

计算机网络

1 概述

1.1 因特网概述

- 网络、互联网、因特网
 - 网络 (Network) 由若干结点 (Node) 和连接这些结点的链路 (Link) 构成
 - 多个网络可以通过路由器互连起来，构成覆盖范围更大的网络，即互联网 (internet)。互联网是网络的网络
 - 因特网 (Internet) 是世界上最大的互连网络
- 基于ISP (因特网服务提供商) 的三层结构的因特网
 - 第一层：国际性区域
 - 第二层：区域性或国家性覆盖规模
 - 第三层：本地范围
- 因特网的组成
 - 边缘部分
 - 由所有连接在因特网上的主机组成。这部分是用户直接使用的，用来进行通信和资源共享
 - 核心部分
 - 由大量网络和连接这些网络的路由器组成，为边缘部分提供服务（提供连通性和交换）

1.2 交换方式

- 电路交换 (Circuit Switching)
 - 定义：电话交换机接通电话线的方式
 - 从通信资源分配的角度来看，交换 (Switching) 就是按照某种方式动态地分配传输线路的资源
 - 步骤：
 1. 建立连接 (分配通信资源)
 2. 通话 (一直占用通信资源)
 3. 释放连接 (归还通信资源)
- 分组交换 (Packet Switching)
 - 发送方：构造分组、发送分组
 - 路由器：缓存分组、转发分组
 - 接收方：接收分组、还原报文
- 报文交换 (Message Switching)
 - 主要用于早期的电报通信网

1.3 计算机网络的定义和分类

- 计算机网络的定义
 - 一些互连的、自治的计算机的集合
 - 互连：计算机之间通过有线或无线的方式进行数据通信
 - 自治：独立的计算机，有自己的软硬件，可以单独运行
 - 集合：至少两台
- 计算机网络的分类
 - 按交换技术分类
 - 电路交换网络
 - 报文交换网络
 - 分组交换网络
 - 按使用者分类
 - 公用网
 - 专用网
 - 按传输介质分类
 - 有线网络
 - 无线网络
 - 按覆盖范围分类
 - 广域网WAN
 - 国家、州、
 - 城域网MAN
 - 城市
 - 局域网LAN
 - 机构、企业、校园
 - 个域网PAN
 - 个人区域网络
 - 按拓扑结构分类
 - 总线型网络
 - 星型网络
 - 环型网络
 - 网状型网络

1.4 计算机网络的性能指标

- 速率
 - 连接在计算机网络上的主机在数字信道上的传送比特的速率，也成为比特率或数据率
 - 单位：bps (bit/s)

1 kbps = 1000 bps
- 带宽
 - 在模拟信号中的意义

- 信号所包含的不同频率成分所占据的频率范围
 - 单位: Hz
- 在计算机网络中的意义
 - 网络的通信线路所能传送数据的能力，表示在单位时间内从网络某一点到另一点所能通过的“最高数据率”
 - 单位: bps
- 吞吐量
 - 在单位时间内通过某个网络（信道、接口）的数据量
 - 受网络的带宽或额定速度的限制
- 时延
 - 发送时延
 - 分组长度 / 发送速率
 - 传播时延
 - 信道长度/ 电磁波传播速率

电磁波传播速率

 - 自由空间: 3e8m/s
 - 铜线: 2.3e8m/s
 - 光纤: 2e8m/s
 - 处理时延
 - 不方便计算
 - 可能包括排队时延
 - 一般忽略不计
- 时延带宽积
 - 传播时延 × 带宽
 - 若发送端连续发送数据，则在发送的第一个比特到达终点时，发送端就已经发送了时延带宽积个比特
 - 链路的时延带宽积又称为以比特为单位的链路长度
- 往返时间RTT
 - 双向交互一次所需时间
- 利用率
 - 两种
 - 信道利用率：表示某信道有百分之几的时间是被利用的（有数据通过）
 - 网络利用率：全网络的信道利用率的加权平均
 - 根据排队论，当某信道的利用率增大时，该信道时延也会增加
 - 如果D₀表示网络空闲时延，D表示网络当前时延，则在适当条件下，
$$D = \frac{D_0}{1 - U}$$
- 信道利用率并非越高越好
- 丢包率
 - 即分组丢失率，是指在一定时间内，传输过程中丢失的分组数量与总分组数量的比率
 - 分为接口丢包率、结点丢包率、链路丢包率、路径丢包率、网络丢包率等
 - 分组丢失两种情况

- 误码
- 结点交换机队列已满

1.5 计算机网络体系结构

- 常见的计算机网络体系结构
 - OSI体系结构
 - 应用层
 - 表示层
 - 会话层
 - 运输层
 - 网络层
 - 数据链路层
 - 物理层
 - TCP/IP体系结构
 - 应用层
 - 运输层
 - 网际层
 - 网络接口层
- | 路由器一般只包含网际层和网络接口层
- 原理体系结构
 - 应用层
 - 传输层
 - 网络层
 - 数据链路层
 - 物理层
- 分层
 - 分层设计理念：化大为小
 - 各层问题
 - 应用层：通过应用进程的交互来实现特定网络应用
 - 运输层：进程间基于网络通信
 - 网络层：分组在多个网络上传输
 - 数据链路层：分组在一个网络上传输
 - 物理层：使用信号传输比特
- 专用术语
 - 实体：任何可发送或接收信息的软硬件进程
 - 对等实体：收发双方相同层次中的实体
 - 协议：控制两个对等实体进行逻辑通信的规则的集合
 - 语法：定义所交换信息的格式（字段等）
 - 语义：定义收发双方所要完成的操作
 - 同步：定义收发双方的时序关系
 - 服务：在协议的控制下，两个对等实体的逻辑通信使得本层能够向上一层提供服务；要实现本层协议，还需要使用下面一层提供的服务
 - 协议是水平的，服务是垂直的

- 下面的协议对上层的实体透明
 - 服务访问点：在同一个系统中相邻两层实体交换信息的逻辑接口，用于区分不同的服务类型
 - 数据链路层-帧的“类型”字段
 - 网络层-IP数据报首部中的“协议字段”
 - 运输层-“端口号”
 - 服务原语：上层使用下层提供的服务必须通过与下一层交换一些命令
 - 协议数据单元PDU：对等层次之间传送的数据包称为协议数据单元
 - 应用层：报文*message*
 - 运输层：TCP报文段*segment* / UDP用户数据段*datagram*
 - 网络层：分组*packet* (IP数据报)
 - 数据链路层：帧*frame*
 - 物理层：比特流*bit stream*
 - 服务数据单元SDU：同一系统内，层与层之间交换的数据包
- | 多个SDU可以合成一个PDU，一个SDU可以划分为多个PDU

2 物理层

2.1 物理层概述

- 考虑如何才能在连接各种计算机的传输媒体上传输数据比特流
- 为数据链路层屏蔽了各种传输媒体的差异
- 接口规范
 - 机械特性
 - 电气特性
 - 功能特性
 - 过程特性

2.2 物理层下的传输媒体

- 导引型传输媒体
 - 同轴电缆
 - 基带同轴电缆
 - 宽带同轴电缆
 - 双绞线
 - 绞合：抵御部分来自外界的电磁波干扰，减少相邻导线的电磁干扰
 - 光纤
 - 多模光纤
 - 存在许多条不同角度的光线在纤芯中不断地全反射
 - 由于色散，光在多模光纤传输一定距离后必然产生信号失真（脉冲展宽）
 - 只适合近距离传输LIFI
 - 发送光源：发光二极管；接受检测：光电二极管

- 单模光纤
 - 光在纤芯中一直向前传输不产生全反射
 - 没有色散
 - 适合长距离传输且衰减小
 - 发送光源：激光发生器；接收检测：激光检波器
- 电力线
- 非导引型传输媒体
 - 无线电波
 - 广播
 - 微波
 - 地面微波接力通信
 - 卫星通信
 - 红外线
 - 点对点无线传输
 - 直线传输，不能有障碍物，传输距离短
 - 可见光
 - LiFi

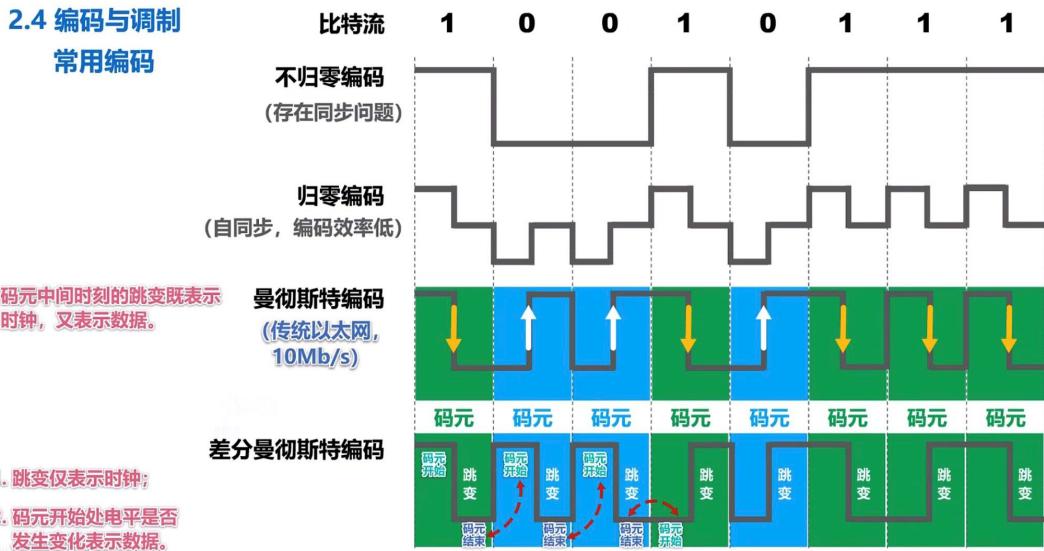
2.3 传输方式

- 串行&并行
 - 串行
 - 并行
- 同步&异步
 - 同步
 - 外同步：在收发双方之间添加一条单独的时钟信号线
 - 内同步：发送端将时钟同步信号编码到发送数据中一起传输
 - 异步
 - 字节之间异步，字节之间的时间间隔不固定
 - 字节中的每个比特同步（各比特的持续时间相同）
- 单向&双向
 - 单向通信（单工）
 - 双向交替通信（半双工）
 - 双向同时通信（全双工）

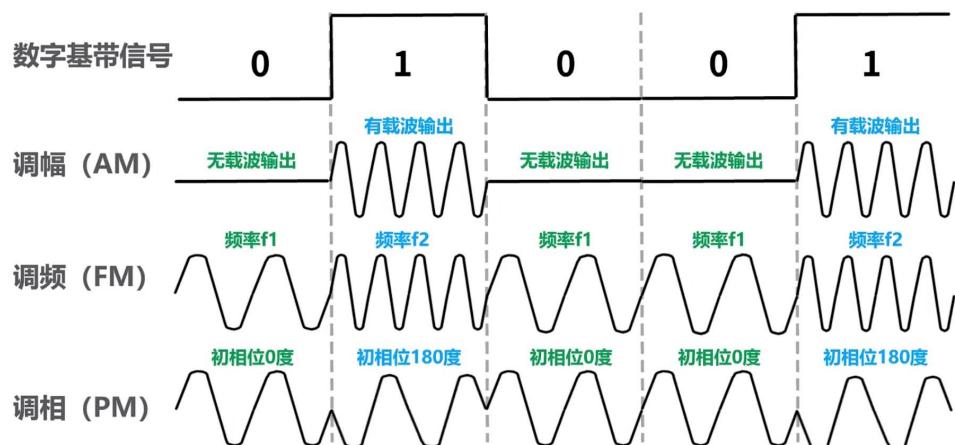
2.4 编码与调制

- 信号：数据的电磁表现
- 基带信号：信源发出的原始电信号
 - 数字基带信号
 - 编码 >> 数字信道
 - 以太网, Manchester
 - 调制 >> 模拟信道

- Wifi
- 模拟基带信号
 - 编码 >> 数字信号道
 - PCM
 - 调制 >> 模拟信道
 - FDM
- 码元：在使用时间域的波形表示数字信号时，代表不同离散数值的基本波形
- 常用编码



- 基本调制方法



- 混合调制方法

- 正交振幅调制QAM
 - QAM-16
 - 12种相位
 - 每种相位有1或2种振幅可选
 - 可以调制出16种码元（波形），每种码元对应4bit
 - 码元和比特的对应关系为格雷码

2.5 信道的极限容量

- 奈氏准则：在假定的理想条件下，为了避免码间干扰，码元的传输速率是有上限的

- 理想低通信道的最高码元传输速率 = $2W$ Baud (一般用这个)
 - 理想带通信道的最高码元传输速率 = W Baud

W: 信道带宽 (Hz)

Baud: 波特，即码元/秒

- 码元传输速率又称为波特率、调制速率、波形速率、符号速率
 - 码元传输速率与比特率的关系：

$$bit_rate = B \times \log_2 v$$

- 要提高信息传输速率，就必须提高码元数，采用多元制
 - 实际信道所能传输的最高码元速率，明显低于奈氏准则给出的这个上限数值

- 香农公式：带宽受限且有高斯白噪声干扰的信道的极限传输速率

- 公式：

$$c = W \times \log_2 \left(1 + \frac{S}{N} \right)$$

- c: 信道的极限传输速率 (bps)
 - W: 信道带宽 (Hz)
 - S/N: 信噪比
 - 信噪比，使用分贝作为单位

$$\text{信噪比}(dB) = 10 \times \log_{10}(S/N)(dB)$$

- 信道带宽或者信噪比越大，信息的极限传输速率越高
 - 实际信道上能够达到的信息传输速率，明显低于极限传输速率（有脉冲干扰、衰减、失真）

3 数据链路层

3.1 数据链路层概述

- 链路：从一个结点到另一个结点的一段物理线路，中间没有任何其他交换结点
- 数据链路：把实现通信协议的软硬件加到链路上，就构成了数据链路
- 数据链路层以帧为单位传输和处理数据
- 三个重要问题
 - 封装成帧
 - 差错检测
 - 可靠传输

3.2 封装成帧

- 封装成帧：数据链路层给上层交付的协议数据单元添加帧头和帧尾，使之成为帧
 - 帧头和帧尾中包含有重要的控制信息
 - 帧头和帧尾的作用之一是帧定界
- 透明传输：数据链路层对上层交付的传输数据没有任何限制，就好像数据链路层不存在一样
 - 面向字节的物理链路使用字节填充（字符填充）的方式，如转义字符
 - 面向比特的物理链路使用比特填充的方式
- 为了提高帧的传输效率，应使帧的数据部分的长度尽可能大
 - 考虑到差错控制等因素，每一种数据链路层的协议都规定了帧的数据部分的长度上限，即最大传送单元MTU

3.3 差错检测

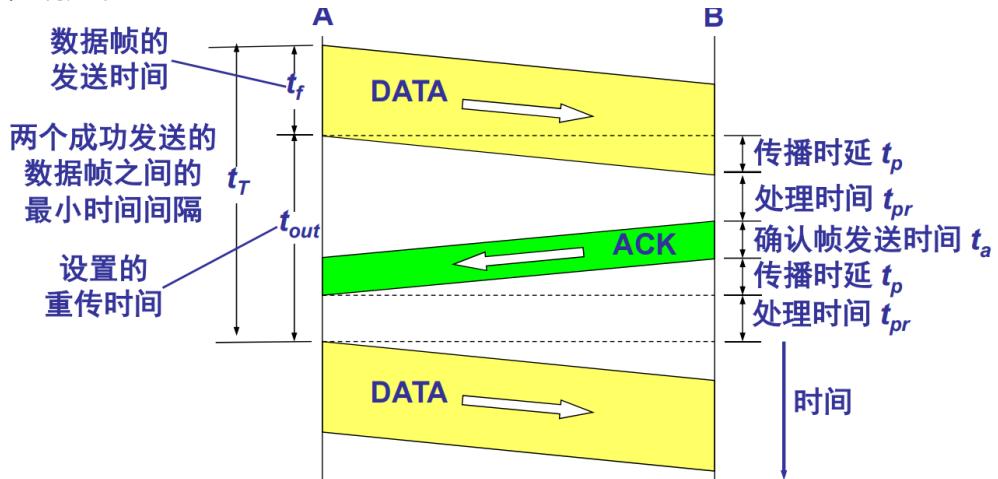
- 比特差错：比特在传输过程中可能会出差错，如 $1>0$, $0>1$
- 在一段时间内，传输错误的比特占所传输比特的总数的比率为误码率BER
- 使用差错检测码来检测数据在传输过程中是否产生了比特差错
- 差错检测方法
 - 奇偶校验
 - 在数据后面添加一位奇偶校验位，使整个数据中1的个数为奇数或偶数
 - 循环冗余校验CRC
 - 收发双方约定好一个生成多项式 $G(x)$
 - 发送方基于待发送的数据和生成多项式计算出差错校验码（冗余码），将其添加到待传输数据后面一起传输
 - 接收方通过生成多项式计算收到的数据是否产生了误码
- 差错纠正
 - 使用冗余信息更多的纠错码进行前向纠错（较少使用）
- 差错控制
 - 可靠传输：检错重传
 - 不可靠传输：丢弃检测到差错的帧

