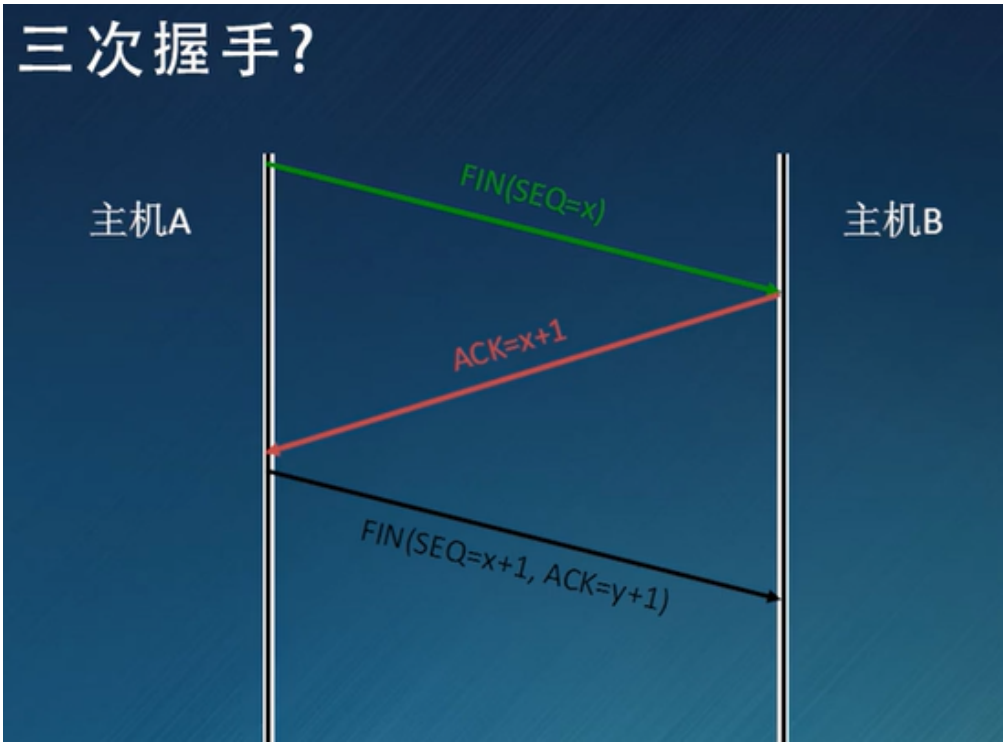
4. **三次握手建立TCP连接**

- 

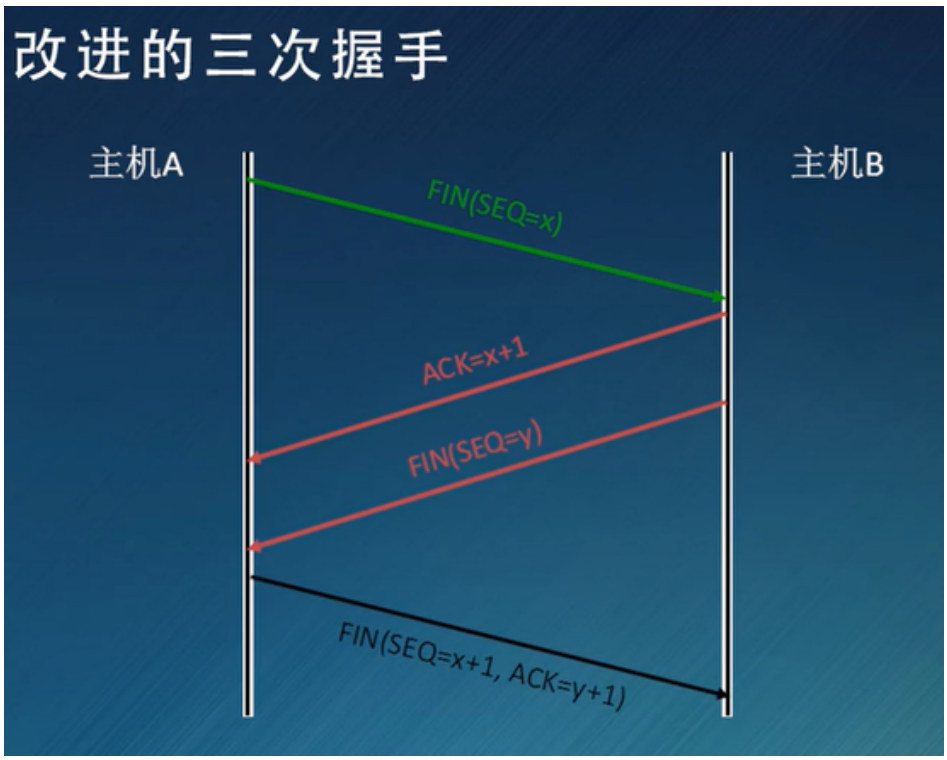
    1. 假设主机A是发起通信请求的客户端，那么它就会向服务器端也就是主机B发送一个请求建立连接的TCP报文，这个TCP报文的SYN=1。假设它的序号是SEQ=x。

2. 这是主机B收到这个请求就要向主机A发送一个应答来表明它已经接收了这个请求。
    - > 当然主机A如果收到了这个应答，就应该知道主机B同意这次连接，可以开始双方通信了。但是前提就向两军问题一样，这个应答报文是有可能丢失的，所以主机B在发出应答报文时，并不知道这个报文是否能够到达主机A。也就是说主机B在发出应答时，并不确定主机A是否能够开始这一次通信，所以在主机B发出应答报文时，就为这次传输预留空间很显然是一件莽撞的决定。
3. 所以为了能够保证主机B发给主机A的应答报文被主机A正确接收，所以主机A还要再向主机B发送一个对于应答报文的应答，这就是第三个报文。
    - > 这个过程称为三次握手。因为在连接过程中需要传输三个TCP报文段分别用于请求、应答、对于应答的应答才能够建立起一次TCP连接。如果这三个报文都正确接受了，这时才说TCP连接建立起来了，才可以开始下此真正的数据通信。

5. 释放连接
    1. 释放TCP连接同样很重要：必须在数据传输结束后由发送方和接收方同时将用于这次通信的资源释放掉，这样这些资源才能被其他进程所继续使用。
    2. 非对称释放
        1. 发送释放请求后单方终止连接
        2. 有可能丢失对方发送的数据
    3. 对称释放
        1. 各自独立发出释放连接请求
        2. 收到对方的释放确认后才可以释放连接
        3. Two-army问题
    4.



1. 当结束了向主机B的数据发送时，显然A希望与主机B断开连接释放本地资源。这是主机A就向主机B发送一个FIN=1的TCP报文段，请求与主机B断开连接。
2. 主机B如果同意与主机A断开连接，这时候主机B就发回一个应答报文。
3. 主机A在收到这个应答报文后，为了告诉主机B它以经收到了这个应答报文，显然它还要对应答报文进行应答。
    - > 这是一种情况，考虑如果主机B不同意与主机A断开连接，这种情况也是经常发生的。主机A的发送缓冲区中已经没有数据了，而主机仍有数据要向主机A发送。因为TCP连接是一个全双工通信，所以前面描述的情况是有可能的。
4.

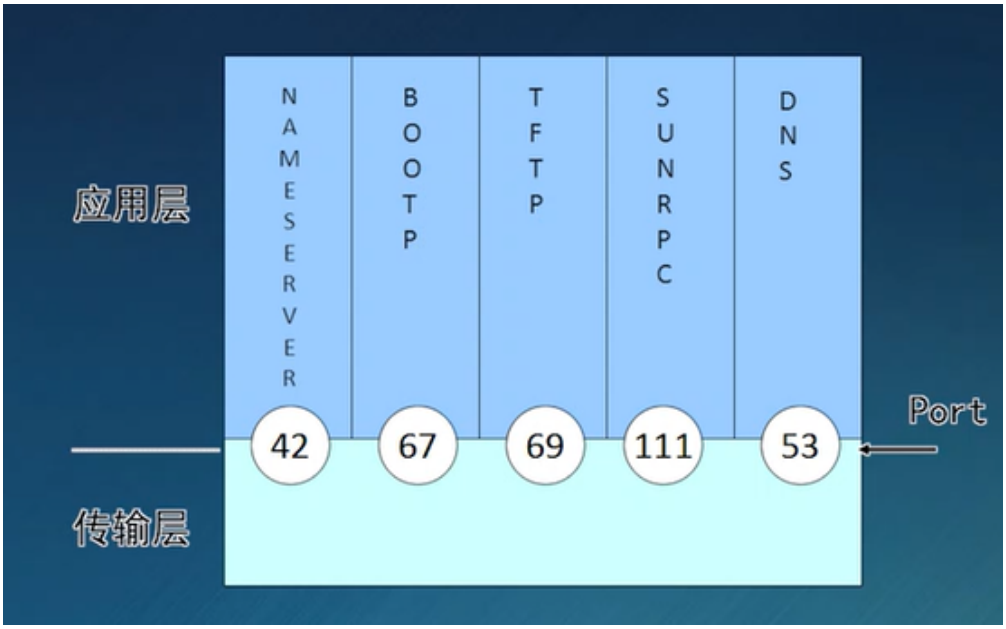


1. 主机A向主机B提出断开连接的请求，发送FIN=1的TCP报文段。
2. 主机B尽管收到了这个报文段，并且给出了回答，但是主机B不同意断开连接，所以这个应答报文段的FIN=0。
3. 在FIN=0的应答报文段发出后，主机B仍然可以向主机A发送TCP报文段。只有主机B的发送缓冲区也清空的情况下，也就是主机B要发送的所有报文段都已经发送完毕时，主机B才向主机A再发送一个FIN=1的TCP报文段进行再次的确认。
4. 这时主机A才能确认主机B也没有数据向主机A发送了，也就是说主机B同意断开连接。这时，主机A再对这个FIN=1的应答报文进行再次确认，这是双方的连接才最终断开。

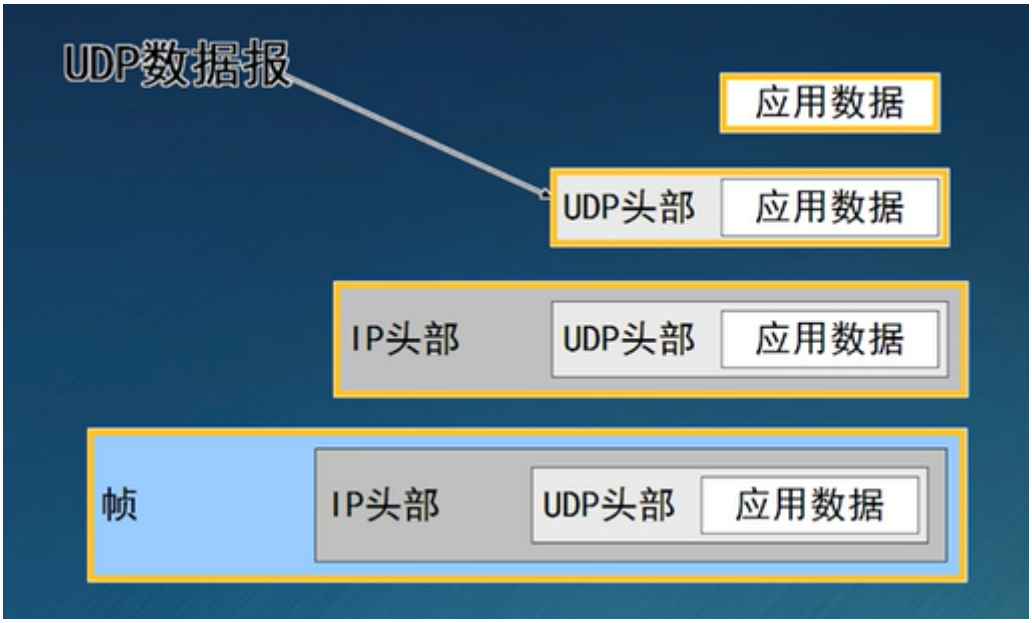
三、无连接的UDP协议

1. UDP协议提供的是无连接的数据报服务----对于UDP的用户来说，它交给UDP的数据，UDP协议将在没有与对方建立连接的情况下发送出去，并且也不提供序号服务和流量控制功能。
2. UDP协议概述
    1. 为应用进程提供无连接的数据传输服务
    2. 使用于传输实时数据
    3. 主要特点：节省了建立和释放连接和重传的开销

4. 由应用层解决纠错和丢失问题
5. 由RFC768定义

3. UDP的端点标识（Port）：在UDP中仍然完成了端口寻址问题
    1. 用16bit区分2^{16}个端口---区分2^{16}个不同的应用进程
    2. 发送端
        - 分配的16bit端口是由操作系统随机分配的
        - 分配源端口，指定宿端口，构造UDP数据报，交IP
        - 节省建立和释放连接和重传的开销
    3. 接收端
        - 而连接的另外一端也就是接收端，如果是服务器得话，服务器的端口仍然是固定端口
        - 匹配UDP头部宿端口的应用进程
        - 匹配成功，数据报排入相应的队列，若端口队列满，则丢弃数据报
        - 匹配不成功，丢弃数据包，回送“宿端口不可达”的ICMP报文
    4. UDP保留端口举例
        - 

4. 由于没有建立连接、序号确认、流量控制等功能，UDP报文格式简化很多，运行起来也简单很多。
5. UDP数据报的封装

- 

6. UDP首部

- 

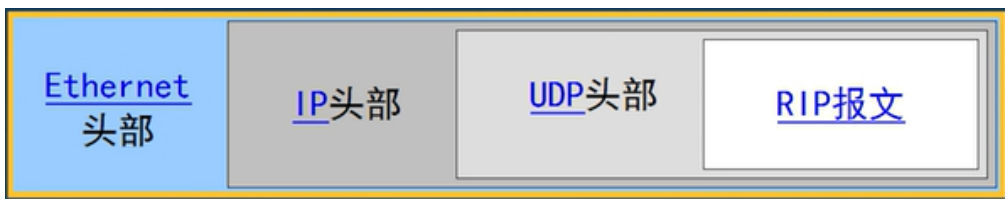
1. 源端口号：16bit，用于标识源进程
2. 目的端口号：16bit，用于标识UDP报文段的目的进程
3. UDP报文长度：16bit（最小为8）
4. UDP校验和：16bit，校验和运算方法同IP分组头部以及TCP头部中的校验和运算方法相同。**注意**：由于UDP是一种乐观的协议，当UDP校验和全0时，表示不适用校验和校验。全1表示校验和0.