3.4 可靠传输

- 比特差错只是传输差错的一种，传输差错还包括分组丢失、分组失序、分组重复
- 可靠传输服务并不局限于数据链路层，其他各层均可选择实现可靠传输
- 停止-等待协议SW
 - 确认-否认：
 - ACK分组
 - NAK分组

接收方检测到数据分组有误码时，将其丢弃等待发送方超时重传；但对于误码率较高的点对点链路，为使发送方尽早重传，也可给发送方发送NAK分组

- 超时重传
 - 重传时间应仔细选择，一般设置为略大于“平均往返时间”
- 避免分组重复：给每个分组带上序号，一个比特编号（0/1）
 - 数据分组编号
 - ACK分组编号
- 信道利用率



$$\text{channel utilization } U = \frac{t_f}{t_T} = \frac{t_f}{t_p + t_{pr} + t_a + t_p + t_{pr} + t_f} = \frac{t_f}{2t_p + t_a + t_f}$$

- 当往返时延远大于数据帧发送时延时，信道利用率非常低
- 回退N帧协议GBN
 - 采用n个比特给分组编号，即序号0-2ⁿ-1
 - 滑动窗口
 - 发送窗口
 - 尺寸W_T取值：1<W_T<=2ⁿ-1
 - 使接收方分辨新旧分组
 - 接收窗口
 - 尺寸W_R取值：1
 - 发送方
 - 在未收到接收方确认分组的情况下，将序号落在发送窗口内的多个分组全部发送
 - 只有收到对已发送分组的确认时，发送窗口才能向前相应滑动
 - 收到多个重复确认时，可在重传计时器超时前尽早开始重传
 - 发送窗口内某个已发送的分组产生超时重发时，其后续在发送窗口内且已经发送的数据也必须全部重传
 - 接收方
 - 只能按序接收分组
 - 累积确认：
 - 可以在连续收到好几个按序到达且无误码的分组后，才针对最后一个分组发送确认
 - 或者可以在自己有数据分组要发送时才对之前按序到达且无误码的分组进行捎带确认
 - 收到未按序到达的分组时，除丢弃外，还要对最近按序接收的分组进行确认
- 选择重传协议SR

- 采用n个比特给分组编号，即序号0-7
- 滑动窗口
 - 发送窗口
 - 尺寸 W_T 取值： $1 < W_T \leq 2^{n-1}$
 - 使接收方分辨新旧分组
 - 接收窗口
 - 尺寸 W_R 取值： $W_R = W_T$
- 发送方
 - 在未收到接收方确认分组的情况下，将序号落在发送窗口内的多个分组全部发送
 - 只有按序收到对已发送分组的确认时，发送窗口才能向前相应滑动
 - 若收到未按序到达的确认分组，对其进行记录，以防止其相应数据分组的超时重发，但发送窗口不能滑动
- 接收方
 - 可接收未按序到达但没有误码的并且序号落在接收窗口内的数据分组
 - 为了使发送方仅重传出现差错的分组，接收方不能再采用累积确认，而需要对每个正确接受到的数据分组进行逐一确认
 - 只有在按序接受到分组后，接收窗口才能向前相应滑动

3.5 点对点协议PPP

- PPP是目前使用最广泛的点对点数据链路层协议
- PPP协议为再点对点链路传输各种协议数据报提供了一个标准方法，主要由三个部分组成：
 - 对各种协议数据报的封装方法：封装成帧
 - 链路控制协议LCP：用于建立、配置以及测试数据链路的连接
 - 一套网络控制协议NCPs：其中的每一个协议支持不同的网络层协议
- 帧格式



标志 (Flag) 字段： PPP帧的定界符，取值为0x7E

地址 (Address) 字段： 取值为0xFF，预留（目前没有什么作用）

控制 (Control) 字段： 取值为0x03，预留（目前没有什么作用）

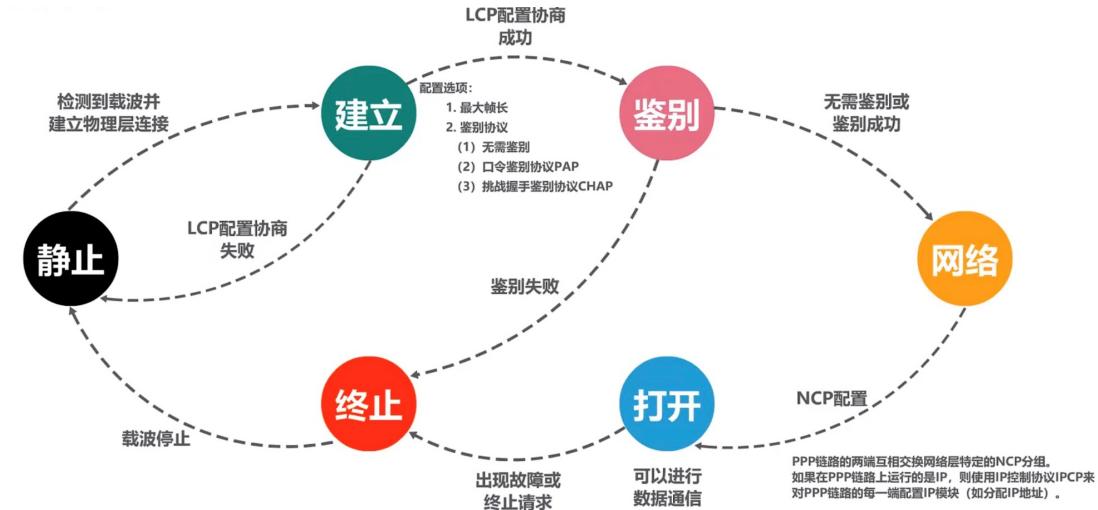
协议 (Protocol) 字段： 指明帧的数据部分递交哪个协议处理

| | | | |
|-------------------------|---------------|-------|--------|
| 取值0x0021表示：帧的数据部分为IP数据报 | 7E FF 03 0021 | IP数据报 | FCS 7E |
| 取值0xC021表示：帧的数据部分为LCP分组 | 7E FF 03 C021 | LCP分组 | FCS 7E |
| 取值0x8021表示：帧的数据部分为NCP分组 | 7E FF 03 8021 | NCP分组 | FCS 7E |

帧检验序列 (Frame Check Sequence) 字段： CRC计算出的校验位

- 实现透明传输的方法
 - 面向字节的异步链路：字节填充
 - 面向比特的同步链路：比特填充
- 使用PPP的数据链路层向上不提供可靠传输服务
 - 检错丢弃

- 工作状态



3.6 媒体介入控制MAC

- MAC概述

- 共享信道要考虑的一个重要问题是如何协调多个发送和接收站点对一个共享传输媒体的占用，即媒体介入控制
- 媒体介入控制分类
 - 静态划分信道 (物理层中使用)
 - 频分多址
 - 时分多址
 - 码分多址

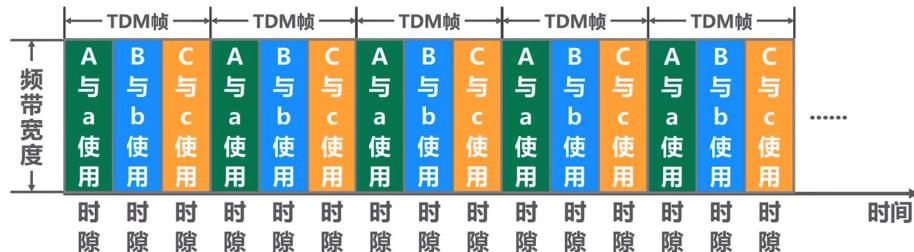
多址：动态分配信道给用户，用户暂时性地占用信道

- 动态接入控制
 - 受控接入
 - 集中控制：有一个主站以循环方式轮循每个站点有无数据发送，只有被轮询到的站点才能发送数据。存在单点故障问题
 - 分散控制：各站点平等，并连接成一个环形网络。令牌（一个特殊的控制帧）沿环逐站传递，接受到令牌的站点才有权发送数据，并在发送完数据后将令牌传递给下一个站点
 - 随机接入：所有站点通过竞争，随机地在信道上发送数据。如果恰巧有两个或更多的站点在同一时刻发送数据，则信号在共享媒体上就要产生碰撞（冲突），使得这些站点地发送都失败
 - CSMA/CD (总线型局域网，如以太网)
 - CSMA/CA (无线局域网，如802.11)
- 静态划分信道
 - 信道复用：复用是通过一条物理线路同时传输多路用户的信号。将单一媒体的频带资源划分成很多的子信道，这些子信道之间相互独立、互不干扰。
 - 常见复用技术

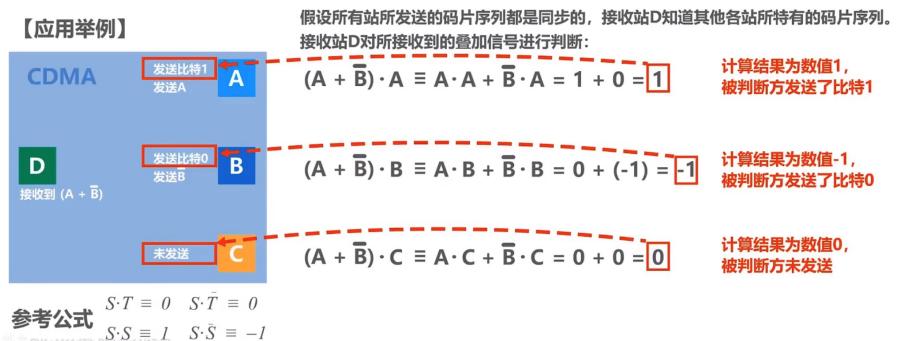
- 频分复用FDM：频分复用的所有用户同时占用不同的频带资源进行通信



- 时分复用TDM：时分复用的所有用户在不同的时间占用相同的频带宽度



- 波分复用WDM：光的频分复用
- 码分复用CDM：由于该技术主要用于多址接入，又称为码分多址CDMA。码分复用的每一个用户可以在相同的时间内使用相同的频带进行通信。由于各用户使用经过特殊挑选地不同码型，各用户之间不会相互干扰。
 - 在CDMA中，每一个比特时间再划分为m个短的时间间隔，称为码片（Chip）。
 - 使用CDMA的每一个站被指派一个唯一的m bit码片序列
 - 一个站如果要发送比特1，则发送它自己的码片序列
 - 一个站如果要发送比特0，则发送它自己的码片序列的二进制反码
 - 码片序列的挑选原则：
 - 分配给每个站的码片序列必须各不相同，实际常采用伪随机码序列
 - 分配给每个站的码片序列必须相互正交（内积为0）



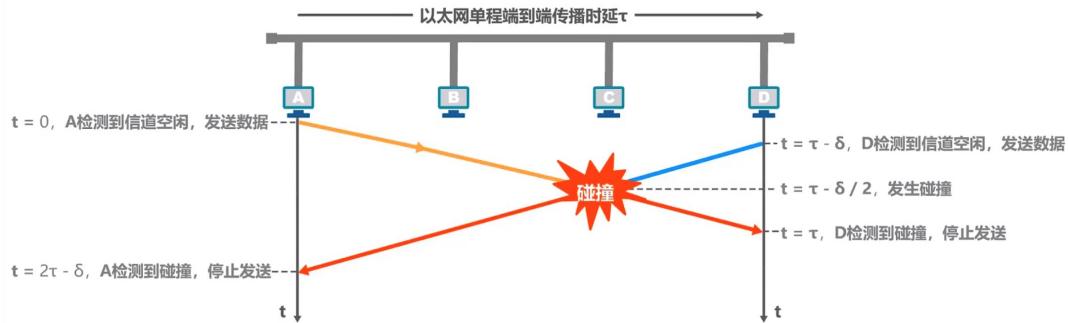
- CSMA/CD

- 载波监听多址接入/碰撞检测CSMA/CD
 - 多址接入MA：多个站连接在一条总线上，竞争使用总线
 - 载波监听CS：每一个站在发送帧之前先要检测总线上是否有其他站点在发送帧
 - 若检测到总线空闲96比特时间（帧间最小间隔），则发送帧
 - 若检测到总线忙，则继续检测并等待总线转为空闲96比特时间，然后发送这个帧
 - 碰撞检测CD：每一个正在发送的站边发送边检测碰撞
 - 一旦发现总线上出现碰撞，则立即停止发送，退避一段随机时间后，再次发送

【强化碰撞】

以太网采取的措施。当发送帧的站点检测到碰撞，除了立即停止发送帧，还要继续发送32比特或48比特的认为干扰信号，以便有足够的碰撞信号使所有的站点都能检测出碰撞

- 争用期（碰撞窗口）



- 主机最多经过 2τ ($\delta \rightarrow 0$) 的时延就可检测到本次发送是否产生碰撞
- 以太网端到端往返时延 2τ 称为争用期或碰撞窗口
- 经过争用期这段时间还没有检测到碰撞，才能肯定这次发送不会发生碰撞
- 以太网中发送帧的主机越多，端到端的往返时延越大，发生碰撞的概率就越大。因此，共享式以太网不能连接太多的主机，使用的总线也不能太长

- 最小帧长

- 最小帧长 = 争用期 × 数据传输速率
- 以太网规定最小帧长为64字节，即512比特（512比特发送时间即为争用期）
 - 如果要发送的数据非常少，那么必须加入一些填充字节，使帧长不少于64字节
- 以太网的最小帧长保证了主机可在发送完成之前就检测到该帧过程中是否遭遇了碰撞

- 最大帧长

- 防止缓冲区满
- 防止占用信道时间过长

- 截断二进制指数退避算法

- 退避时间 = 基本退避时间 × 随机数r
 - 基本退避时间：争用期 2τ
 - 随机数r：从离散的整数集合 $\{0, 1, \dots, (2^k - 1)\}$ 中随机选出一个数， $k = \text{Min}[\text{重传次数}, 10]$
- 若连续发生多次碰撞，就表明可能有较多的主机参与竞争信道
- 使用退避算法可使重传需要推迟的平均时间随重传次数而增大（也称为动态退避），因而减小发生碰撞的概率
- 当重传达16次仍不能成功时，表明打算发送帧的主机太多，以至于连续发生碰撞，则丢弃该帧，向高层报告

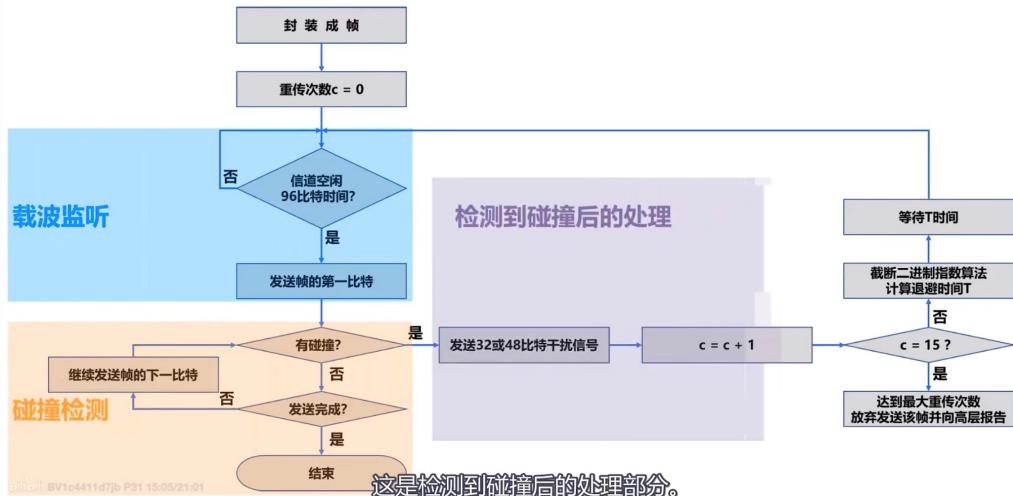
- 信道利用率

- 理想情况下（没有碰撞，总线充分利用）的极限信道利用率

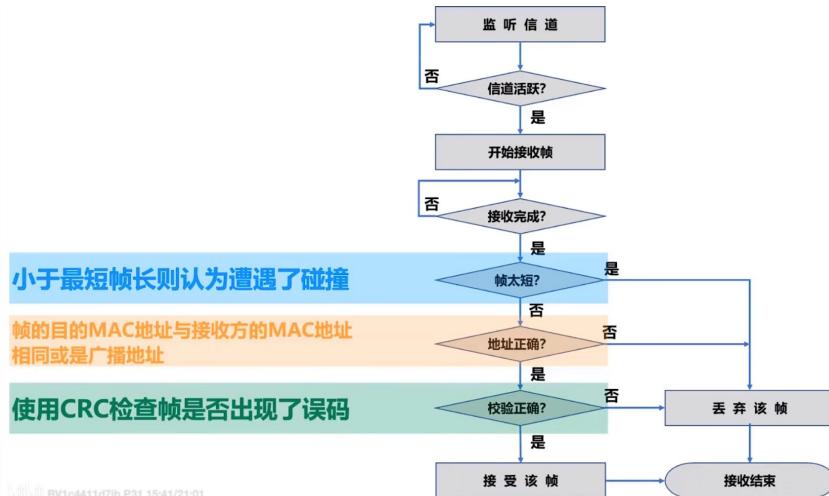
$$S_{max} = \frac{T_0}{T_0 + \tau} = \frac{1}{1 + \frac{\tau}{T_0}}$$