7. UDP伪首部

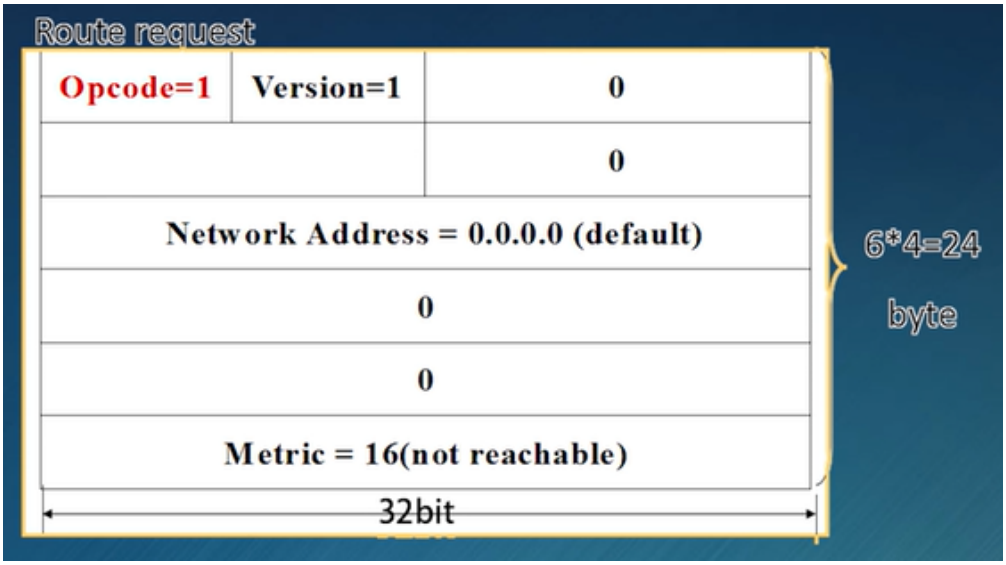
- 

1. 伪头部12字节，伪头部只参与校验和的运算，不被真正的发送出去。因为接收端也可以从接收端的头部组成一个伪首部。这是因为UDP头部过短所造成的校验和容易产生差错的情况而故意增加的用于运算的一个伪首部。
2. 4字节源IP地址，4字节宿IP地址，8bit0，8bit协议字段值固定17，16bitUDP长度。

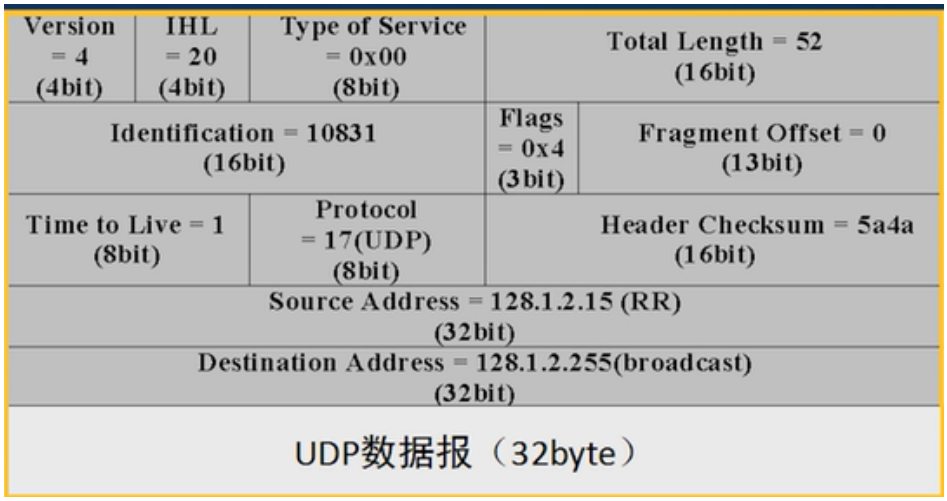
8. UDP传输某种报文的过程-----传输RIP报文
   - 

   1. 路由请求
      - 

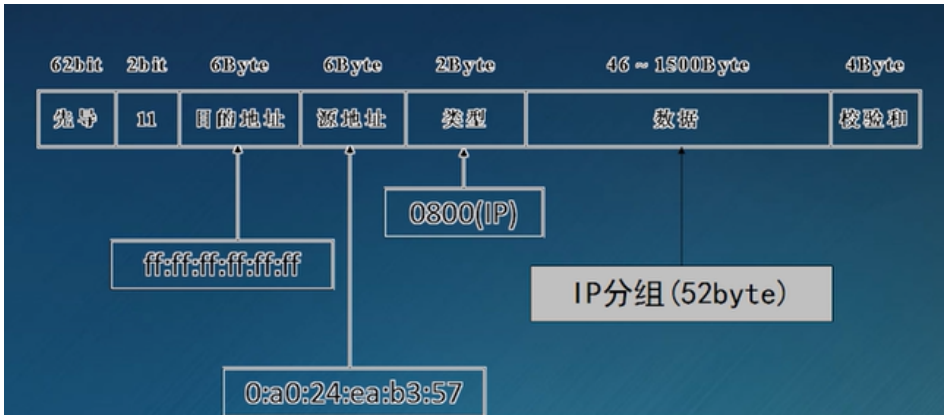
   2. RIP：请求路由信息

- 

- 假设RIP报文的数据格式如上图，即24字节的RIP协议报文。里面是一个请求，是发给邻居路由器请求邻居路由器将自己的路由信息填在这个报文当中的空白位置的一个RIP请求路由信息。

1. UDP分配端口：
   - 
   - 首先将这24个字节作为UDP的数据部分，加上UDP的8字节头部。由于RIP协议固定的目的端口号是520，源端口号也是520，所以源端口和目的端口被填上了520。数据部分24字节加上头部8字节，所以长度32字节。校验和就是按照Check sum的方法计算的校验和。
2. IP：赋予IP地址
   - 

   - 上述的UDP报文向下交给IP协议，再由IP协议封装成IP报文，那么这个IP报文的数据部分就是刚才封装好的UDP的32字节的数据报。而IP的头部是固定的20字节的头部，相应的值也填进去。
3. Ethernet：赋予以太网地址

- 

- 再将IP报文封装在物理网络数据帧中，以太网帧的数据部分就是上述封装好了的IP分组。IP头部20字节，IP数据部分是UDP报文32字节，所以总体长度52字节。还有就是数据链路层的数据帧的头部当中的各个字节把它填充好。由于这个RIP协议需要广播发送，所以目的地址填全1。这样一个以太网的数据帧可以在以太网中进行传输了。任何接受到这个数据帧的主机由于目的地址是全1，就会将它接受到自己的缓冲区当中，然后逐层解析，最终拿到RIP报文。

八、应用层

一、电子邮件Email

1. Internet Email的工作模型
   - 

   - 首先Email的服务提供者一定要向用户提供一个用户接口。假设用户现在要发送一封邮件，它要先在用户接口中输入必要的信息，然后将发送的邮件放在发出邮件缓冲区，由客户机进行后台发送。这时客户机要做的工作就是立即与原邮件服务器建立TCP连接，再由原邮件服务器与对方的接收邮件的服务器建立TCP连接，最终将这个邮件传送给目的端的邮件服务器，这是发送一封邮件的过程。

    - 在接收一封邮件时，我们要在自己的邮件服务器当中提供用户名和密码，也是通过用户接口的方式。然后到自己的邮箱当中按照一定的规则取出自己想要的邮件，这是一封邮件在传输过程中的工作模型。
    - 上述存在的问题：传输过程中各个协议如何协调工作的、email格式是什么样子的
2. Email格式
    1. Internet Email的格式在RFC822中定义，由两部分组成：header、body
    2. header：
        - RFC 822 header fields related to message transportation

Header	Meaning
To:	Email address(es) of primary recipient(s)
Cc:	Email address(es) of secondary recipient(s)
Bcc:	Email address(es) for blind carbon copies
From:	Person or people who created the message
Received	Email address of actual sender
Return-Path:	Can be used to identify a path back to the sender

        - Some fields used in the RFC 822 message header

Header	Meaning
Date:	The date and time the message was sent
Reply to:	Email address to which replies should be sent
Message id:	Unique number for referencing this message later
In-Reply-To:	Message-id of the message to which this is a reply
Keywords:	User chosen keywords
Subject:	Short summary of the message for the one-line display

    3. Body：内容是任意的，但必须是7位标准ASCII字符集。
    4. 扩展Email的格式：扩展Email称为MIME（Multipurpose Internet Mail Extensions），最初在RFC1314中定义的，修订版发布在RFC1521中。主要是Email中存在诸多不足和限制而提出的。

5. RFC822Email的缺点
    1. 不能传送可执行文件或其他二进制对象。

        > 人们试图将二进制文件转换成SMTP使用的ASCII文本，例如流行的UNIX UUencode/UUdecode 方案，但这些均未形成“正式标准或事实上的标准”。

    2. 限于传送7位的ASCII字符。

        > 许多非英语国家的文字（如中文、俄文、甚至带重音符号的法文或德文）就无法传送。即使在SMTP转换成ASCII码时也会遇到一些麻烦。

    3. 服务器会拒绝超过一定长度的邮件。

        > RFC822对于超长的邮件是拒绝发送的，所以MIME中对这个问题也进行了修复。

    4. 某些实现并没有完全按照的标准。

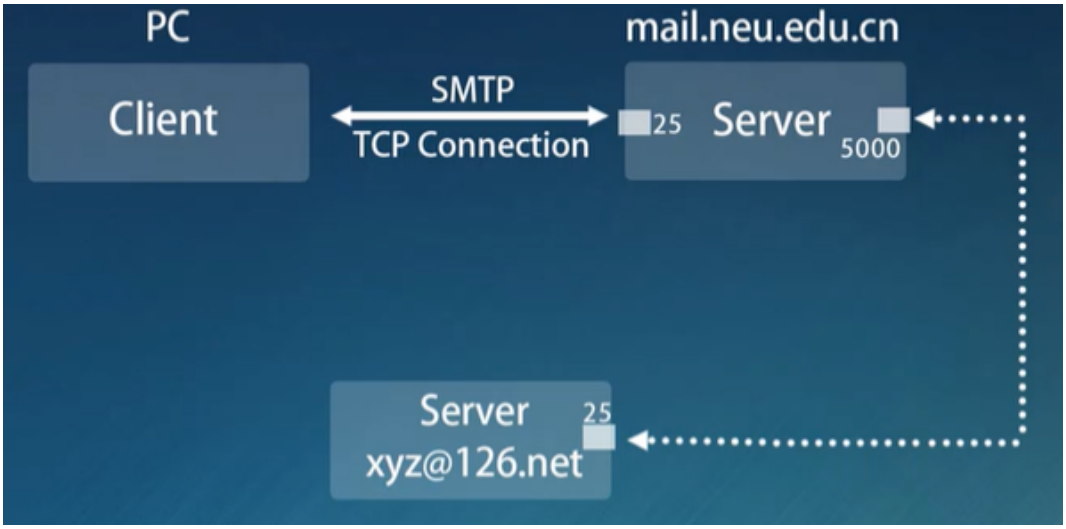
        > 常见问题如下：
        >
        > - 回车、换行的删除和增加
        > - 超过76个字符时的处理：截断或自动换行
        > - 后面多余空格的删除
        > - 将制表符转换为多个空格

6. MIME的处理方法：MIME利用了RFC822的格式，但又加入了一个扩展的头部（实际上位于RFC822的header中）。主要是为了克服RFC822只能传输标准7位ASCII文本的缺点，使Email既可以传输标准ASCII码文本，也可以传输像图片、声音、各种语言文本等。

3. 邮件的传送是一种客户服务器的交互模式，为了保证邮件传输的可靠性，邮件传输协议都是基于TCP协议的。
    - 

- 

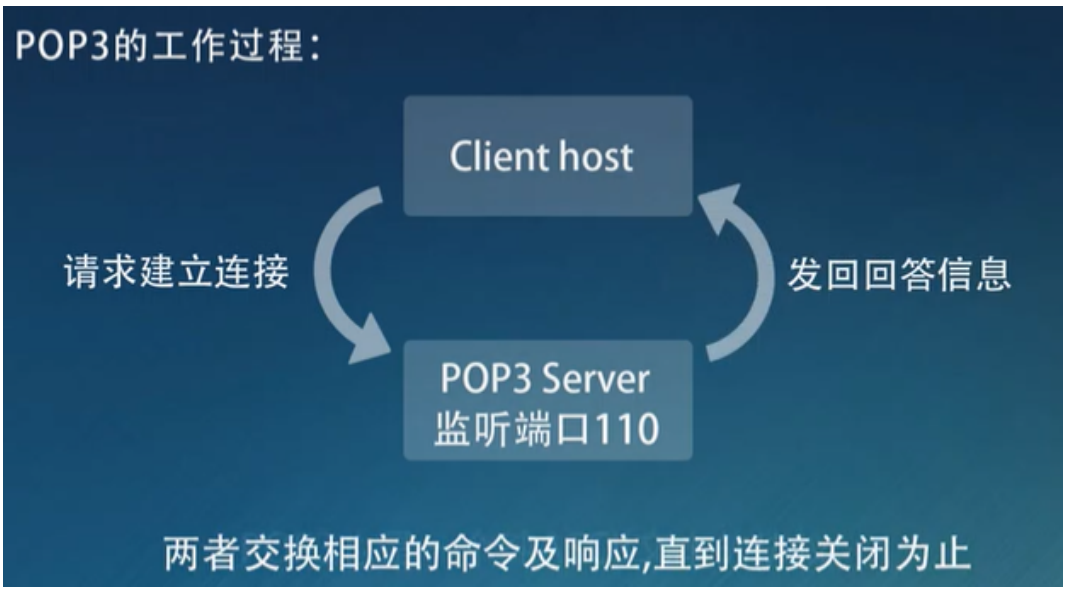
1. 上图中分为三个阶段，前两个阶段有一个共同点---接收方都是服务器，这就使得在传输邮件时，可以以比较简单的方式去链接服务器上WLAN的端口来传输一封Email。这种传输邮件的协议称为SMTP协议（Simple Mail Transfer Protocol，Defined in RFC 821）。
2. SMTP协议的服务器是在本机的25号端口上进行监听，当有连接请求进来时，接受链接，然后传输邮件，传输完毕后再将连接断开。
3. SMTP是一个纯ASCII字符的协议，所有命令和数据均以标准ASCII字符传输。