- 以太网端到端的距离尽量小
- 以太网帧的长度尽量大

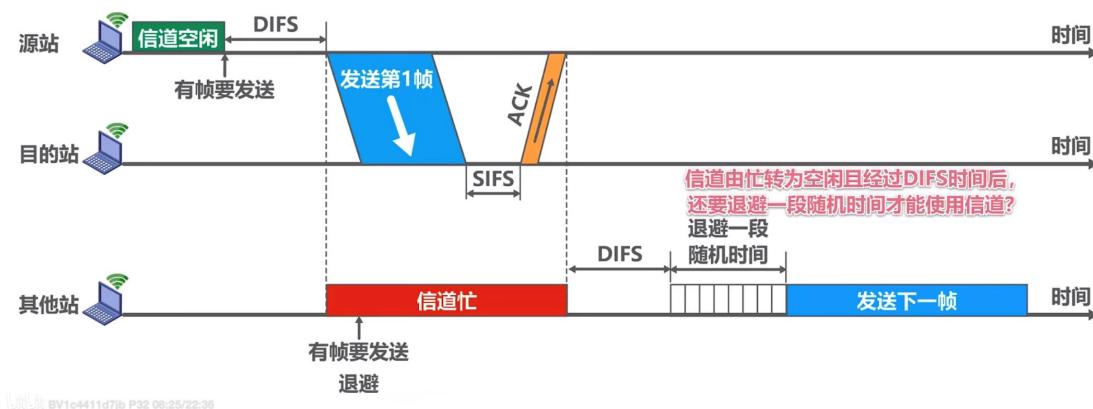
- 帧发送流程



- 帧接收流程



- CSMA/CA



- 载波监听多址接入/碰撞避免 CSMA/CA

- 无线局域网中，不能使用碰撞检测
- 由于不可能避免所有碰撞，且无线信道误码率高，802.11标准使用了数据链路层确认机制（SW）
- 802.11的MAC层标准定义了两种媒体接入控制方式
 - 分布式协调功能DCF（使用CSMA/CA协议，默认）
 - 点协调功能PCF（接入点AP集中控制，较少使用）

- 帧间间隔IFS

- 802.11标准规定，所有站点必须在持续检测到信道空闲一段指定时间后才能发送帧，这段时间间隔称为IFS

- 帧间间隔的长短取决于该站点要发送的帧的类型
 - 高优先级帧需要等待的时间较短，因此可优先获得发送权
 - 低优先级帧需要等待的时间较长。
- 常用的两种帧间间隔
 - 短帧间间隔SIFS (28μs)
 - 最短的帧间间隔，用来分隔开属于一次对话的各帧
 - 一个站点应当能够在这段时间内从发送方式切换到接收方式
 - 使用SIFS的帧类型有：ACK、CTS、由过长的MAC帧分片后的数据帧、所有回答AP探询的帧、PCF方式下接入点AP发送出的任何帧
 - DCF帧间间隔DIFS (128μs)
 - 在DCF方式中用来发送数据帧和管理帧

○ 工作原理

- 源站在检测到信道空闲后还要再等待一段DIFS再发送数据帧：考虑可能有其他的站有高优先级的帧要发送
- 目的站在正确接收数据帧后还要等待一段SIFS再发送ACK帧：保证站点从发送方式切换到接收方式
- 信道从忙转为空闲且经过一段DIFS，还要退避一段随机时间才能使用：防止多个站点同时发送数据而产生碰撞
- 当站点检测到信道是空闲的，且所发送的数据帧不是成功发送完上一个数据帧之后立即连续发送的数据帧，则不使用退避算法
- 必须使用退避算法的情况：
 - 在发送数据帧之前检测到信道处于忙状态
 - 在每一次重传一个数据帧时；
 - 在每一次成功发送后要连续发送下一个帧时（为了避免长时间占用信道）

○ 退避算法

- 在执行退避算法时，站点为退避计时器设置一个随机的退避时间
 - 当退避计时器的时间减小到0时，开始发送数据
 - 当退避计时器的时间还未减小到零时而信道又转变为忙状态，这时就冻结退避计时器的数值，重新等待信道变为空闲，再经过时间DIFS后，继续启动退避计时器
- 在进行第*i*次退避时，退避时间在时隙编号 (0, 1, ..., 2ⁱ⁺¹-1) 中随机选择一个，然后乘以基本退避时间（一个时隙的长度），得到随机的退避时间。当时隙编号达到255（对应第6次退避）就不再增加了

○ 信道预约和虚拟载波监听

- 信道预约：为了尽可能减少碰撞的概率和降低碰撞的影响，802.11标准允许要发送的数据的站点对信道进行预约
 1. 源站在发送数据帧之前先发送一个短的控制帧，称为请求发送RTS (Request To Send)，包括源地址、目的地址和这次通信（包括相应的确认帧）所需的持续时间
 2. 若目的站正确的收到源站发来的RTS帧，且媒体空闲，就发送一个响应控制帧，称为允许发送CTS (Clear To Send)，它也包括这次通信所需的持续时间（从RTS帧中复制）
 3. 源站收到CTS帧后，再等待一段时间SIFS后，就可以发送其数据帧
 4. 若目的站正确收到了源站发来的数据帧，在等待时间SIFS后，就向源站发送确认帧ACK
- 除源站和目的站外的其他各站，在收到CTS帧（或数据帧）后就推迟接入到无线局域网中

- 如果RTS帧发生碰撞，源站就收不到CTS帧，需执行退避算法重传RTS帧。由于RTS帧和CTS帧很短，发送碰撞的概率、碰撞产生的开销以及本身的开销都很小。
而对于一般的数据帧，其发送时延往往大于传播时延（因为是局域网），碰撞的概率很大，且一旦发生碰撞而导致数据帧重发，则浪费的时间就多。
因此用很小的代价对信道进行预约是值得的
- 802.11标准规定了3种可选情况
 - 使用RTS和CTS
 - 不使用RTS和CTS
 - 只有当数据帧长度超过一定值时，才使用RTS和CTS
- 虚拟载波监听机制：除RTS和CTS会携带通信所需要持续的时间，数据帧也能携带通信所需要持续的时间
- 利用虚拟载波监听，站点只需要监听到RTS、CTS、数据帧中的任何一个，就能知道信道被占用的持续时间，而不需要真正监听到信道上的信号，因此虚拟载波监听机制能减少隐蔽站带来的碰撞问题

3.7 MAC地址、IP地址、ARP协议

- 概念
 - MAC地址是以太网的MAC子层所使用的地址
 - IP地址是TCP/IP体系结构网际层使用的地址
 - ARP协议属于TCP/IP的网际层，其作用是已知设备所分配到的IP地址，使用ARP协议可以通过该IP地址获取到设备的MAC地址
- MAC地址
 - 当多个主机连接在同一个广播信道上，要想实现两个主机之间的通信，则每个主机都必须有一个唯一的标识，即一个数据链路层地址
 - 使用点对点信道的数据链路层不需要使用地址
 - 在每个主机发送的帧中必须携带标识发送主机和接收主机的地址。由于这类地址是用于媒体接入控制MAC，因此这类地址被称为MAC地址
 - MAC地址一般被固化在网卡（网络适配器）的电可擦可编程只读存储器EEPROM中，因此MAC地址也被称为硬件地址
 - MAC地址有时也被称为物理地址。（不属于物理层）
 - 一般情况下，用户主机会包含两个网络适配器：有线局域网适配器（有线网卡）和无线局域网适配器（无线网卡）。每个网络适配器都有一个全球唯一的MAC地址。而交换机和路由器往往拥有更多的网络接口，所以拥有更多的MAC地址。严格来说，MAC地址是对网络上接口的唯一标识，而不是对设备的唯一标识
 - MAC地址格式

IEEE 802局域网的MAC地址格式

扩展的唯一标识符EUI

EUI-48

| 组织唯一标识符OUI (由IEEE的注册管理机构分配) | | | | | | 网络接口标识符 (由获得OUI的厂商自行随意分配) | | | | | |
|--------------------------------|----|------|----|------|----|------------------------------|----|------|----|------|----|
| 第一字节 | | 第二字节 | | 第三字节 | | 第四字节 | | 第五字节 | | 第六字节 | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | b7 | b6 | b5 | b4 |
| 0: 全球管理 1: 本地管理 | | | | | | 0: 单播 1: 多播 | | | | | |

| 第一字节的b1位 | 第一字节的b0位 | MAC地址类型 | 地址数量占比 | 总地址数量 |
|----------|----------|---|--------|---|
| 0 | 0 | 全球管理 单播地址 厂商生产网络设备（网卡、交换机、路由器）时固化 | 1/4 | $2^{48} = 281,474,976,710,656$ (二百八十多万亿) |
| | 1 | 全球管理 多播地址 标准网络设备所支持的多播地址，用于特定功能 | 1/4 | |
| 1 | 0 | 本地管理 单播地址 由网络管理员分配，覆盖网络接口的全局管理单播地址 | 1/4 | |
| | 1 | 本地管理 多播地址 用户对主机进行软件配置，以表明其属于哪些多播组 <small>注意：剩余46位为1时，就是广播地址FF-FF-FF-FF-FF-FF</small> | 1/4 | |

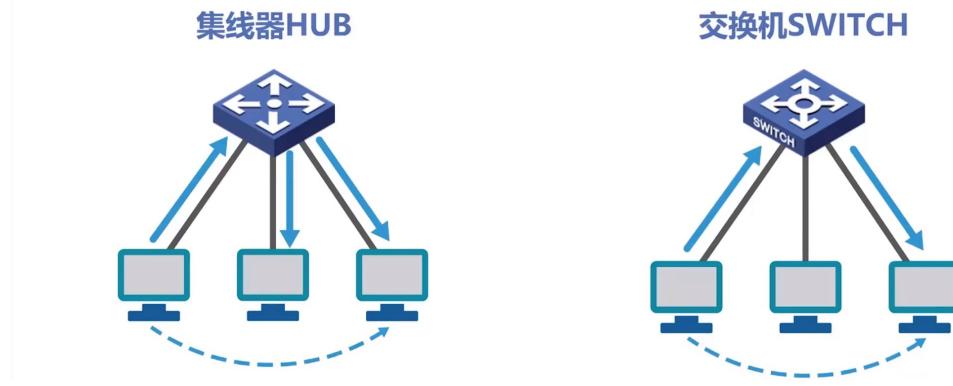
- 扩展的唯一标识符EUI：EUI-48
 - 组织唯一标识符OUI
 - 网络接口标识符
 - 表示方法
 - 标准表示法：`xx-xx-xx-xx-xx-xx` (Windows)
 - 其他表示法：`xx:xx:xx:xx:xx:xx` (ios, Android) / `xxxx.xxxx.xxxx` (Packer Trace)
 - MAC地址发送顺序
 - 字节发送顺序：第一字节 >> 第六字节
 - 字节内的比特发送顺序： $b_0 >> b_7$
- IP地址
 - IP地址是因特网上的主机和路由器所使用的地址，用于标识两部分信息：
 - 网络编号：标识因特网上数以万计的网络
 - 主机编号：标识同一网络上不同主机的接口（或路由器各接口）
 - MAC地址不具备区分不同网络的功能
 - 如果只是一个单独的网络，不接入因特网，则可以只使用MAC地址（不是一般方式）
 - 如果主机所在网络需要接入因特网，则IP地址和MAC地址都要使用
 - 数据包转发过程中
 - 源IP地址和目的IP地址保持不变
 - 源MAC地址和目的MAC地址逐个链路（逐个网络）改变
 - 对于一个链路，发送方知道接收方的IP地址，但不知道接收方的MAC地址
- ARP协议
 - 地址解析协议ARP：通过IP地址找到MAC地址
 - ARP协议只能在一段链路或一个网络内使用，不能跨网络使用
 - 除ARP请求和响应外，还有其他类型报文如无故ARP、免费ARP
 - ARP没有安全验证机制，存在ARP攻击问题
 - ARP请求报文（广播）
 - 封装在MAC帧中，目的地地址是FF-FF-FF-FF-FF-FF
 - 包含信息：
 - 本机IP地址
 - 本机MAC地址
 - 请求查询IP地址
 - ARP响应报文（单播）
 - 封装在MAC帧中
 - 包含信息：
 - 本机IP地址
 - 本机MAC地址
 - ARP高速缓存
 - 包含信息：
 - IP地址
 - MAC地址
 - 类型

- 静态：手工设置，不同操作系统下的生命周期不同，例如系统重启后不存在或系统重启后仍有效
- 动态：自动获取，生命周期默认为两分钟，生命周期结束时该记录自动删除
- 地址解析过程
 1. 发送方查找ARP高速缓存，若存在则直接找到MAC地址
 2. 若不存在，则发送方广播ARP请求报文
 3. 目的接收方收到ARP请求报文并确认后
 - 将发送方的IP地址与MAC地址记录到自己的ARP高速缓存中
 - 给发送方发送ARP响应，告知自己的MAC地址
 4. 发送方收到ARP响应后，将MAC地址记录在ARP高速缓存

3.8 集线器与交换机的区别

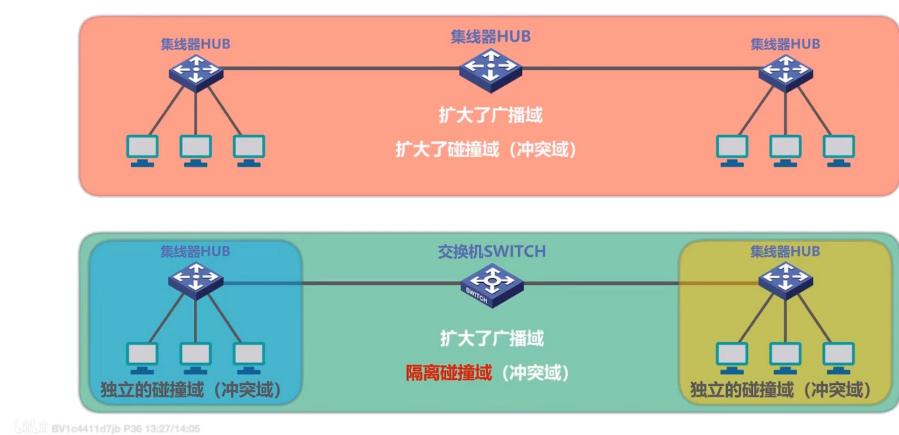
- 早期的总线型以太网
 - 使用同轴电缆和机械接头
- 星型以太网
 - 使用双绞线和集线器HUB
- 集线器HUB
 - 使用集线器的以太网逻辑上仍是一个总线网，各站共享总线资源，使用的还是CSMA/CD协议
 - 集线器只工作在物理层，每个接口仅简单地转发比特，不进行碰撞检测（由各站的网卡检测）
 - 集线器一般都有少量的容错能力和网络管理功能
 - 使用集线器HUB在互联网扩展以太网
- 以太网交换机
 - 以太网交换机通常有多个接口，每个接口都可以直接与一台主机或另一个以太网交换机相连。一般工作在全双工方式。
 - 具有并行性，能同时连同多对接口，使多对主机能同时通信，无碰撞（不使用CSMA/CD协议）
 - 一般都具有多种速率的接口
 - 以太网交换机工作在数据链路层（也包括物理层），收到帧后，在帧交换表中查找帧的目的MAC地址所对应的接口号，然后通过该接口转发帧
 - 即插即用设备，其内部的帧交换表是通过自动学习算法自动地逐渐建立起来的
 - 帧的两种转发方式
 - 存储转发
 - 直通交换：采用基于硬件的交换矩阵（交换时延小；不检查帧是否差错）
- 集线器和交换机对比

- 单播：



- 集线器对接受到的信号进行放大转发
- 交换机根据MAC地址对帧进行转发
- 广播：使用集线器或使用交换机互连起来的主机都属于同一广播域
- 碰撞：

 - 使用集线器互连起来的所有主机属于同一个碰撞域
 - 交换机可以根据MAC地址过滤帧，即隔离碰撞域



3.9 以太网交换机自学习和转发帧的流程

- 以太网交换机收到帧后，在帧交换表中查找帧的目的MAC地址所对应的接口号，然后通过该接口转发帧
- 以太网交换机刚通电时其内部的帧交换表是空的，随着网络中各主机的通信，以太网交换机通过自学习算法自动逐渐地建立帧交换表
- 自学习和转发帧的流程
 - 登记 转发（盲目泛洪）：帧交换表中找不到相应MAC地址
 - 登记 转发（明确）：帧交换表中找到相应MAC地址
 - 登记 丢弃：帧交换表中查找到的MAC地址对应的接口即是源接口
- 每条记录都有自己的有效时间，到期自动删除
 - 这是因为MAC地址与交换机接口的对应关系并不是永久性的