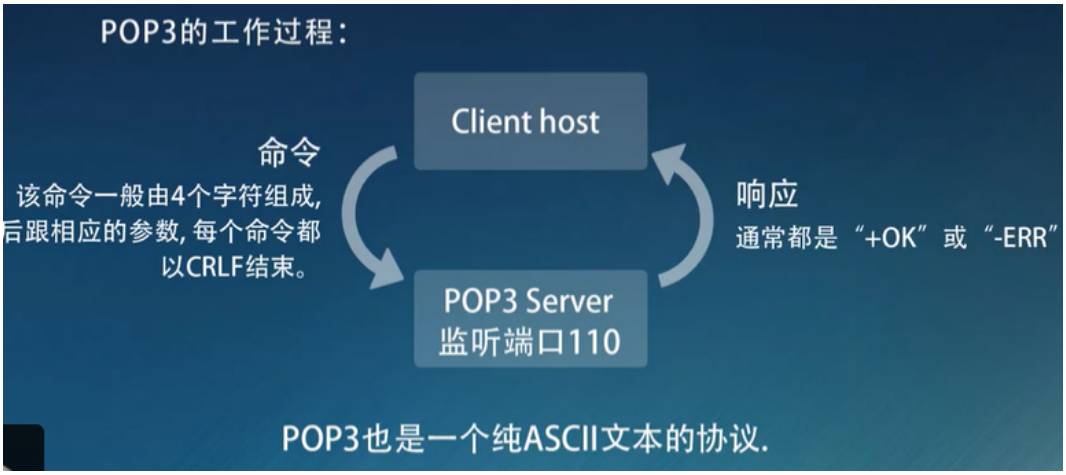
4. 邮件传输例子
   - 

   - 假设某个用户---mail.neu.edu.cn这个邮件服务器的用户，现在他想发送一封邮件给xyz@126.net这样的用户。首先用户在自己的客户机上打开一个进程，这个进程是由操作系统随机分配的进程号。那么与本地的服务器的25号端口建立连接，将这样的Email按照上文822数据格式组织好发送给本地的Email服务器的25号端口。本地的Email服务器在收到这个Email后就会再随机打开本地的一个端口号，假设是5000，然后去链接目的邮件服务器，也就是126.net的25号端口，将这样的邮件发送给126.net的25号端口，然后这个

邮件服务器收到了这个邮件之后就会保存在本地的邮箱当中，等待接收端即xyz这个用户来取邮件。

5. 取邮件
    - 

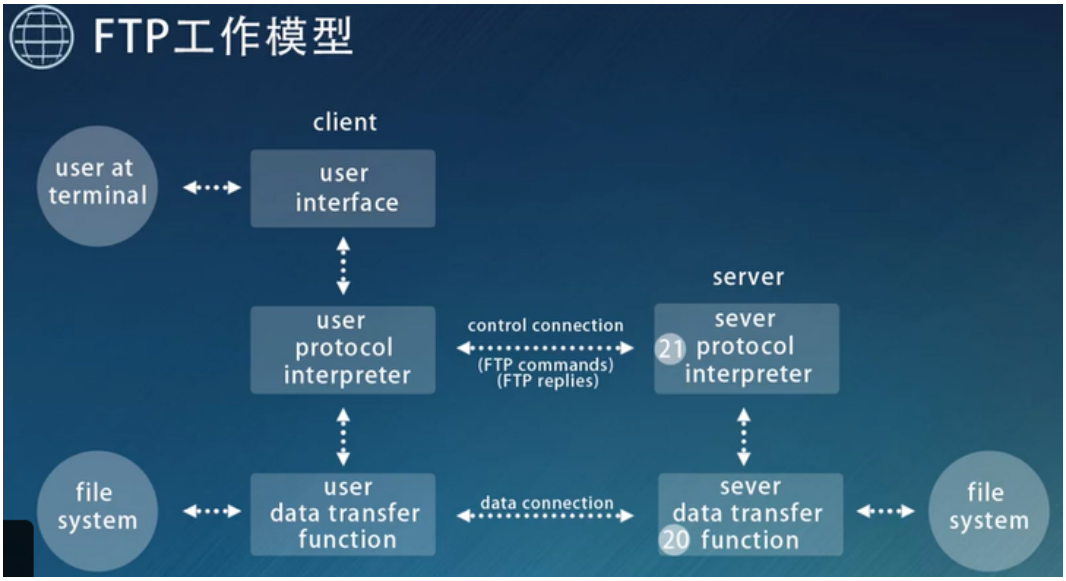
    - 大多数用户是在个人计算机上来收邮件的，而个人计算机不能长时间连网，即使连接在网上也不一定有固定的域名或ip地址，因此不能作为smtp服务器。这一点是个人计算机和服务器的本质区别。因此如果需要在Internet上面来接收邮件，那么用户就一定要将自己的PC机主动的与Email服务器建立连接，提供自己的用户名、密码、以及接收邮件的请求，这时候才能接收邮件，显然这个接收邮件的协议与SMTP协议不同。这个协议就是POP3（第三个版本）协议----邮局协议。
    - 

- 

    1. POP3工作过程：POP服务器监听110号端口，等待用户连接请求。用户发出链接请求后，POP3服务器接受请求并与之建立TCP连接。客户端在建立连接后就要提供用户名、密码、以及它需要取出的邮件的序号等等。POP3服务器更具用户民、密码确定邮箱入口地址，同时将用户要求的邮件返回给用户。
6. 总结：发信人的用户代理向源邮件服务器发送邮件，源邮件服务器向目的邮件服务器发送邮件使用的都是SMTP协议；POP3等协议只是从目的邮件服务器上读取邮件。

二、文件传输协议FTP

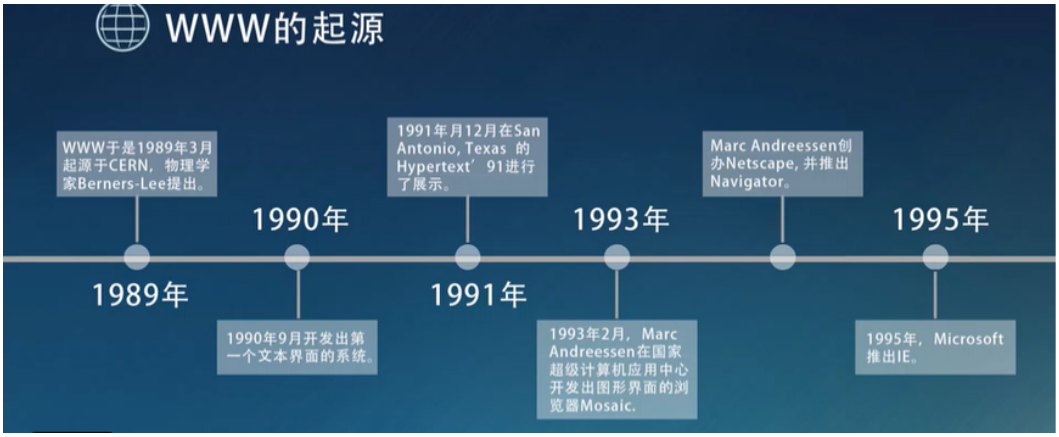
1. File Transfer Protocol，一种文件传输协议，支持异构的计算机进行文件传输，支持ASCII文件和纯二进制文件的传输，是在RFC的959中定义的。
2. 例子：QQ这种即时通讯软件中，常用离线文件传输的功能。你的朋友同你分隔两地并且需要向你发送一个600M的文件，如何实现?
3. FTP协议的宗旨是允许我们在Internet的范围内上传和下载文件，这就需要有一个文件服务器，它有一个大大的存储空间，并且打开固定端口，等待用户的上传和下载的需求，同时FTP服务还应该包含一些用户名的验证，这样才能防止非法上传和下载。
4. FTP的工作模式