3.10 以太网交换机生成树协议STP

- 添加冗余链路可以提高以太网的可靠性
 - 冗余链路也会带来负面效应：形成网络环路
 - 网络环路问题：
 - 广播风暴：大量消耗网络资源，使得网络无法正常转发其他数据帧
 - 主机收到重复的广播帧：大量消耗主机资源
 - 交换表的帧交换表震荡（漂移）
- 以太网交换机使用生成树协议STP，可以在增加冗余链路的同时提高网络的可靠性。同时避免网络环路问题
 - 无论交换机之间采用怎样的物理连接，交换机都能自动地计算并构建一个逻辑上没有环路的网络，其逻辑拓扑结构必须是树型的（无逻辑环路）
 - 利用最小生成树算法，计算哪些接口应被阻塞，并确保联通整个网络
 - 当首次连接交换机或网络物理拓扑发生变化时（可能是人为改变或故障），交换机都将进行最小生成树重新计算

3.11 虚拟局域网VLAN

- VLAN概述
 - 巨大的广播域会带来很多弊端：
 - 广播风暴
 - 难以维护管理
 - 潜在的安全问题
 - 网络中会频繁使用广播信息
 - TCP/IP协议栈中很多协议使用广播
 - 地址解析协议ARP
 - 路由信息协议RIP
 - 动态主机配置协议DHCP
 - NetBEUI：Windows下使用的广播型协议
 - IPX/SPX：Novell网络的协议栈
 - Apple Talk：Apple公司的网络协议栈
 - 分隔广播域的方法
 - 使用路由器隔离广播域
 - 成本较高
 - 虚拟局域网技术VLAN
 - VLAN是一种将局域网内的设备划分成与物理位置无关的逻辑组的技术，这些逻辑组具有某些共同的需求
 - 同一个VLAN中可以广播通信，不同的VLAN中不能广播通信
- VLAN实现机制
 - 基于交换机的两大功能：
 - 能够处理VLAN帧：IEEE 802.1Q帧
 - IEEE 802.1Q帧（Dot One Q帧）对以太网的MAC帧格式进行拓展，插入了4字节的VLAN标记

- VLAN标记的最后12bit称为VLAN标识符VID，它唯一地标识了以太网所属VLAN
 - VID取值0~4095
 - 0和4095都不用来标识VLAN，因此VID有效取值是1~4094
- 802.1Q帧是由交换机处理，而不是由主机处理
 - 当交换机收到普通的以太网帧时，会将其插入4字节的VLAN标记转变为802.1Q帧，即“打标签”
 - 当交换机转发802.1Q帧时，可能会删去4字节VLAN标记变为普通以太网帧，即“去标签”
- 支持不同的端口类型
 - 三种端口类型
 - Access
 - 一般用于连接用户计算机
 - 只能属于一个VLAN
 - PVID值与端口所属VLAN的ID相同（默认为1）
 - 接收处理方法：
 - 一般只接收“未打标签”的普通以太网MAC帧，根据接收帧的端口的PVID给帧打标签，及插入4字节的VLAN标记字段，字段中的VID取值与端口的PVID取值相同
 - 发送处理方法：若帧中的VID与端口的PVID相等，则“去标签”并转发该帧；否则不转发
 - Trunk
 - 一般用于交换机之间或交换机与路由器之间的互连
 - 可以属于多个VLAN
 - 用户可以设置Trunk端口的PVID值。默认情况下为1
 - 发送处理方法：
 - 对VID等于PVID的帧，“去标签”再转发
 - 对VID不等于PVID的帧，直接转发
 - 接收处理方法：
 - 接收“未打标签”的帧，根据接收帧的端口的PVID给帧“打标签”，即插入4字节的VLAN标记字段，字段中的VID值与端口的PVID取值相等
 - 接收“已打标签”的帧
 - Hybrid
 - 华为交换机私有
 - 既可以用于交换机之间或交换机与路由器之间的互连，也可用于交换机与用户计算机之间的互连
 - 交换机各端口的缺省VLAN ID
 - 在思科交换机上称为Native VLAN，即本征VLAN
 - 在华为交换机上称为Port VLAN ID，即端口VLAN ID，简记为PVID

4 网络层

4.1 网络层概述

- 网络层主要任务是实现网络互连，进而实现数据包在各网络之间传输
- 网络层解决的问题
 - 网络层向运输层提供怎样的服务
 - 网络层寻址问题
 - 路由选择问题
- 由于TCP/IP协议栈的网络层使用网际协议IP，它是整个协议栈的核心协议，因此TCP/IP协议栈中网络层常称为网际层

4.2 网络层提供的两种服务

- 面向连接的虚电路服务
 - 可靠通信由网络来保证
 - 必须建立网络层的连接——虚电路VC
 - 通信双方沿着已建立的虚电路发送分组
 - 目的主机的地址仅在连接建立阶段使用，之后每个分组的首部只需携带一条虚电路的编号（构成虚电路的每一段链路都有一个虚电路编号）
 - 这种通信方式如果再使用可靠传输的网络协议，就可使所发送的分组最终正确到达接收方（无差错按序到达、不丢失、不重复）
 - 通信结束后，需要释放之前所建立的虚电路
- 无连接的数据报服务
 - 可靠通信由用户主机来保证
 - 不需要建立网络层连接
 - 每个分组走不同的路径
 - 每个分组的首部必须携带目的主机的完整地址
 - 所传送的分组可能误码、丢失、重复和失序
 - 由于网络本身不提供端到端的可靠服务，这就使得网络中的路由器可以做的比较简单且价格低廉
 - 因特网采用这种设计思想，即将复杂的网络处理功能置于因特网边缘（用户主机和其内部的运输层），而将相对简单的尽最大努力分组交付功能置于因特网核心

4.3 IPv4地址

- IPv4概述
 - IPv4地址就是给因特网上的每一台主机（或路由器）的每一个接口分配一个在全世界范围内唯一的32bit的标识符
 - IP地址由因特网名字和数字分配机构ICANN进行分配
 - 我国用户可向亚太网络信息中心APNIC申请IP地址，需缴费
 - 编址方法发展：
 - 分类编址
 - 划分子网
 - 无分类编址
 - 采用点分十进制表示方法
 - 每8bit分为一组，共四组

- 写出每组的十进制数
- 分类编址的IPv4地址



bilibili BV1c4411d7jb P44 01:07/16:41

- 只有A、B、C类地址可分配给网络中的主机或路由器的各接口
- 主机号全0的地址是网络地址，不能分配给主机或路由器的各接口
- 主机号为全1的地址是广播地址，不能分配给主机或路由器的各接口
- 各分类详细：

■ A类

最小网络号0，保留不指派



00000000

第一个可指派的网络号为1，网络地址为1.0.0.0

00000001 00000000000000000000000000000000

最大网络号127，作为本地环回测试地址，不指派

01111111

最小的本地环回测试地址为127.0.0.1

01111111 00000000000000000000000000000001

最大的本地环回测试地址为127.255.255.254

01111111 11111111111111111111111111111110

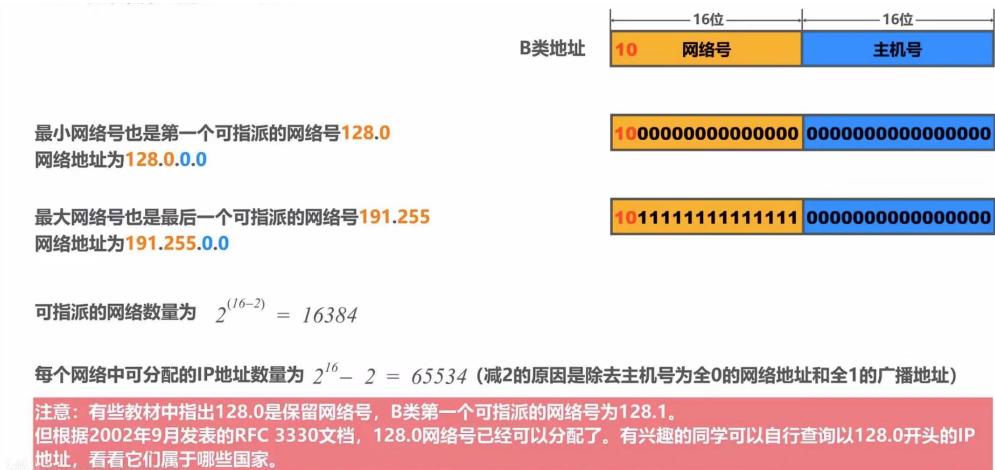
最后一个可指派的网络号为126，网络地址为126.0.0.0

01111110 00000000000000000000000000000000

可指派的网络数量为 $2^{(8-1)} - 2 = 126$ (减2的原因是除去最小网络号0和最大网络号127)

每个网络中可分配的IP地址数量为 $2^{24} - 2 = 16777214$ (减2的原因是除去主机号为全0的网络地址和全1的广播地址)

■ B类



■ C类



■ D类和E类

| 网络类别 | 作用 | 第一个地址 | 最后一个地址 | 地址数量 | 占总地址空间 |
|------|---------|-----------|-----------------|---------------------------|-------------------------------------|
| D | 多播地址 | 224.0.0.0 | 239.255.255.255 | 268435456 (2^{28}) | 1/16 ($2^{(32-4)}$ / 2^{32}) |
| E | 保留为今后使用 | 240.0.0.0 | 255.255.255.255 | 268435456 (2^{28}) | 1/16 ($2^{(32-4)}$ / 2^{32}) |

○ 一般不使用的特殊IP地址

| 一般不使用的特殊IP地址 | | | | |
|--------------|---------|-------|--------|-----------------------|
| 网络号 | 主机号 | 作为源地址 | 作为目的地址 | 代表的意思 |
| 0 | 0 | 可以 | 不可 | 在本网络上的本主机 (DHCP协议) |
| 0 | host-id | 可以 | 不可 | 在本网络上的某台主机host-id |
| 全1 | 全1 | 不可 | 可以 | 只在本网络上进行广播 (各路由器均不转发) |
| net-id | 全1 | 不可 | 可以 | 对net-id上的所有主机进行广播 |
| 127 | 非全0或全1 | 可以 | 可以 | 用于本地软件环回测试 |

● 划分子网的IPv4地址

- 从主机号借用一些位作为子网号
- 32bit的子网掩码可以表明分类IP地址的主机号部分被借用了几个bit作为子网号
 - 子网掩码使用连续的比特1对应网络号和子网号
 - 子网掩码使用连续的比特0对应主机号
 - 将划分子网的IPv4地址与其相应的子网掩码进行逻辑与运算就可得到IPv4地址所在子网的网络地址 (网络号和子网号被保留，主机号被清零)

划分子网数量 = $2^{\text{子网号位数}}$

每个子网可分配的IP地址数量 = $2^{\text{主机号位数}} - 2$

- 默认的子网掩码是指在未划分子网的情况下使用的子网掩码
 - A类: 255.0.0.0
 - B类: 255.255.0.0
 - C类: 255.255.255.0
- 无分类编址的IPv4地址
 - 无分类域间路由选择CIDR
 - CIDR消除了传统的A类、B类和C类地址，以及划分子网的概念
 - CIDR可以有效地分配IPv4的地址空间，并且可以在新的IPv6之前允许因特网的规模继续增长
 - 使用斜线记法，或称CIDR记法，即在IPv4地址后面加上斜线 /，在斜线后面写上网络前缀所占的比特数量
 - CIDR实际上是将网络前缀都相同的连续IP地址组成一个CIDR地址块
 - 只要知道CIDR地址块中的任何一个地址，就可以知道该地址块的全部细节
 - 地址块的最小地址
 - 地址块的最大地址
 - 地址块中的地址数量
 - 地址块聚合某类网络（A/B/C）的数量
 - $2^{\text{地址块主机号位数} - \text{某类网络主机号位数}}$
 - 地址掩码（子网掩码）
 - 路由聚合（构造超网）
 - 【举例】

| 路由器R2的路由表 | |
|----------------|-----|
| 目的网络 | 下一跳 |
| 172.1.4.0/25 | R1 |
| 172.1.4.128/25 | R1 |
| 172.1.5.0/24 | R1 |
| 172.1.6.0/24 | R1 |
| 172.1.7.0/24 | R1 |
| ... | ... |

找共同前缀

共同前缀
共22位

聚合地址块: 172.1.4.0 / 22

- 网络前缀越长，地址块越小，路由越具体
- 若路由器查表转发分组时发现有多条路由可选，则选择网络前缀最长的那条，称为最长前缀匹配；因为这样的路由更具体
- Ipv4地址的应用规划
 - 定长的子网掩码FLSM
 - 使用同一个子网掩码来划分子网
 - 每个子网所分配的IP地址数量相同，造成IP地址浪费
 - 变长的子网掩码VLSM
 - 使用不同的子网掩码来划分子网
 - 每个子网所分配的IP地址数量可以不同，尽可能减少对IP地址的浪费

4.4 IP数据报发送和转发

- 主机发送IP数据报
 - 若在一个网络，则直接交付
 - 若不在一个网络，则间接交付，传输给主机所在网络的默认网关（路由器），由默认网关进行转发
- 路由器转发IP数据报
 1. 检查IP数据报首部是否出错
 - 若出错，则直接丢弃该IP数据报并通告源主机
 - 若没有出错，则进行转发
 2. 根据IP数据报的目的地址在路由表中查找匹配的条目
 - 若找到匹配条目，则转发给条目中指示的下一跳
 - 若找不到，则丢弃该IP数据报并通告源主机

路由器不会转发广播数据报，隔离广播域

4.5 静态路由配置

- 路由条目类型
 - 直连网络
 - 静态路由
 - 动态路由
- 静态路由配置
 - 指用户或网络管理员使用路由器的相关命令给路由器人工配置路由表
 - 人工配置方式简单、开销小。但不能及时适应网络状态（流量、拓扑）的变化
 - 一般只在小规模网络中使用
- 静态路由配置
 - 默认路由：0.0.0.0/0
 - 特定主机路由：特定主机的IP地址，地址掩码为255.255.255.255
 - 黑洞路由
- 最长前缀匹配
- 使用静态路由配置可能出现以下导致产生路由环路的错误
 - 配置错误
 - 为了防止IP数据报在路由环路永久兜圈，在IP数据报首部设有生存时间TTL字段；IP数据报进入路由器后，TTL字段的值减1。若TTL不等于0，则被路由器转发，否则丢弃
 - 聚合了不存在的网络
 - 在路由表中添加针对所聚合的、不存在的网络的黑洞路由，下一跳为虚拟接口null0
 - 网络故障
 - 在路由表中添加故障网络的黑洞路由，下一跳为虚拟接口null0

4.6 路由选择协议

- 动态路由选择
 - 路由器通过路由选择协议自动获取路由信息
 - 比较复杂、开销比较大、能较好地适应网络状态变化
 - 适用于大规模网络
- 因特网所采用的路由选择协议的主要特点
 - 自适应
 - 动态路由选择，能较好地适应网络状态变化
 - 分布式
 - 路由器之间交换路由信息
 - 分层次
 - 将整个因特网划分为许多较小的自治系统AS
- 分层次的路由选择协议
 - 域间路由选择
 - 外部网关协议EGP（外部路由协议）
 - 边界网关协议BGP
 - 域内路由选择
 - 内部网关协议IGP（内部路由协议）
 - 路由信息协议RIP
 - 内部网关路由协议IGRP
 - 增强型内部网关路由协议EIGRP
 - 开放式最短路径优先OSPF
 - 中间系统到中间系统IS-IS
- 路由器的基本结构
 - 路由选择部分
 - 路由选择处理机：根据路由选择协议周期性地与其他路由器进行路由信息的交互来更新路由表
 - 路由表
 - 一般仅包含从目的网络到下一跳的映射
 - 需要对网络拓扑变化的计算最优化
 - 分组转发部分
 - 交换结构
 - 转发表
 - 从路由表得出
 - 转发表的结构应当使查找过程最优化
 - 一组输入端口
 - 输入缓冲区
 - 一组输出端口
 - 输出缓冲区
- 路由信息协议RIP的基本工作原理
 - 是内部网关协议IGP中最先得到广泛使用的协议之一
 - RIP报文被封装为UDP协议

- 要求自治系统AS内每一个路由器都要维护从它自己到AS内其他每一个网络的距离记录。这一组距离，称为“距离向量D-V”
- RIP使用跳数作为度量来衡量到达目的网络的距离
 - 路由器到直连网络的距离定义为1
 - 路由器到非直连网络的距离定义为所经过的路由器数加1
 - 允许一条路径最多只能包含15各路由器。“距离”等于16时相当于不可达。因此，RIP只适用于小型互联网
- RIP认为好的路由就是“距离短”的路由，也就是所通过路由器数量最少的路由
- 当到达同一目的网络有多条“距离相等”的路由时，可以进行等价负载均衡
- RIP包含以下三个要点：
 - 仅和相邻路由器交换信息
 - 交换的信息为自己的路由表
 - 周期性交换信息
- 基本工作过程
 - 路由器刚开始工作时，只知道自己到直连网络的距离为1
 - 每个路由器仅和相邻路由器周期性地交换并更新路由信息
 - 若干次交换和更新后，每个路由器都知道到达本AS内各网络的最短距离和下一跳地址，称为收敛
- 路由条目更新规则
 - 到达目的网络，相同下一跳，最新消息：更新
 - 发现新的网络：添加
 - 到达目的网络，不同下一跳，新路由优势：更新
 - 到达目的网络，不同下一跳，等价负载均衡
 - 到达目的网络，不同下一跳，新路由劣势：不更新

路由器C给路由器D发送路由表，路由表中的下一跳均记为?，可以理解为路由器D不关心路由器C路由表中的下一跳；

路由器D收到路由器C发来的路由表后，对其进行改造，将距离+1，将?改为C，可以理解为路由器C所有可达网络对于路由器D来说其下一跳均为C

- “坏消息传得慢”问题
 - 又称路由环路或距离无穷计数问题，这是距离向量算法的一个固有问题
 - 措施
 - 限制最大路径距离为15（16表示不可达）
 - 当路由表发生变化时就立即发送更新报文（“触发更新”），而不是周期性发送
 - 让路由器记录收到某特定路由信息，而不让同一路由信息再通过此接口反方向传送（“水平分割”）
- 开放最短路径优先OSPF的基本工作原理
 - 基于链路状态
 - 采用SPF计算路由，从算法上保证了不会产生路由环路
 - 不限制网络规模，更新效率高，收敛速度快
 - 链路状态是指本路由器都和哪些路由器相邻，以及相邻链路的“代价”
 - “代价”用来表示费用、时延、距离、带宽等。
 - OSPF相邻路由器之间通过交互问候分组，建立和维护邻居关系
 - Hello分组封装在IP数据报中，发往组播地址为244.0.0.5
 - 发送周期为10秒

- 40秒未收到来自邻居路由器的Hello分组，则认为该邻居路由器不可达
- 使用OSPF的每个路由器都会产生链路状态通告LSA，包含以下内容：
 - 直连网络的链路状态信息
 - 邻居路由器的链路状态信息
- LSA被封装在链路状态更新分组LSU中，采用洪泛法发送
- 使用OSPF的每个路由器都有一个链路状态数据库LSDB，用于存储LSA
- 通过各路由器洪泛发送封装有自己LSA的LSU分组，各路由器的LSDB最终将到达一致
- 使用OSPF的各路由器基于LSDB进行最短路径优先SPF计算，构建出各自到达其他各路由器的最短路径，即构建各自的路由表
- OSPF有以下分组类型
 - 问候分组：发现和维护邻居路由器的可达性
 - 数据库描述分组：向邻居路由器给出自己的链路状态数据库中的所有链路状态项目的摘要信息
 - 链路状态请求分组：向邻居路由器请求发送某些链路状态项目的详细信息
 - 链路状态更新分组：路由器使用这种分组将其链路状态进行洪泛发送，即用洪泛法对全网更新链路状态
 - 链路状态确认分组：对链路状态更新分组的确认分组
- 基本工作过程
 1. 给邻居路由器发送问候分组，建立和维护邻居关系
 2. 给邻居发送数据库描述分组
 3. 若发现自己缺少链路状态项目，则给邻居发送链路状态请求分组
 4. 发送链路状态更新分组
 5. 发送链路状态确认分组
 6. 链路状态数据库达到同步
- OSPF在多点接入网络中路由器邻居关系的建立
 - 选举指定路由器DR和备用的指定路由器BDR
 - 所有的非DR/BDR只与DR/BDR建立邻居关系
 - 非DR/BDR之间通过DR/BDR交换信息
- 为了使OSPF能够用于规模很大的网络，OSPF把一个自治系统再划分为若干个更小的范围，叫做区域
 - 32bit区域标识符
 - 把利用洪泛法交换链路装填信息的范围局限于每一个区域而不是整个自治系统，减少了整个网络上的通信量
 - 路由器分类
 - 区域内路由器IR：所有接口都在一个区域内
 - 区域边界路由器ABR：有一个接口连接主干区域
 - 主干路由器BBR：主干区域内的路由器
 - 自治系统边界路由器ASBR：有一个接口连接至其他自治系统
- 边界网关协议BGP的基本工作原理
 - 在不同的自治系统内，度量路由的代价可能不同。因此，对于自治系统之间的路由选择，使用“代价”作为度量来寻找最佳路由是不行的
 - 自治系统之间我的路由选择必须考虑相关策略（政治，经济，安全等）
 - BGP只能是力求寻找一条能够到达目的网络且比较好的路由（不能兜圈子），而并非寻找一条最佳路由
 - 在配置BGP时，每个自治系统的管理员要选择至少一个路由器作为该自治系统的“BGP发言人”

- 不同自治系统的BGP发言人要交换路由信息，首先必须建立TCP连接，端口号为179
 - 在此TCP连接上交换BGP报文以建立BGP对话
 - 利用BGP会话交换路由信息（增加新的路由、撤销过时的路由、报告出错的情况）
 - 使用TCP连接交换路由信息的两个BGP发言人，彼此称为对方的邻站或对等站
- BGP发言人除了运行BGP之外，还必须运行自己所在自治系统所使用的内部网关协议IGP，例如OSPF或RIP
- BGP发言人交换网络可达性的信息（要到达某个网络所要经过的一系列自治系统）
- 当BGP发言人互相交换了网络可达性的信息后，各BGP发言人就根据所采用的策略从收到的路由信息中找出到达各自系统的较好的路由。也就是构造出树形结构、不存在回路的自治系统连通图
- BGP适用于多级结构的因特网
- BGP-4有以下四种报文
 - OPEN（打开）报文：用来与相邻的另一个BGP发言人建立关系，使通信初始化
 - UPDATE（更新）报文：用来通告某一路由的信息，以及列出要撤销的多条路由
 - KEEPALIVE（保活）报文：用来周期性地证实邻站的连通性
 - NOTIFICATION（通知）报文：用来发送检测的差错

4.7 IPv4数据报的头部格式

- 版本
 - 占4bit，表示IP协议的版本
 - 通信双方使用的IP协议版本必须一致。目前广泛使用的IP协议版本号为4（IPv4）
- 首部长度
 - 占4bit，表示IP数据报首部的长度。该字段的取值以4字节为单位
 - 最小十进制取值为5，最大十进制取值为15
- 可选字段
 - 长度从1byte到4byte不等，用来支持排错、测量以及安全等措施
 - 可选字段增加了IP数据报的功能，但这同时也使得IP数据报的首部长度称为可变的。这就增加了每一个路由器处理IP数据报的开销。实际上可选字段很少被使用
- 填充
 - 确保首部长度为4字节的整数倍，使用全0进行填充
- 区分服务
 - 占8bit，用来获得更好的服务
 - 该字段在旧标准中叫做服务类型，实际上从未被使用过
 - 利用该字段的不同数值可提供不同等级的服务质量
- 总长度
 - 占16bit，表示IP数据报的总长度（首部+数据载荷）
 - 最大取值为十进制的65535，以字节为单位
- 标识
 - 占16bit，属于同一个数据报的各分片数据报应该具有相同的标识
 - IP软件维持一个计数器，每产生一个数据报，计数器值加1并将此值赋给标识
- 标志
 - 占3bit
 - DF位：

- 1: 不允许分片
 - 0: 允许分片
 - MF位:
 - 1: 后面还有分片
 - 0: 当前为最后一个分片
 - 保留位: 必须为0
 - 片偏移
 - 占13bit, 指出分片数据报的数据载荷部分偏移其在原数据报的位置有多少个单位
 - 片偏移以8字节为单位
- 每个IP数据报分片都会被加上首部, 首部只有片偏移不同
- 生存时间
 - 占8bit, 最初以秒为单位, 最大生存周期为255秒; 路由器转发IP数据报时, 将IP数据报首部中的该字段的值减去IP数据报在本路由器上所耗费的时间, 若不为0就转发, 否则就丢弃
 - 现在以跳数为单位, 路由器转发IP数据报时, 将IP数据报首部中的该字段的值-1, 若不为0就转发, 否则就丢弃
 - 协议
 - 占8bit, 指明IPv4数据报的数据部分是何种协议数据单元。常用的一些协议和相应的协议字段值如下

| 协议名称 | ICMP | IGMP | TCP | UDP | IPv6 | OSPF |
|-------|------|------|-----|-----|------|------|
| 协议字段值 | 1 | 2 | 6 | 17 | 41 | 89 |

- 首部检验和
 - 占16bit, 用来检测首部在传输过程中是否出现差错。比CRC检验码简单, 称为因特网检验和
 - IP数据报没经过一个路由器, 路由器都要重新计算首部检验和, 因为某些字段值可能发生变化
- 源IP地址和目的IP地址
 - 各占32bit, 用来填写发送该IP数据报的源主机IP地址和接收该IP数据报的目的主机IP地址

4.8 网际控制报文ICMP

- 为了更有效地转发IP数据报和提高交付成功的机会, 在网际层使用了网际控制报文协议ICMP
- 主机或路由器使用ICMP来发送差错报告报文和询问报文
- ICMP报文被封装在IP数据报中发送
- ICMP差错报告报文
 - 终点不可达
 - 路由器或主机不能交付数据报, 发送终点不可达报文给源点
 - 源点抑制
 - 路由器或主机由于拥塞而丢弃数据报, 发送源点抑制报文给源点
 - 使源点知道应当把数据报的发送速率放慢
 - 时间超过
 - 路由器收到一个目的IP地址不是自己的IP数据报, 将其TTL减1, 若TTL变为0, 则丢弃, 并发送时间超过报文给源点

- 当终点在预先规定的时间内不能收到一个数据报的全部数据报片时，就把已收到的数据报片都丢弃，并发送时间超过报文给源点
- 参数问题
 - 路由器或主机收到IP数据报，根据首部的检验和字段发现首部出现误码，丢弃数据报并发送参数问题报文给源点
- 改变路由（重定向）
 - 路由器把改变路由报文发送给主机，让主机知道下次应将数据报发送给另外的路由器（可通过更好的路由）

以下情况不应发送ICMP差错报告报文

- 对ICMP差错报告报文
- 对第一个分片的数据报片的所有后续数据报片
- 对具有多播地址的数据报
- 对具有特殊地址（127.0.0.0、0.0.0.0）的数据报
- ICMP询问报文
 - 回送请求和回答
 - 会送请求报文是由主机或路由器向一个特定的目的主机发出的询问
 - 收到此报文的主机必须给源主机或路由器发送ICMP回送回答报文
 - 这种询问报文用来测试目的站是否可达及了解有关状态
 - 时间戳请求和回答
 - 时间戳请求报文是请某个主机或路由器回答当前的日期和时间
 - 用来进行时间同步和测量时间
- ICMP应用举例
 - 分组网间探测PING
 - 用来测试主机或路由器间的连通性
 - 应用层直接使用网际层的ICMP（没有通过运输层的TCP/UDP）
 - 使用ICMP回送请求和回答报文
 - 跟踪路由
 - 用来测试IP数据报从源主机到达目的主机要经过哪些路由器
 - Windows版本
 - tracert命令
 - 应用层直接使用网际层ICMP
 - 使用ICMP回送请求和回答报文及差错报告报文
 - Unix版本
 - traceroute命令
 - 在运输层使用UDP协议
 - 仅使用ICMP差错报告报文

4.9 虚拟专用网VPN与网络地址转换NAT

- VPN
 - 利用公用的因特网作为本机构各专用网之间的通信载体，这样的专用网又称为虚拟专用网
 - 由于IPv4地址紧缺，一个机构能够申请到的IPv4地址数量有限。因此，VPN中各主机所分配的地址应该是本机构可自由分配的专用地址，而不是需要申请的在因特网上使用的公有地址

- 专用（私有）地址
 - 10/8地址块
 - 172.16/12地址块
 - 192.168/16地址块
- 同一机构内不同部门的内部网络所构成的虚拟专用网VPN又称为内联网VPN
- 有时一个机构的VPN需要某些外部机构参加，就称为外联网VPN
- 远程接入VPN：在外地访问某机构的专用网络，只需接入因特网，运行VPN软件，在本机和机构的主机之间建立VPN隧道，就可访问专用网络中的资源
- NAT
 - 缓解了IPv4地址空间即将耗尽的问题
 - 能使大量使用内部专用地址的专用网络用户共享少量外部全球地址来访问因特网上的主机和资源
 - NAT路由器：将私有地址转换为公有地址
 1. 修改IP数据报的源地址为全球IP地址
 2. 记录私有地址与公有地址的关系
 3. 转发IP数据报
 - 存在问题：如果NAT路由器具有N个全球IP地址，那么最多只能有N个内网主机同时与因特网上的主机通信
 - 解决：由于绝大多数网络应用都是使用运输层协议TCP/UDP进行传送数据，因此可以利用运输层的端口号和IP地址一起进行转换。
这样，用一个全球IP地址就可以使多个拥有本地地址的主机同时和因特网上的主机进行通信。这种将端口号和IP地址一起进行转换的技术叫做网络地址与端口号转换NAPT
 - 外网主机不能首先发起通信
 - 对于一些P2P网络应用，需要外网主机主动与内网主机进行通信，在通过NAT时会遇到问题，需要网络应用自己使用一些特殊的NAT穿越技术解决
 - 由于NAT对外网屏蔽了内网主机的网络地址，能为内网的主机提供一定的安全保护

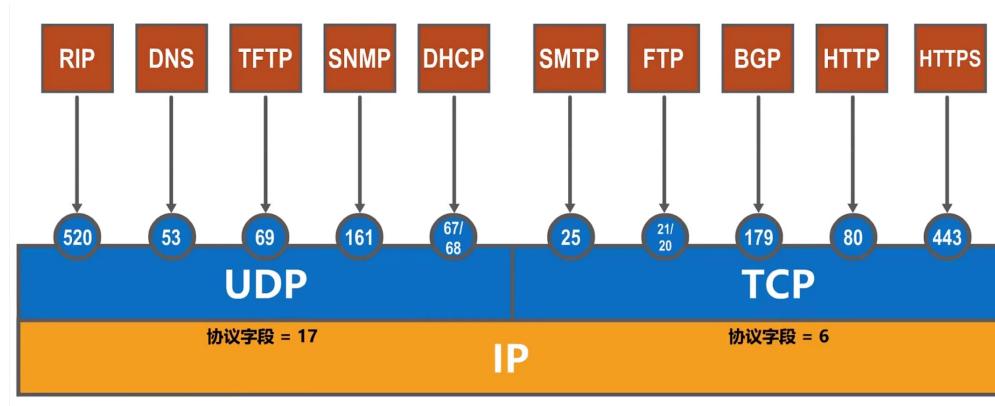
5 运输层

5.1 运输层概述

- 运输层直接为应用进程间的逻辑通信提供服务
- 运输层向高层用户屏蔽了网络核心的细节（网络拓扑、路由选择协议）。
- 根据应用需求不同，运输层为应用层提供两种不同的运输协议
 - 面向连接的TCP
 - 无连接的UDP

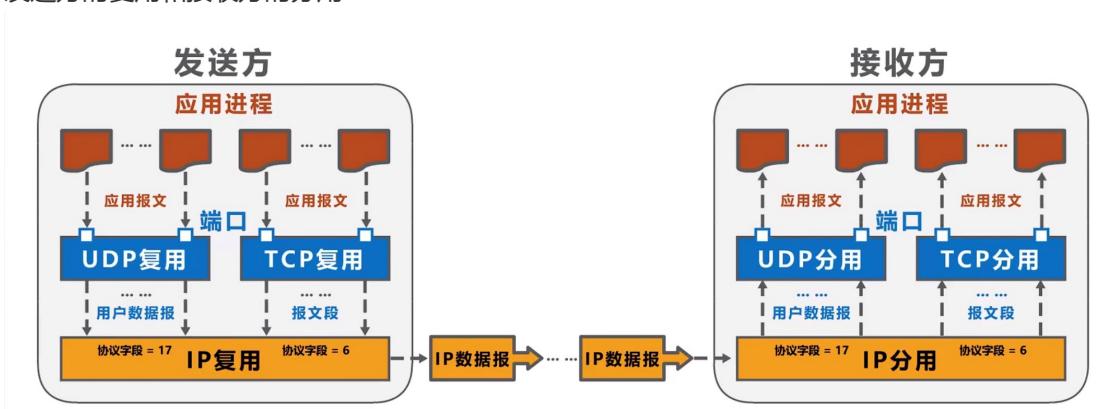
5.2 运输层端口号、复用与分用的概念

- 运行在计算机上的进程使用进程标识符PID来标志
- 因特网上的计算机并不是统一使用的操作系统，不同的操作系统使用不同格式的进程标识符
- 为使运行不同操作系统的计算机应用进程之间能够进行网络通信，就必须使用统一的方法对TCP/IP体系的应用程序进行标识
- TCP/IP体系的运输层使用端口号来区分应用层的不同应用进程
 - 端口号使用16bit，取值范围是0-65535
 - 熟知端口号：0-1023，IANA把这些端口号指派给了TCP/IP中最重要的一些应用协议