- 

  - 客户端首先要有一个用户接口，在用户接口用户提出上传某个文件或是下载某个文件的需求，然后由用户的协议的解析器来解析这些命令是上传还是下载，文件的位置具体在哪，这是文件的解析器就能够理解用户的需求了。然后按照用户的需求与文件服务器建立连接，而文件服务器这时一定要打开的是21号端口等待用户的连接请求。同时FTP协议还有另外一个固定端口就是20号端口。21号端口用于命令传输，20号端口用于数据的传输。在20号端口即server端，接受了用户的请求和命令后，就将命令进行解析，然后来到本地的文件系统找到用户要上传或者下载的文件的位置，然后将文件交到20号端口，再由20号端口与对方建立连接，将数据下载给客户端。客户端收到下载数据后，再将它存放在本地的文件系统中。这就完成了一次数据的下载，数据上传基本一样。

三、WWW应用

1. WWW起源

- 

2. WWW系统必须解决的问题

- 

3. WWW的工作模式
   - 

     1. WWW的工作模型中，首先要有一个WWW的服务器，它是等待用户的连接请求的。用户想要访问一个WWW服务器，那么就要连接它的80端口与他建立TCP连接，然后将自己要访问的WWW文档的地址发给这个服务器，这个服务器再到自己的WWW文档库中找到相应的文档返回给客户端，在这个过程中就要解决刚才的3个问题。
4. 如何标识一个WWW文档：标识的方法就是----URL（Uniform Resource Locator---统一资源定位符）的缩写，用来解决上述第一个问题。
   - URL是能对从Internet上得到的资源的位置和访问方法的一种简洁的表示。
   - 

- 格式：`<访问方式>://<主机>:<端口>/<路径>`。
- 访问WWW服务器是http的方式，由于统一资源定位符还允许我们采用其他方法访问其他类型的主机的其他端口，所以实际上的访问方法很多。

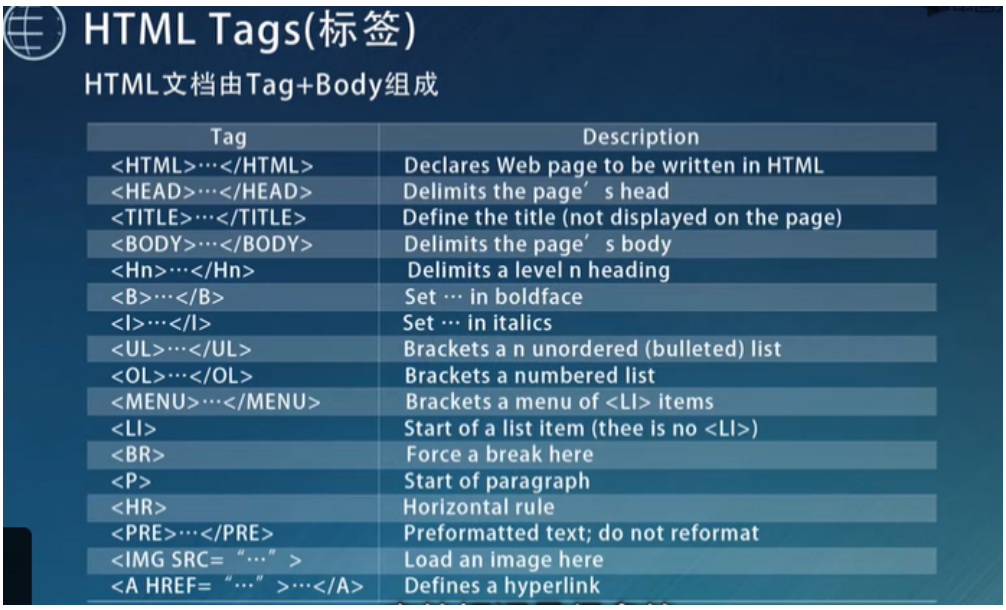
1. URL访问方式

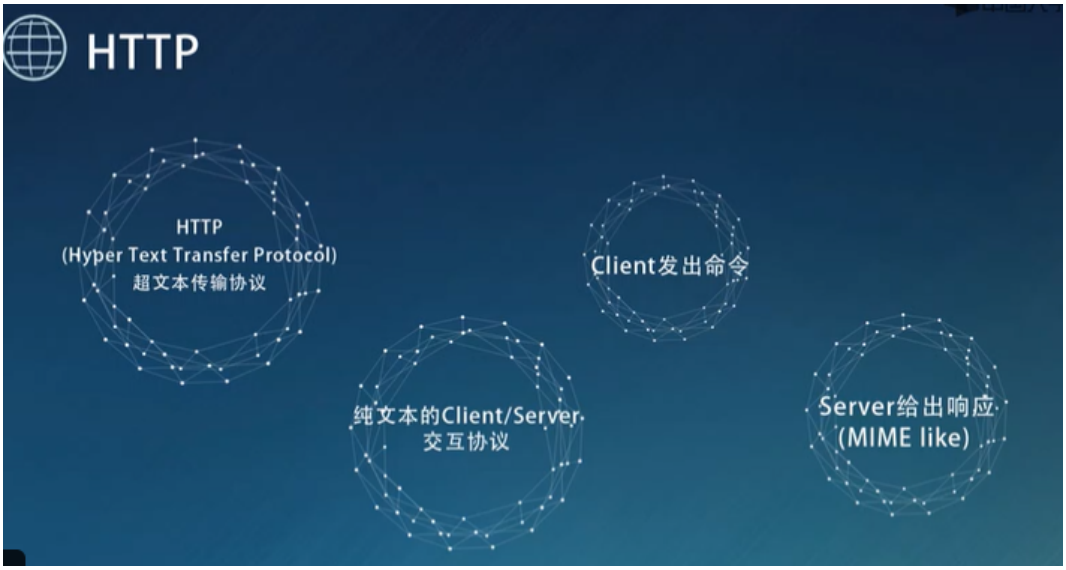
Name	Used for	Example
http	HTML	http://www.cs.uv.nl/~ast
ftp	FTP	ftp://ftp.neu.edu.cn/incoming/readme
file	Local file	d:/data/x1.txt
news	News group	news:comps.os.minix
news	News article	news:AA0134223211@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/libraries
mailto	Sending email	mailto:gaofx@necmail.neu.edu.cn
telnet	Remote login	telnet://bbs.neu.edu.cn

   - 访问www服务器时并没有给出端口号，这是因为访问所使用的用户接口将这个端口号进行了默认设置，默认80。但如果一个www服务器它使用的不是80端口等待用户等的请求，有些是8080或是3128，这时候我们就一定要在浏览器中用冒号8080或冒号3128的形式指明端口号

5. 如何标识一个www文档使得再异构计算机上通过浏览器或是翻译器使它能展现相同的风格：方法是采用HTML语言---超文本的标记语言

1. HTML Tags

- 

6. 客户和服务器之间传输www文档使用的协议：http超文本传输协议
   - 

   - 基于Client & Server模式的交互协议，也是一个纯文本协议。是由客户端发送请求，Server端相应这个请求按照客户的要求在将相应的www文档返回给客户的这样一种协议，基于TCP连接的。