如，FTP-21/20，HTTP-80，DNS-53

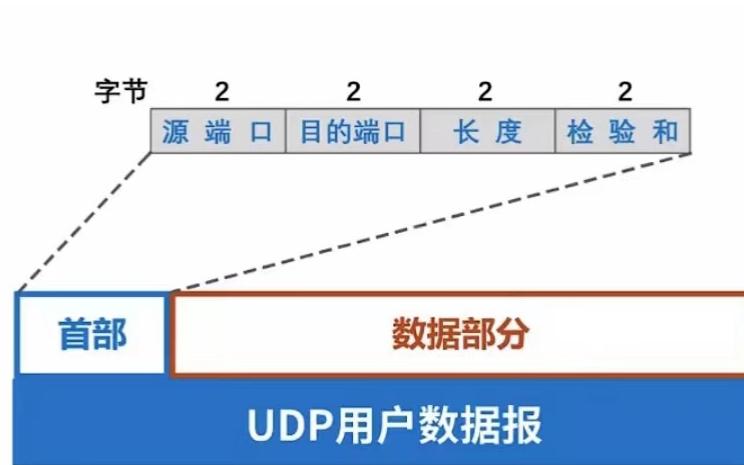
- 登记端口号：1024-49151，为没有熟知端口号的应用程序使用。使用这类端口号必须在IANA按照规定的手续登记，以防止重复
- 短暂端口号：49152-65535，留给客户进程选择暂时使用。当服务器进程收到客户进程的报文时，就知道了客户进程所使用的动态端口号。通信结束后，这个端口号可以给其他客户进程使用。
 - 端口号只有本地意义，即端口号只是为了标识本机应用层中的各进程，在因特网中，不同计算机的相同端口是没有联系的
- 发送方的复用和接收方的分用



5.3 UDP和TCP的对比

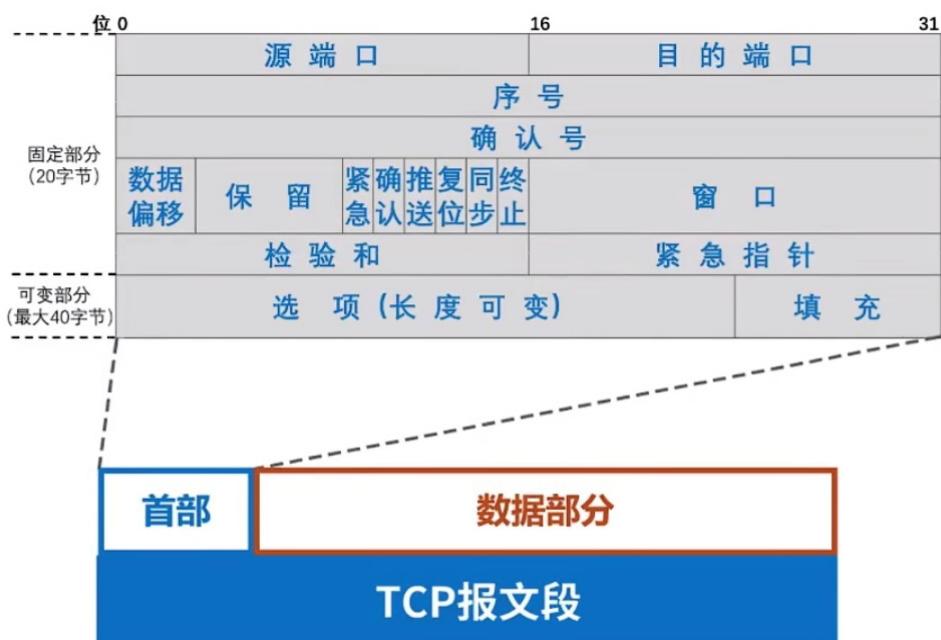
- 用户数据报协议UDP
 - 无连接
 - 支持单播、多播、广播
 - 面向应用报文

- 向上提供无连接不可靠传输服务（适用于IP电话、视频会议等应用）
- 首部格式



UDP用户数据报头部仅8字节

- 传输控制协议TCP
- 面向连接
- “三报文”握手建立可靠连接，“四报文”握手释放连接
- 仅支持单播
- 面向字节流
- 向上提供面向连接的可靠传输服务（适用于文件传输等）
- 首部格式



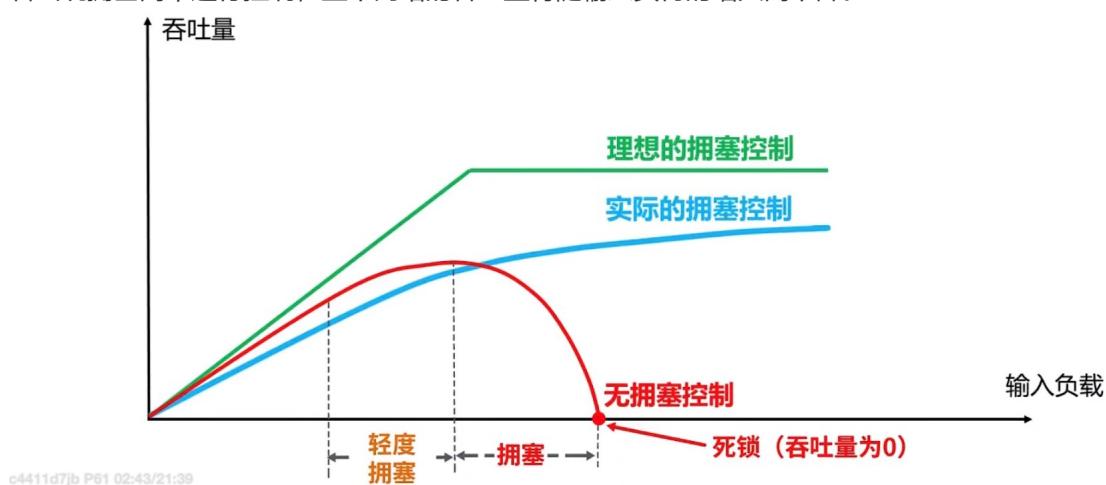
TCP报文段首部最小20字节，最大60字节

5.4 TCP流量控制

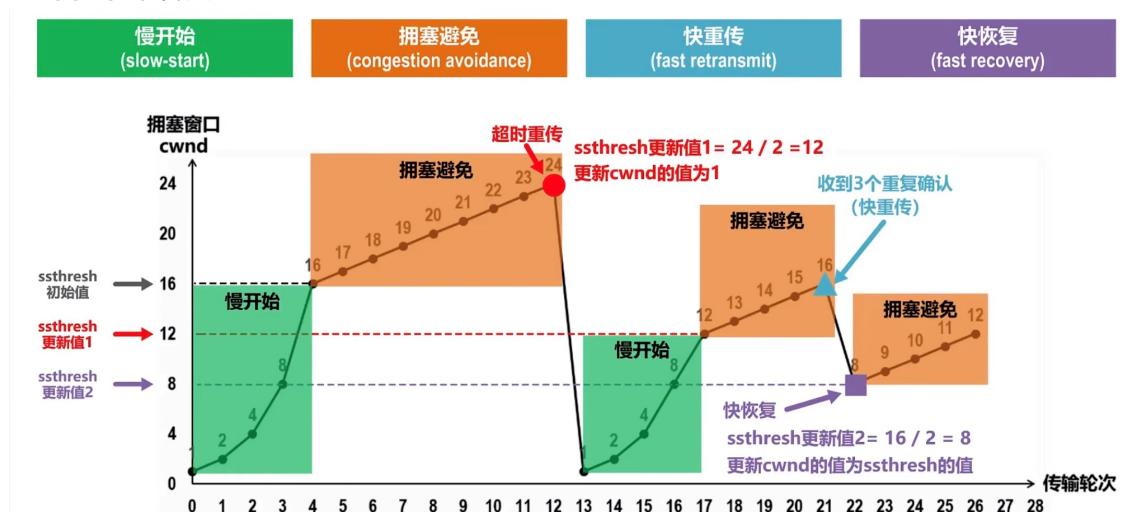
- 流量控制：让发送方的发送速度不要太快，要让接收方及时接收
- 利用滑动窗口机制在TCP连接上实现发送方流量控制
 - TCP接收方利用自己的接收窗口的大小来限制发送方发送窗口的大小
 - TCP发送方收到接收方的零窗口通知后，应启动持续计时器。持续计时器超时后，向接收方发送零窗口探测报文

5.5 TCP拥塞控制

- 拥塞：在某段时间，若对网络中某一资源的需求超过了该资源所能提供的可用部分，网络性能就要变坏。
 - 在计算机网络中的链路容量（带宽）、交换结点中缓存和处理机等，都是网络资源
- 若出现拥塞而不进行控制，整个网络的吞吐量将随输入负载的增大而下降。



- TCP拥塞控制算法



假定：

- 数据单方向传送，而另一个方向只传送确认
- 接收方有足够大的缓存空间，因而发送方发送窗口的大小由网络的拥塞程度决定
- 以最大报文段MSS的个数为讨论问题的单位，而不是以字节为单位
- 慢开始和拥塞避免算法 (TCP Tahoe版本)
 - 发送方维护一个叫做拥塞窗口cwnd的状态变量，其值取决于网络的拥塞程度，且动态变化

- 拥塞窗口的维护原则：只要网络没有出现拥塞，拥塞窗口就再增大一些；但只要网络出现拥塞，拥塞窗口就减少一些
- 判断出现网络拥塞的依据：没有按时收到应当到达的确认报文（发生超时重传）
- 发送方将拥塞窗口作为发送窗口swnd，即 $swnd = cwnd$
- 维护一个慢开始门限ssthresh状态变量：
 - 当 $cwnd < ssthresh$ 时，使用慢开始算法
 - 当 $cwnd > ssthresh$ 时，停止使用慢开始算法而改用拥塞避免算法
 - 当 $cwnd = ssthresh$ 时，既可使用慢开始算法，也可使用拥塞避免算法
- 慢开始阶段
 - 拥塞窗口指数增长
- 拥塞避免阶段
 - 拥塞窗口线性增长
 - 当重传计时器超时，判断网络可能出现拥塞，进行以下工作
 1. 将ssthresh值更新为发生拥塞时cwnd值得一半
 2. 将cwnd减少为1，并重新开始执行慢开始算法
- 快重传和快恢复算法（TCP Reno版本）
 - 让发送方尽早知道发生了个别报文得丢失
 - 快重传：使发送方尽快进行重传，而不是等超时重传计时器超时再重传
 - 要求接收方不要等待自己发送数据时才捎带确认，而是要立即发送确认
 - 即使收到了失序的报文段也要立即发出对已收到报文段的重复确认
 - 发送方一旦收到3个连续的重复确认，就将相应的报文段立即重传，而不是等该报文段的超时重传计时器超时再重传
 - 快恢复：发送方一旦收到3个重复确认，就知道丢失了个别报文段。于是不启动慢开始算法，而执行快恢复苏算法
 - 发送方将慢开始门限ssthresh和拥塞窗口cwnd调整为当前窗口的一般；开始执行拥塞避免算法
 - 也有的快恢复实现是把快恢复开始时的拥塞窗口cwnd值再增大一些，即等于新的 $ssthresh + 3$
 - 既然发送方收到3个重复确认，就表明有3个数据报文段已经离开了网络
 - 这3个报文段不再消耗网络资源，而是停留在接收方的接收缓存中；因此可以适当增大cwnd

5.6 TCP超时重传时间选择

- 超时重传时间RTO应略大于往返时间RTT
- 不能直接使用某次测量得到的RTT样本来计算超时重传时间RTO
- 利用每次测量得到的RTT样本，计算加权平均往返时间RTT（又称为平滑的往返时间）

$$RTT_{S1} = RTT_1$$

$$\text{新的}RTT_S = (1 - \alpha) \times \text{旧的}RTT_S + \alpha \times \text{新的}RTT\text{样本}$$

- 若 α 接近0，则新的RTT样本对RTTs的影响不大
- 若 α 接近1，则新的RTT样本对RTTs的影响很大
- 标准推荐的 α 值为0.125
- 利用每次测量得到的RTT样本，计算RTT偏差的加权平均RTT_D

$$RTT_{D1} = RTT_1 \div 2$$

新的 $RTT_D = (1 - \beta) \times \text{旧的}RTT_D + \beta \times |RTT_S - \text{新的}RTT\text{样本}|$

- 标准推荐 β 值为0.25
- 综合上述计算超时重传时间RTO

$$RTO = RTT_S + 4 \times RTT_D$$

- 往返时间RTT的测量比较复杂
 - 针对出现超时重传时无法测准RTT的问题，使用Karn算法：
 - 在计算加权平均往返时间RTTs时，只要报文重传了，就不采用其往返时间RTT样本，RTO不会重新计算
 - Karn算法修正：
 - 报文段每重传依次，就把超时重传时间RTO增大一些。典型的做法是将新RTP的值取为原来的2倍。

5.7 TCP可靠传输实现

- TCP基于以字节为单位的滑动窗口来实现可靠传输
 - 发送窗口
 - 后沿
 - 不动（没有收到新的确认）
 - 前移（收到了新的确认）
 - 前沿
 - 通常不断向前移动
 - 不动：
 - 没有收到新的确认，对方通知的窗口大小也不变
 - 收到新的确认但对方通知的窗口缩小，使得发送窗口正好前沿不动
 - 向后收缩（对方通知的窗口缩小）
 - 状态描述



如何描述发送窗口的状态？

使用三个指针P1, P2, P3分别指向相应的字节序号

- 小于P1的是已发送并已收到确认的部分；
- 大于等于P3的是不允许发送的部分；
- $P3 - P1 = \text{发送窗口的尺寸}$ ；
- $P2 - P1 = \text{已发送但尚未收到确认的字节数}$ ；
- $P3 - P2 = \text{允许发送但当前尚未发送的字节数（又称为可用窗口或有效窗口）}$ ；

6:06/18:04

- 接收窗口
- 在同一时刻，发送方的发送窗口并不总是和接收方的接收窗口一样大
 - 网络传送窗口值需要经历一定时间滞后

- 发送方可能根据网络当时的拥塞情况适当减小自己的发送窗口尺寸
- 对于不按序到达的数据的处理
 - 如果接收方对其一律丢弃，那么接收窗口的管理较为简单，但这样做对网络资源利用率不利
 - TCP通常对不按序到达的数据先临时存放在接收窗口中，等到字节流中所缺少的字节收到后，再按序交付上层的应用进程
- TCP要求接收方必须有累积确认和捎带确认机制，可以减少传输开销
 - 接收方不应过分推迟发送确认 ($\leq 0.5s$)
 - 携带确认实际上并不经常发生
- TCP的通信是全双工通信。通信中的每一方都在发送和接收报文段

5.8 TCP的运输连接管理

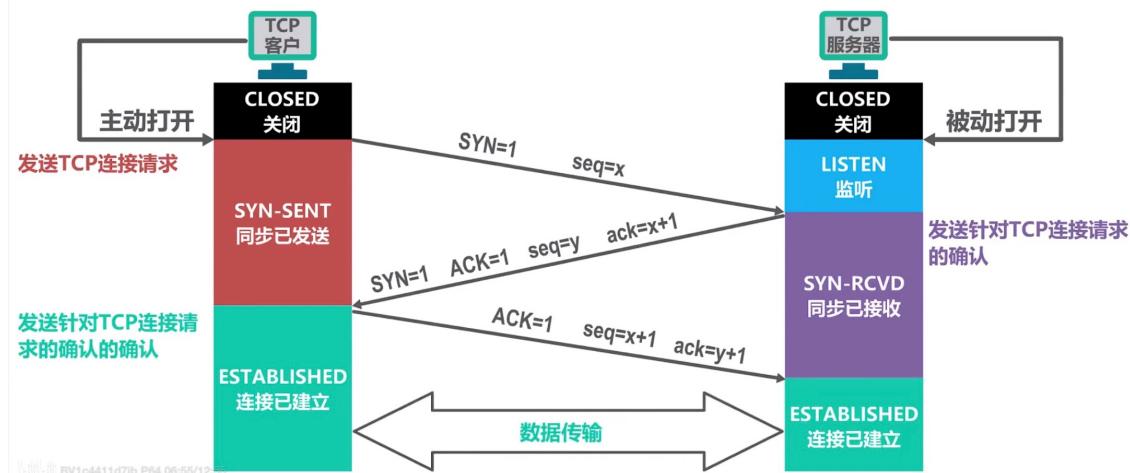
- TCP运输连接管理
 - TCP是面向连接的协议，它基于运输连接来传送报文段
 - TCP运输连接三个阶段：
 1. 建立TCP连接
 2. 数据传送
 3. 释放TCP连接
 - TCP运输连接管理就是使TCP连接的建立和释放都能正常运行

Alice代指TCP客户进程， Bob代指TCP服务器进程

TCP客户进程主动打开连接，主动关闭连接

TCP服务器进程被动打开连接，被动关闭连接

- TCP的连接建立



- "三报文握手"建立连接

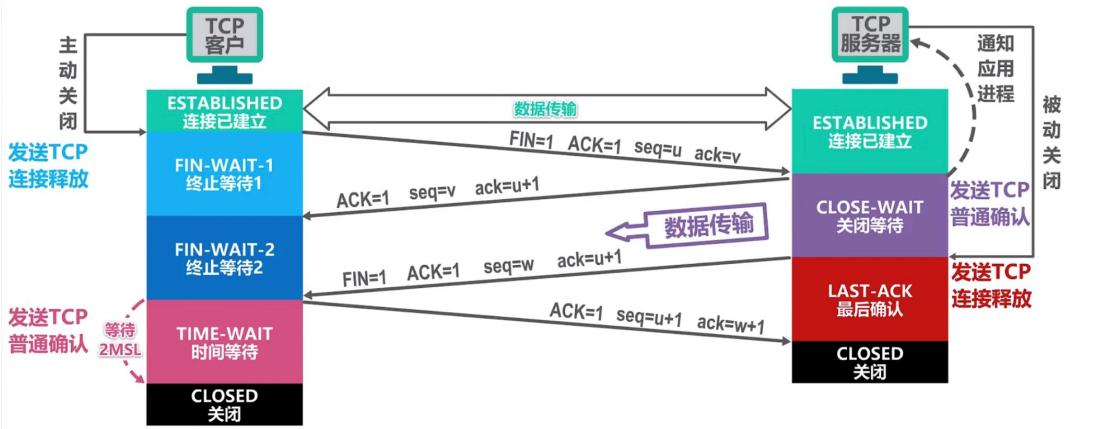
1. Alice发送TCP连接请求
 - `SYN = 1, seq = x`
2. Bob发送针对TCP连接请求的确认
 - `SYN = 1, ACK = 1, seq = y, ack = x+1`
3. Alice发送针对TCP连接请求的确认的确认
 - `ACK = 1, seq = x+1, ack = y+1`

普通的确认报文段可以携带数据，但如果不能携带数据，则不消耗序号，seq仍为x

- 不使用“两次握手”原因是：

- 防止已失效的连接请求报文段突然又传到了Bob，因而导致错误

- TCP连接释放



- “四报文挥手”释放连接

1. Alice发送TCP连接释放

- FIN = 1, ACK = 1, seq = u, ack = v**

u等于TCP客户进程之间已传送的数据的最后一个字节的序号+1；TCP规定终上位FIN=1的报文段即使不携带数据，也要消耗掉一个序号

v等于TCP客户进程之前已收到的数据的最后一个字节的序号+1

2. Bob发送TCP普通确认

- ACK = 1, seq = v, ack = u+1**

v等于TCP服务器进程之前已传送的数据的最后一个字节的序号+1（与上一条的v匹配）

- 此时Bob到Alice方向的连接并未关闭

3. Bob发送TCP释放连接

- FIN = 1, ACK = 1, seq = w, ack = u+1**

4. Alice发送TCP普通确认

- ACK = 1, seq = u+1, ack = w+1**

5. Alice等待2MSL时间后 (MSL是最长报文寿命，建议为2分钟)

- TCP保活计时器

- TCP服务器进程每次收到一次TCP客户进程的数据，就重新设置并启动保活计时器（2小时定时）
- 若保活计时器定时周期内未收到TCP客户进程发来的数据，则到时后，TCP服务器进程就向TCP客户进程发送一个探测报文段，以后每隔75秒发送一次，若连续发送10个探测报文段后仍无响应则认为TCP客户端发生故障，关闭连接

5.9 TCP报文段的首部格式

- 一个TCP报文段由首部和数据载荷两部分构成；
- TCP全部功能体现在它首部中各字段的作用
- 首部格式



- 源端口
 - 占16bit, 写入源端口号, 用来标识发送该TCP报文段的应用进程
 - 目的端口
 - 占16bit, 写入目的端口号, 用来标识接收该TCP报文段的应用进程
 - 序号
 - 占32bit, 取值[0, $2^{32}-1$], 序号增加到最后一个后, 下一个序号就又回到0
 - 指出本TCP报文段数据载荷的第一个字节的序号
 - 确认号
 - 占32bit, 取值[0, $2^{32}-1$], 序号增加到最后一个后, 下一个序号就又回到0
 - 指出期望收到对方下一个TCP报文段的数据载荷的第一个字节的序号, 同时特使对之前收到的所有数据的确认

若确认号= n, 则表明到序号n-1为之的所有数据都已正确接收, 期望接收序号为n的数据
 - 确认标志位ACK
 - 取值为1时, 确认号有效; 取值为0时, 确认号无效
 - 在连接建立后的所有传送的TCP报文段都必须把ACK置1
 - 数据偏移
 - 占4bit, 以4byte为单位
 - 用来指出TCP报文段的数据载荷部分的起始处距离TCP报文段的起始处有多远
 - 这个字段实际上是指出了TCP报文段的首部长度
 - 首部固定长度为20byte, 因此数据偏移字段的最小值为(0101)₂
 - 首部最大长度为60byte, 因此数据偏移字段的最大值为(1111)₂
 - 保留
 - 占6bit, 保留为今后使用, 目前置0
 - 窗口
 - 占16bit, 以字节为单位。指出发送本报文段的一方的接收窗口
 - 窗口值作为接收方让发送方设置其发送窗口的依据, 这是以接收方的接收能力来控制发送方的发送能力 (流量控制)
- 发送窗口 = Min[接收窗口, 拥塞窗口]
- 校验和

- 占16bit，检查范围包括首部和数据载荷
- 在计算校验和时，要在TCP报文段前面加上12byte的伪首部
- 同步标志位SYN
 - 在TCP连接时用来同步序号
- 终止标志位FIN
 - 用来释放TCP连接
- 复位标志位RST
 - 用来复位TCP连接
 - RST=1时，表明TCP连接异常，需要重新建立连接
 - RST置1还用来拒绝一个非法的报文段或拒绝打开一个TCP连接
- 推送标志位PSH
 - 接收方的TCP收到PSH=1的报文段会尽快上交应用进程，而不必等到接收缓存都填满后再向上交付
- 紧急标志位URG
 - 取值为1时紧急指针有效；取值为0时紧急指针无效
- 紧急指针
 - 占16bit，以字节为单位，用来指明紧急数据的长度
 - 当发送方有紧急数据时，可将紧急数据插队排到发送缓存的最前面，并立刻封装到一个TCP报文段中进行发送。紧急指针指出本报文段数据载荷部分包含了多长的紧急数据
- 选项
 - 最大报文段长度MSS选项：TCP报文段数据载荷部分最大长度
 - 窗口最大选项：为了扩大窗口（提高吞吐率）
 - 时间戳选项
 - 计算往返时间RTT
 - 用于处理序号超范围的情况，又称为防止序号绕回PAWS
 - 选择确认选项
- 填充
 - 保证报文段首部被4整除

6 应用层

6.1 应用层概述

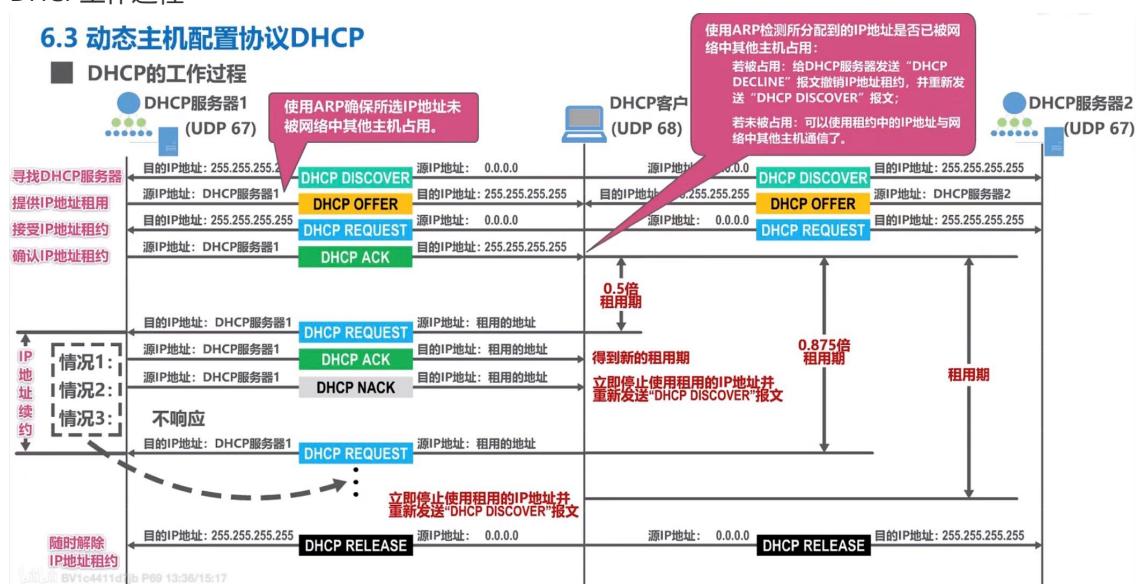
- 最顶层，是设计和建立计算机网络的最终目的
- 网络应用
 - 域名系统DNS
 - 万维网WWW
 - 动态主机配置DHCP
 - 电子邮件
 - 文件传送FTP和P2P文件共享
 - 多媒体网络应用

6.2 客户/服务器方式和对等方式

- 网络应用程序在各种端系统上的组织方式和它们之间的关系
 - 客户/服务器 (C/S) 方式
 - 客户和服务器是指通信中涉及的两个应用进程
 - 客户/服务器方式所描述的是进程之间的服务和被服务的关系
 - 客户是服务的请求方，服务器是服务提供方
 - 服务器总是处于运行状态，并等待客户的服务请求。服务器具有固定端口号（如Http-80），而运行服务器的主机也具有固定的IP地址
 - C/S方式网络应用：万维网、电子邮件、文件传输
 - 基于C/S的网络应用都是服务集中型的，即应用服务集中在网络中比客户计算机少得多的服务器计算机上
 - 由于一台服务器计算机为多个客户机提供服务，常会出现服务器计算机跟不上客户机请求的情况；
 - 因此，常用计算机群集（服务器场）构建一个强大的虚拟服务器
 - 对等 (P2P) 方式
 - 没有固定的服务请求者和服务提供者，网络应用进程之间是对等的，称为对等方。对等方之间直接通信
 - P2P方式网络应用：即时通信、P2P文件共享、P2P流媒体、分布式存储
 - 基于P2P的应用是服务分散型的
 - 可扩展性：系统性能不会因为规模的增大而降低
 - 具有成本上的优势

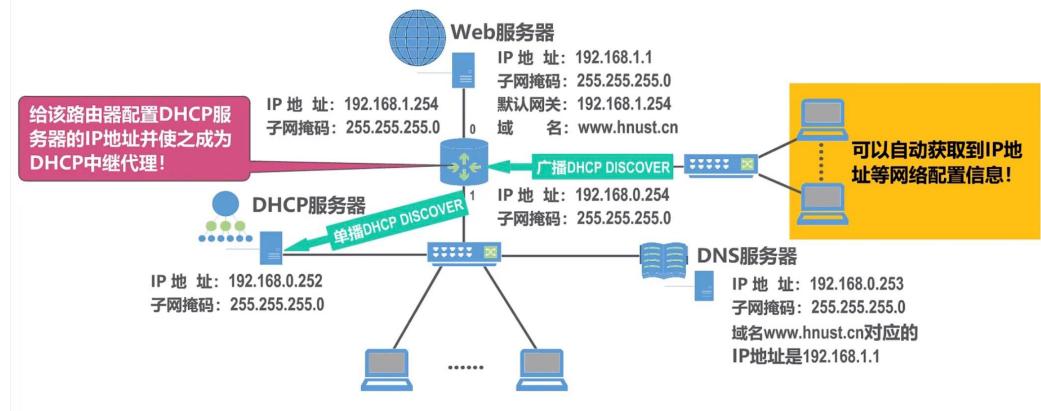
6.3 动态主机配置协议DHCP

- DHCP的作用
 - 各主机可以通过DHCP服务器自动获取网络配置信息，而不需要手工配置
- DHCP工作过程



- DHCP中继代理

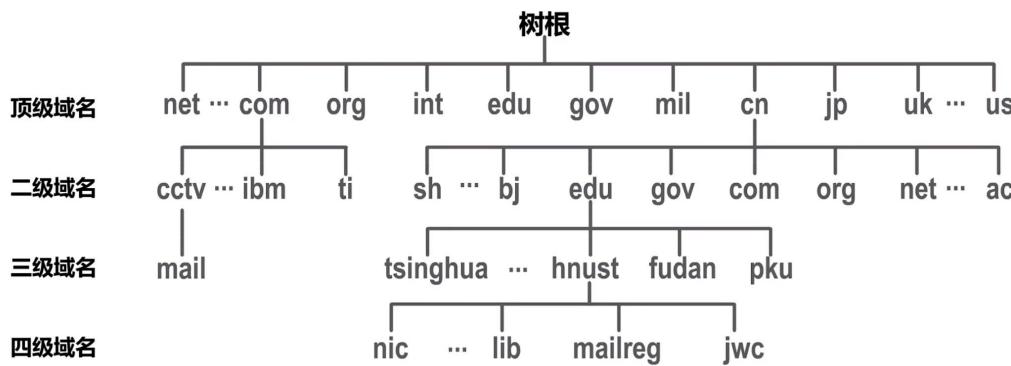
- 不需要再每一个网络上都配置一个DHCP服务器



6.4 域名系统DNS

- DNS作用
 - 将域名解析为IP地址
- 不只使用一台DNS服务器
 - 因特网采用层次结构的命名树作为域名，并使用分布式的DNS
 - DNS使大多数域名都在本地解析，仅少量解析需要在因特网上通信
 - 即使某个DNS服务器故障，也不会影响域名系统正常运行
- 层次树状结构的域名结构
 - 域名结构由若干个分量组成，各分量用点隔开，代表不同级别的域名
 - `.三级域名.二级域名.顶级域名`
 - 每一级的域名都由英文字母和数字组成，不超过63个字符，不区分大小写字母
 - 完整的域名不超过255个字符
 - 域名系统既不规定一个域名包含多少个下级域名，也不规定每一级的域名代表什么意思
 - 各级域名由上一级的域名管理机构管理，而最高的顶级域名则由因特网名称与数字地址分配机构ICANN进行管理
 - 顶级域名TLD
 - 国家顶级域名nTLD
 - `.cn`-中国, `.us`-美国, `.uk`-英国
 - 通用顶级域名gTLD
 - `.com`-公司企业, `.net`-网络服务机构, `.org`-非营利性组织, `.int`-国际组织, `.edu`-美国教育机构, `.gov`-美国政府部门, `.mil`-美国军事部门
 - 反向域arpa
 - 用于反向域名解析，即IP地址反向解析为域名
 - 在国家顶级域名下注册的二级域名均由该国家自行确定
 - 我国二级域名
 - 类别域名
 - `.ac`-科研机构, `.com`-企业, `.edu`-教育机构, `.gov`-政府部门, `.net`-网络服务机构, `.mil`-军事机构, `.org`-非营利性组织
 - 行政区域名
 - 共34个

- 因特网域名空间



这种按等级管理的命名方法便于维护名字的唯一性，并且也容易设计出一种高效的域名查询机制。需要注意的是，域名只是个逻辑概念，并不代表计算机所在的物理地点。

- 分布式域名服务器

- 根域名服务器

- 最高层次的域名服务器。每个根域名服务器都知道所有顶级域名服务器的域名及其IP地址。因特网上有13个不同IP地址的根域名服务器，每台服务器实际上是由许多分布在世界各地的计算机构成的服务器集群
 - 根域名服务器通常并不直接对域名进行解析，而是返回该域名所属顶级域名的顶级域名服务器的IP地址

- 顶级域名服务器

- 负责管理该顶级域名服务器注册的所有二级域名。

- 权限域名服务器

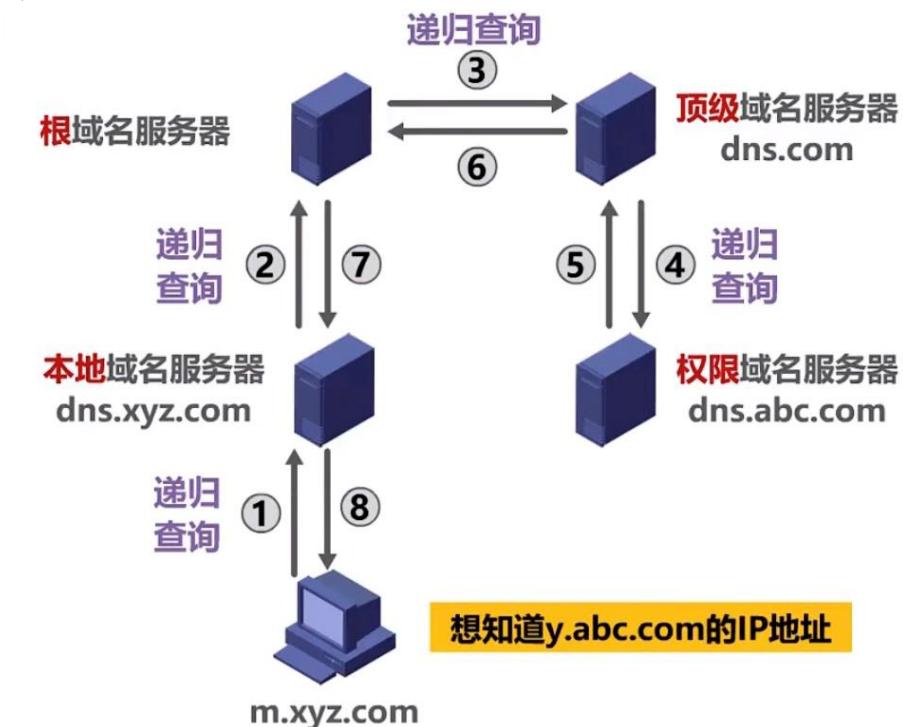
- 负责管理某个区的域名。每个主机的域名都必须在某个权限域名服务器处注册登记。因此权限域名服务器知道其管辖的域名与IP地址的映射关系，也知道其下级域名服务器的地址