四、DNS应用

1. Internet上每台主机的唯一标识是它的IP地址，当要访问某台主机时，必须要知道它的IP地址才行。实际上，我们更多的是通过主机的名字而不是地址来访问的，这是因为在Internet当中提供了一种可以将主机的名字和IP地址互相转换的应用层协议，这就是DNS协议。
2. DNS概述
   1. DNS应用也叫域名字系统的应用，主要任务是将一个域名翻译成对应的IP地址。

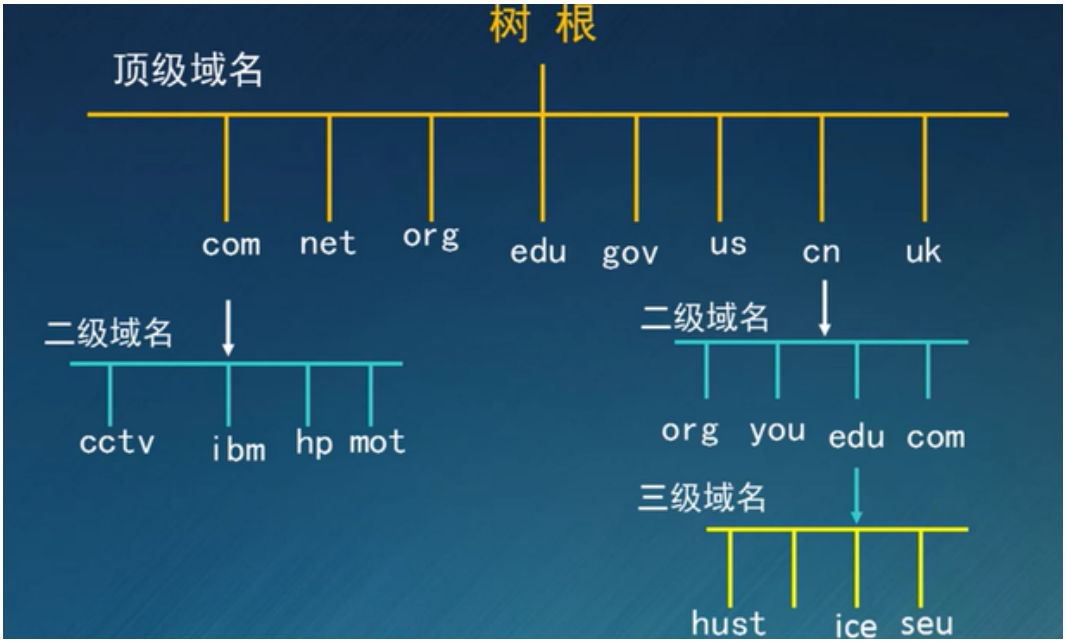
- 如何实现这种翻译呢？简单想法是：设计一个大大的数据库，里面存放所有的域名和IP地址的对应关系。如果某个客户想要知道这一个对应关系，那么就发来传输请求来进行查询。这种方式在Internet中不可行，原因是：第一---没办法保证数据库的完整性，数据库的更新非常频繁；第二---这个数据库要接受来自全Internet的访问要求，这是非常大的负担。

2. Internet的域名系统：DNS被设计成为一个联机分布数据库系统、DNS采用客户服务器模式、DNS由若干台域名服务器组成（1.太多域名的转换都是在本地映射；2.少数域名在Internet网上通信映射）。

3. Internet域名结构：采用层次树状结构的名命方法

    1. 说明：

        1. 域名的结构由若干个分量组成，各分量之间用点隔开；如`.qcxy.hb.cn`：最右边的是顶级域名，从右往左域名等级依次下降。
        2. 每一级的域名都由英文字母和数据组成（<63个，不分大小写）
        3. 级别最低的域名在最左边，最高的在最右边。（整个域名长度<255）

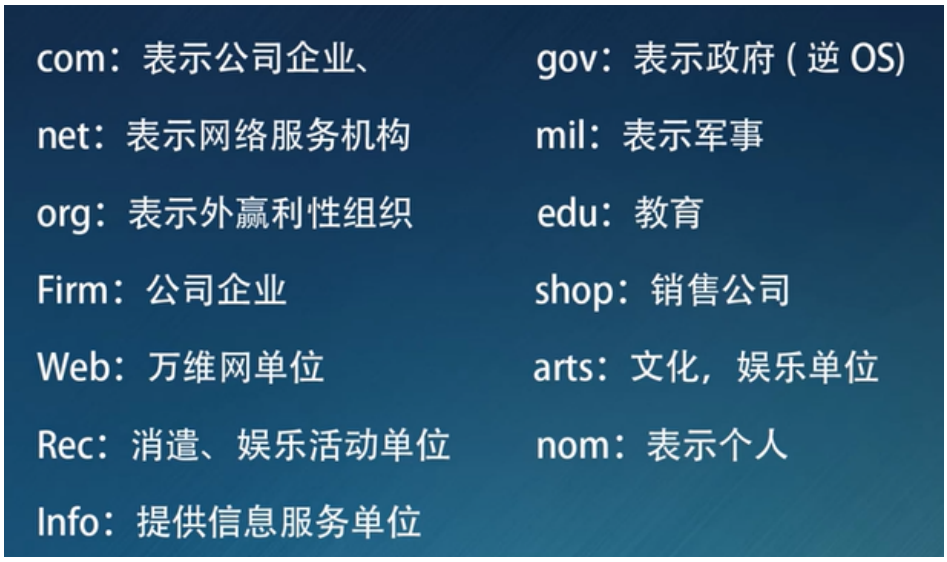
    2. 

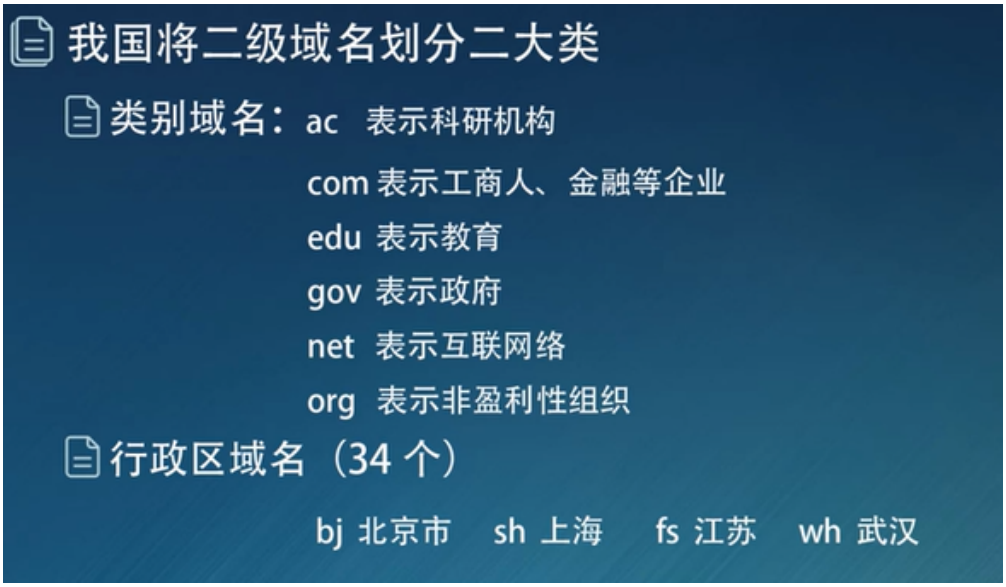
        - 域名只是逻辑概念，并不反映出计算机所在的物理地点/网络信息。

    3. 顶级域名由Internet的有关机构ICANN管理决定。

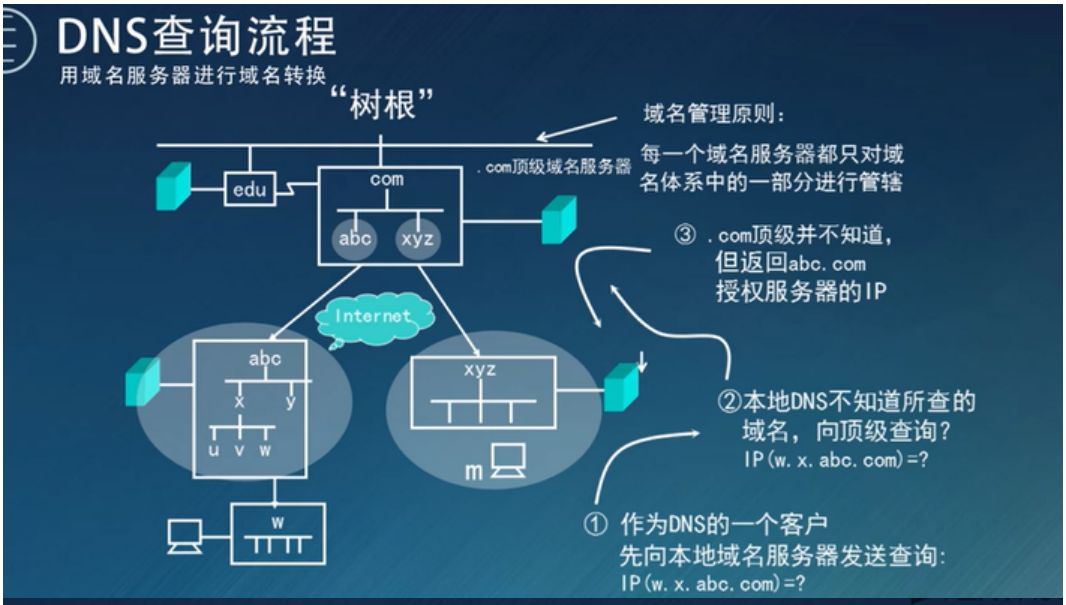
    4. 目前顶级域名分为3类：

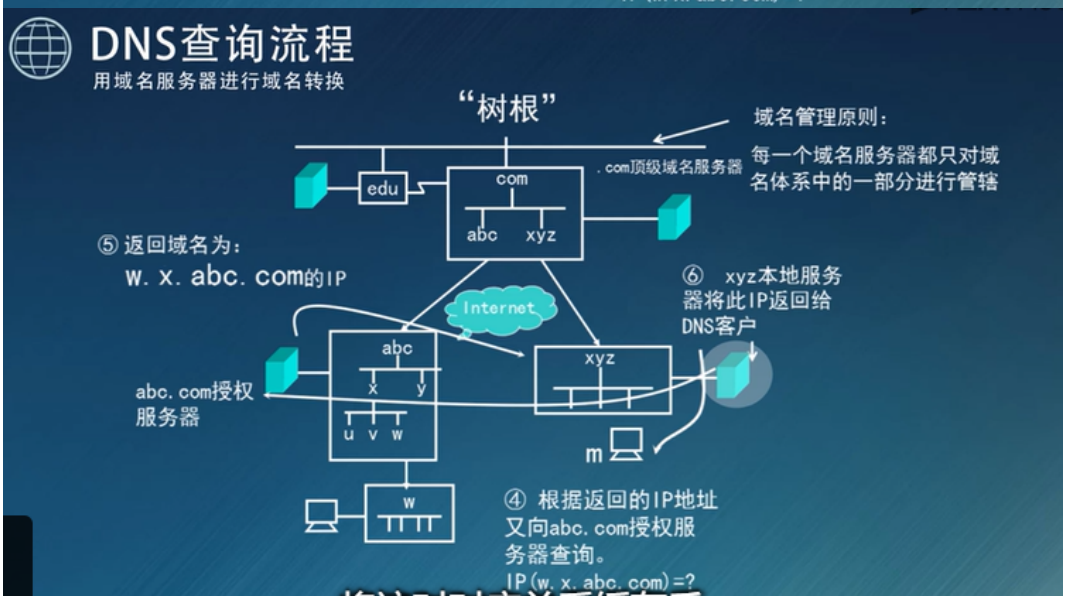
        1. 国家顶级域名：采用ISO3166规定。cn：中国、us：美国、uk：英国、jp：日本、sg：新加披

2. 国际顶级域名：采用int国际性组织可在int下注册
3. 通用顶级域名：根据[RFC1591]规定（最早6个+新增7个=13个）
    - 

5. 二级域名由国家自行确定
    - 

6. 三级域名的申请注册
    1. 我国三级域名的管理和申请（除edu网外），应向中国互联网信息中心CNNIC申请。
    2. 域名有相对域名和绝对域名之分
        1. 相对域名：至某一级的域名的下属域名，如：nju是edu下属的一个相对域名。
        2. 绝对域名：是一个完整的域名，一直写出根域名（又称：完全合格的域名.FQDN）如：nju.edu.cn
4. DNS查询流程

- 

- 

- xyz、abc都是小范围的DNS服务器---本地域名服务器，在其之上有顶级/二级域名服务器（com）。
- 在主机M上，它作为DNS服务的一个客户，想向本地的域名服务器询问`w.x.abc.com`的IP地址是什么，本地的域名服务器也就是`xyz.com`这个域当中的域名服务器也许只保存了本地的域名与IP地址之间的关系，那么这时它并不知道`abc.com`这个域名下的其他主机的IP地址。这时候他就必须将这个请求回溯到比它更高一级的顶级域名服务器去。`.com`这个顶级域名服务器当中是否一定保存了`abc.com`和`xyz.com`下的所有主机域名和IP的对应关系的呢？显然这时不一定的。这时候`.com`的顶级域名服务器并不知道`w.x.abc.com`的域名所对应的IP地址。但它却知道`abc.com`的域名服务器的地址，所以这时候顶级域名服务器就会将`abc.com`的域名服务器IP地址发送给本地DNS服务器。这时本地的DNS服务器就可以根据从顶级域名服务器上获得的`abc.com`的服务器的地址去向`abc.com`的服务器查询`w.x.abc.com`的IP地址，结果一定可以查询

到，因为`abc.com`的服务器当中就应该包含有它所在域的所有主机的域名域IP的对应关系。那么`abc.com`的域名服务器当中一定存有`w.x.abc.com`的域名和IP地址的对应关系。并且将IP地址返回给本地的DNS服务器。本地的DNS服务器将这个对应关系缓存后将结果返回给主机M。这样就完成了一次DNS域名到IP的查询。**同时DNS也提供了IP到域名字转换**。个人主机在使用DNS服务时，就要在本机配置一个默认的DNS服务器。有时这个服务器的IP地址是由手工配置的，也有自动获取的，有时是由ISP来提供的。