- 本地域名服务器

- 不属于上述域名服务器的等级结构。当一个主机发出DNS请求报文时，这个报文就首先被送往该主机的本地域名服务器。
 - 本地域名服务器起着代理的作用，会将该报文转发到上述的域名服务器的等级结构中
 - 每一个因特网服务提供者ISP，一个大学或者一个学院，都可以拥有一个本地域名服务器，也称为默认域名服务器
 - 本地域名服务器距离用户较近，一般不超过几个路由器，其IP地址需要直接配置在需要域名解析的主机中

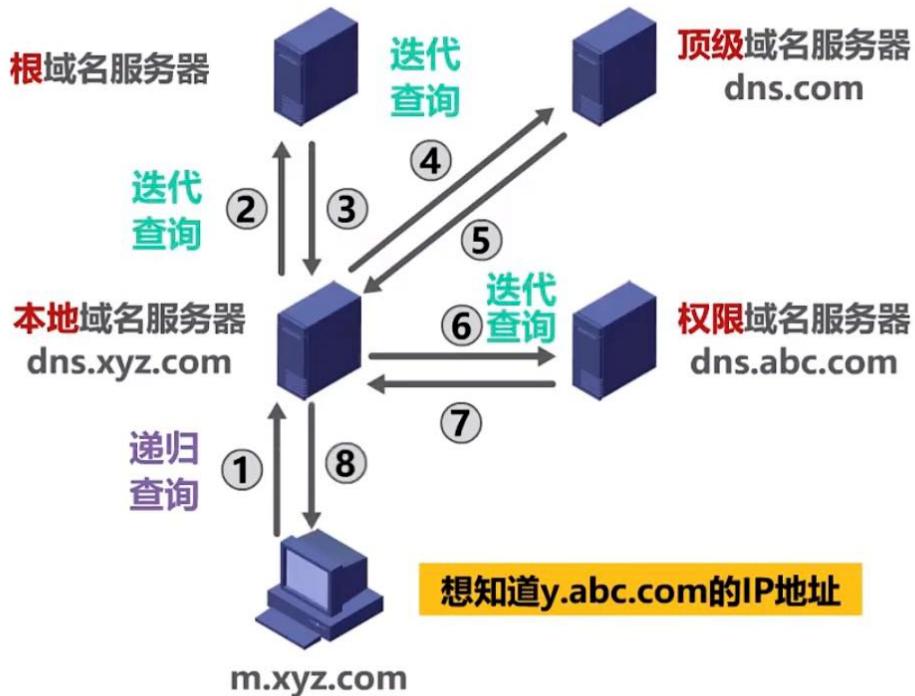
- 域名解析的过程

- 递归查询



Bilibili BV1c4411d7jb P70 15:05/20:17

- 迭代查询

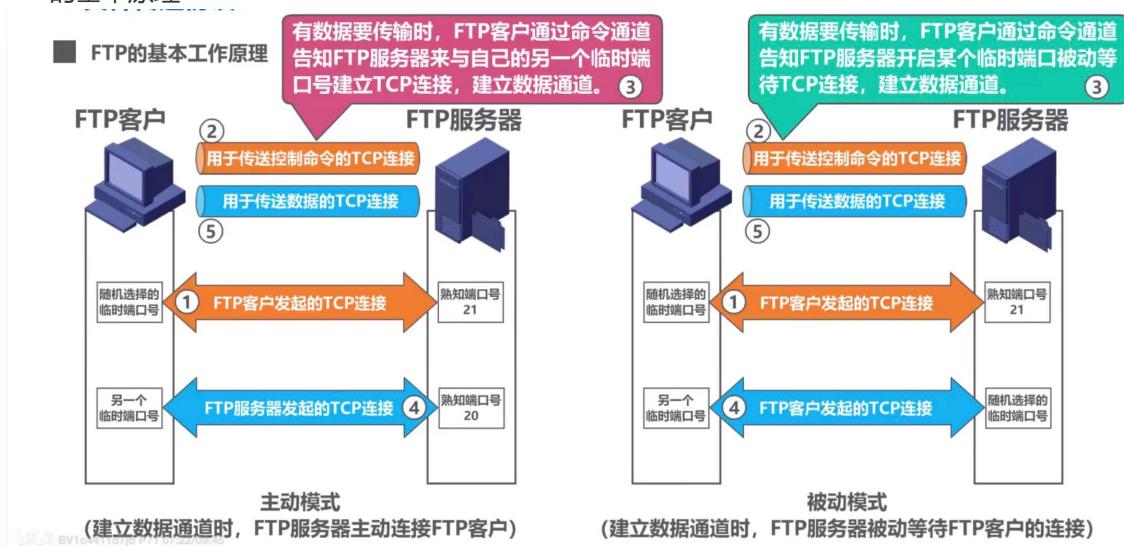


- 由于递归查询对于被查询的域名服务器负担太大，通常采用：
 - 从请求主机到本地域名的查询是递归查询；而其余查询都是迭代查询
- 在域名服务器中广泛使用高速缓存
 - 高速缓存存放最近查询过的域名以及从何处获得域名映射信息的记录
 - 由于域名到IP地址的映射关系并不是永久不变，为保持高速缓存中的内容正确，域名服务器应为每项内容设置计时器闭关删除超过合理时间的项

- 用户主机中也维护自己最近使用的域名的高速缓存。同理，主机也要保持高速缓存中内容的正确性
- DNS报文使用运输层的UDP协议进行封装，运输层端口号为53

6.5 文件传送协议FTP

- 文件传送：将某台计算机中的文件通过网络传送到相距很远的另一台计算机中
- 文件传送协议DTP
 - 提供交互式的访问，允许客户指明文件的类型与格式，并允许文件具有存取权限
 - 屏蔽了各计算机系统的细节，适用于在异构网络中任意计算机之间传送文件
- FTP常见用途
 - 在计算机之间传输文件，尤其是批量传输
 - 让网站设计者将构成网站内容的大量文件批量上传到Web服务器
- FTP的基本原理



- 主动模式：传输数据时，FTP服务器主动连接FTP客户
 1. FTP客户发起TCP连接
 2. 建立其传送控制命令的命令通道
 3. 有数据要传输时，FTP客户通过命令通道告知FTP服务器来与自己的另一个临时端口建立TCP连接
 4. FTP服务器发起TCP连接
 5. 建立数据通道
- 被动模式：传输数据时，FTP服务器被动等待FTP客户的连接
 1. FTP客户发起TCP连接
 2. 建立其传送控制命令的命令通道
 3. 有数据要传输时，FTP客户通过命令通道告知FTP服务器开启某个临时端口被动等待TCP连接
 4. FTP客户发起TCP连接
 5. 建立数据通道
- FTP客户和服务器之间要建立两个并行的TCP连接
 - 控制连接在整个会话期间一直保持打开，用于传送FTP相关控制命令
 - 数据连接用于文件传输，在每次文件传输时才建立，传输结束就关闭
 - 默认情况下，FTP使用TCP 21端口进行控制连接，TCP 20端口进行数据连接。但是，是否使用TCP20端口建立数据连接与传输模式有关，主动方式使用TCP20端口，被动方式

由服务器和客户端自行协商决定

6.6 电子邮件

- 电子邮件概述
 1. 发件人将邮件发送到自己使用的邮件服务器
 2. 邮件服务器将收到的邮件按照其目的地址转发到收件人邮件服务器中的收件人邮箱
 3. 收件人在方便的时候访问收件人邮件服务器中自己的邮箱，获取电子邮件
- 电子邮件系统
 - 采用C/S方式
 - 三个主要组成构件
 - 用户代理
 - 是用户与电子邮件系统的接口，又称为电子邮件客户端软件
 - 邮件服务器
 - 是电子邮件系统的基础设施。功能是发送和接收邮件、以及负责维护用户的邮箱
 - 电子邮件所需的协议
 - 包括邮件发送协议（SMTP）和邮件读取协议（POP3、IMAP）
 - 邮件发送和接收过程



- 邮件发送协议
 - 简单邮件传送协议SMTP
 - 基本工作原理

周期性对邮件缓存扫描
(例如, 30分钟)
如发现有邮件

客户端向服务器说明身份, 告知自己SMTP服务器的域名

客户端告诉服务器邮件来自何方

客户端告诉服务器邮件去往何地

客户端告诉服务器自己准备发送邮件内容

客户端向服务器发送邮件内容

客户端发送完邮件内容后, 还要发送结束符

客户端向服务器请求断开连接

主动推送服务就绪应答

若身份有效, 发回应答代码250

若合理, 发回应答代码250; 否则, 发回其他错误代码

若该邮箱存在, 发回应答代码250; 否则, 发回其他错误代码

若准备就绪, 发回应答代码354; 否则, 发回其他错误代码

若收件成功, 发回应答代码250; 否则, 发回其他错误代码

发回应答代码221表示接受请求并主动断开连接
 - SMTP协议只能传送ASCII码文本数据, 不能传送可执行文件或其他的二进制对象; 不能满足多媒体邮件或非英语文字的需要
 - 电子邮件的信息格式
 - 信封
 - 内容
 - 首部

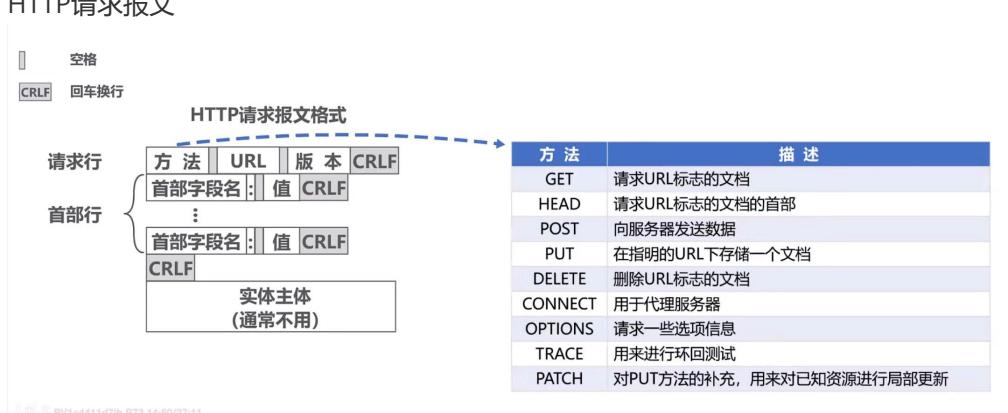
| | |
|---|--------------|
| 1 | From:(发送人) |
| 2 | To:(接收人) |
| 3 | Cc:(抄送人) |
| 4 | Subject:(主题) |

- 主体
 - 多用途因特网邮件扩展MIME
 - 将非ASCII码转换为ASCII码
 - 增加了5个新的邮件首部字段，提供了有关邮件主体的信息
 - 定义了许多邮件内容的格式，对多媒体电子邮件的表示方法进行了标准化
 - 定义了传送编码，可对任何内容格式进行转换
 - 不仅用于SMTP，也用于HTTP
 - 邮件读取协议
 - 邮局协议POP
 - 简单功能有限
 - 用户只能以下载并删除方式或下载并保留方式从邮件服务器下载邮件到用户方计算机。
不允许用户在邮件服务器上管理自己的邮件（如分类管理等）
 - 因特网邮件访问协议IMAP
 - 用户在自己的计算机上就可以操控邮件服务器中的邮箱
 - 基于万维网的电子邮件
 - 通过浏览器登录邮件服务器万维网网站就可以撰写、收发、阅读和管理电子邮件
 - 用户计算机无需安装专门的用户代理程序，只需要使用通用的万维网浏览器
 - 使用了HTTP协议

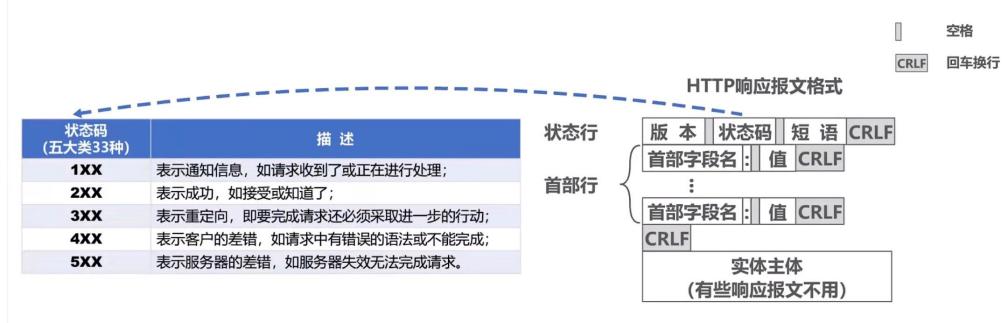
6.7 万维网WWW

- WWW并非某种特殊的计算机网络。它是一个大规模的、联机式的信息储藏所，是运行在因特网上的一个分布式应用
- 利用网页之间的超链接将不同网站的网页链接成一张逻辑上的信息网
- 浏览器
 - 渲染引擎（浏览器内核）：负责对网页内容进行解析和显示
 - 不同的浏览器内核对网页内容的解析不同，因此同一网页在不同内核的浏览器的显示效果可能不同
 - 网页编写者需要在不同内核的浏览器中测试网页显示效果
- 统一资源定位符URL
 - 为了方便地访问世界范围内的文档，万维网使用统一资源定位符URL来指明因特网上的任何种类资源的位置
 - URL一般形式：
`<协议>://<主机>:<端口>/<路径>`
- 万维网文档
 - html：超文本标记语言
 - css：层叠样式表
 - javascript：一种脚本语言
- 超文本传输协议HTTP

- 定义了浏览器怎样向万维网服务器请求万维网文档，以及万维网服务器怎样把万维网文档传送给浏览器
- HTTP/1.0采用非持续连接访问：每次浏览器要请求一个文件都要与服务器建立TCP连接，当收到响应后就立即关闭连接
 - 每次请求一个文档就要有两倍的RTT的开销。若一个网页上有很多引用对象，那么请求每一个对象都需要花费2RTT的时间。
 - 为了减少时延，浏览器通常会建立多个并行的TCP连接同时请求多个对象
- HTTP/1.1采用持续连接：万维网服务器在发送响应后仍然保持连接
 - 为了进一步提高效率，HTTP/1.1的持续连接还可以使用流水线方式工作，即浏览器在收到HTTP响应报文之前就能够连续发送多个请求报文
- HTTP报文格式
 - HTTP是面向文本的，其报文中的每一个字段都是一些ASCII码串，并且每个字段的长度都是不同的
 - HTTP请求报文

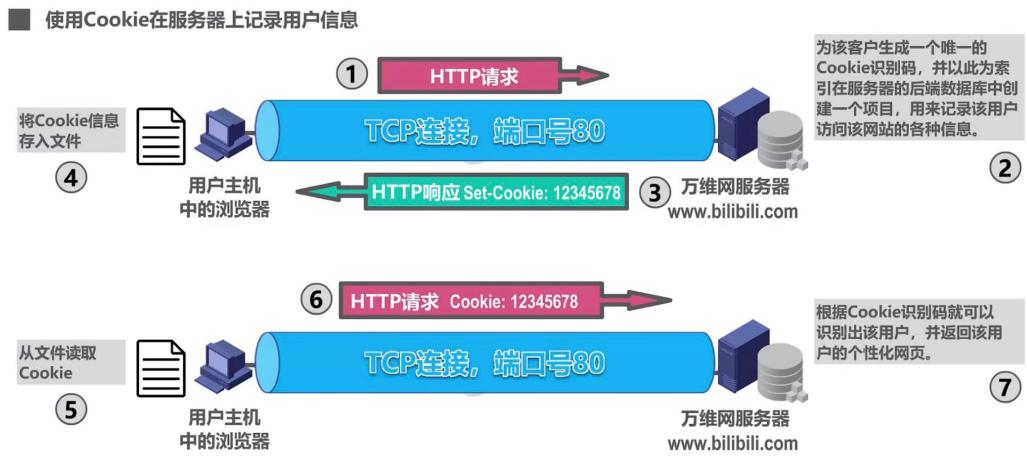


■ HTTP响应报文



• Cookie

- 使用Cookie在服务器上记录用户信息，是一种对无状态的HTTP进行状态化的技术
- 原理



• 万维网缓存与代理服务器

- 在万维网中还可以使用缓存机制提高互联网的效率
- 万维网缓存又称为Web缓存，可位于客户机，也可位于中间系统，位于中间系统的Web缓存又称为代理服务器
- Web缓存把最近一些请求和响应暂存在本地磁盘中。当新请求到达时，若发现这个请求与暂存的请求相同，就返回暂存的响应，而不需要按照URL的地址再次去因特网访问该资源