第一章

网络协议三要素

• 语法：规定传输数据的格式（如何讲）

• 语义：规定所要完成的功能（讲什么）

• 时序：规定各种操作的顺序（双方讲话的顺序）

协议分层结构

- 层次栈

• 每一层都使用其下一层所提供的服务，并为上层提供自己的服务

- 对等实体

• 不同机器上构成相应层次的实体成为对等实体

- 接口

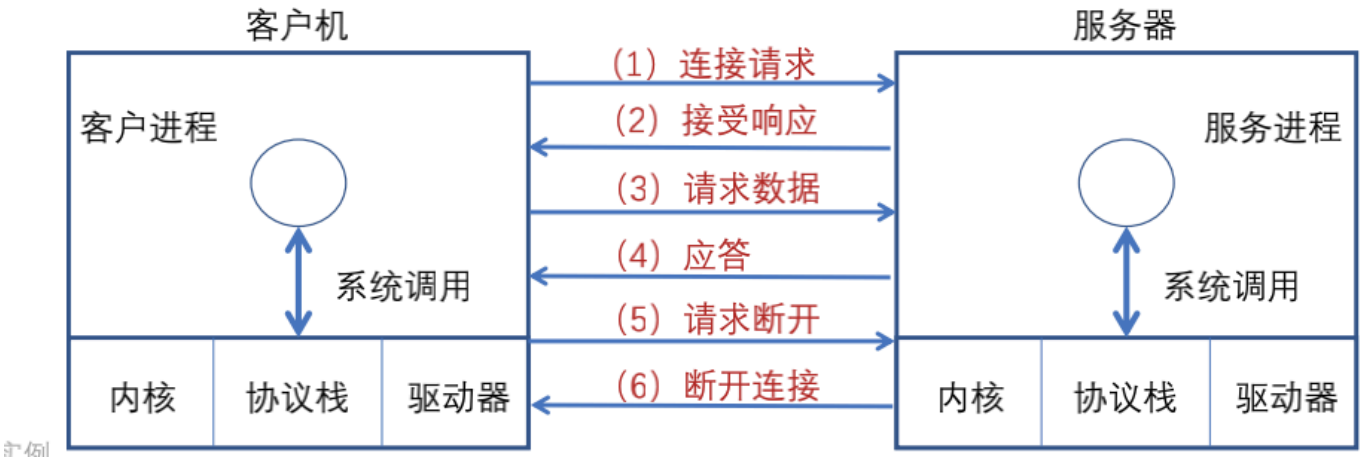
• 在每一对相邻层次之间的是接口；接口定义了下层向上层提供哪些**服务原语**

- 网络体系结构

• 层和协议的集合为网络体系结构，一个特定的系统所使用的一组协议，即每层的协议，称为协议栈

服务原语

➢六个核心服务原语（以面向连接服务为例）



- 实体使用**协议**来实现其定义的服务
- 上层实体通过**接口**使用下层实体的服务

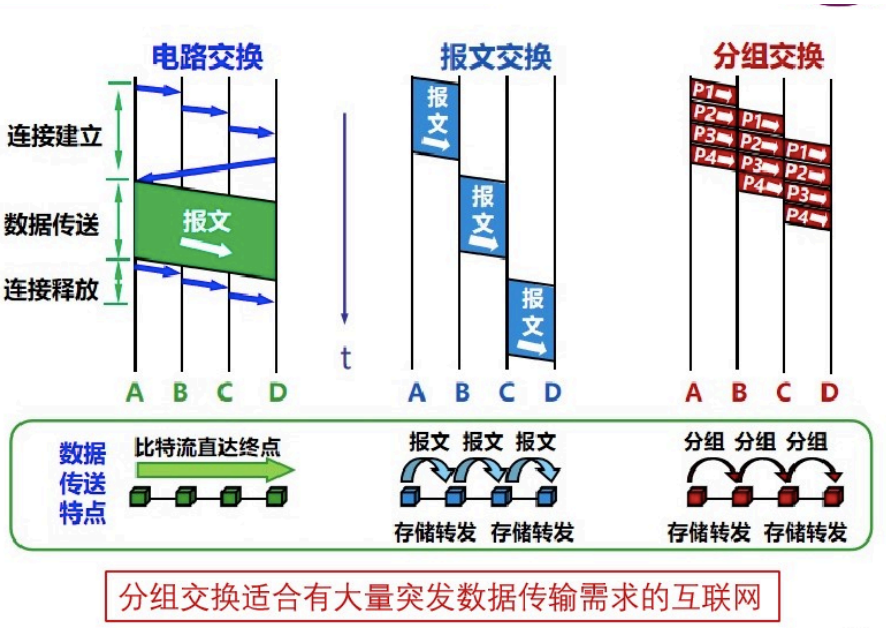
网络核心功能

Ø 功能1：路由

Ø 功能2：转发

➢ 三种交换的比较

- 电路交换需要建立连接并预留资源，难以实现灵活复用
- 报文交换和分组交换较灵活，抗毁性高，在传送突发数据时可提高网络利用率
- 由于分组长度小于报文长度，分组交换比报文交换的时延小，也具有更好的灵活性



OSI标准

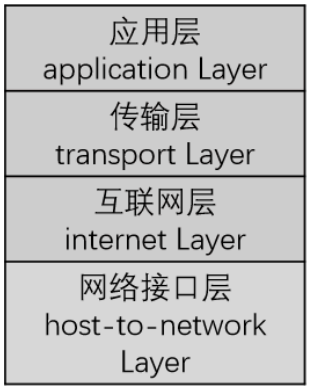
- 物理层
  - 定义如何在信道上传输0、1
- 数据链路层
  - 实现相邻（Neighboring）网络实体间的数据传输
  - 成帧（Framing）：从物理层的比特流中提取出完整的帧
  - 错误检测与纠正：为提供可靠数据通信提供可能
  - 物理地址（MAC address）：48位，理论上唯一网络标识，烧录在网卡，不便更改
  - 流量控制，避免“淹没”（overwhelming）:当快速的发送端遇上慢速的接收端，接收端缓存溢出
  - 共享信道上的访问控制（MAC）：同一个信道，同时传输信号。如同：同一间教室内，多人同时发言，需要纪律来控制
- 网络层
  - 将数据包跨越网络从源设备发送到目的设备（host to host）
  - 路由（Routing）：在网络中选取从源端到目的端转发路径，常常会根据网络可达性动态选取最佳路径，也可以使用静态路由
  - 路由协议：路由器之间交互路由信息所遵循的协议规范，使得单个路由器能够获取网络的可达性等信息
  - 服务质量（QoS）控制：处理网络拥塞、负载均衡、准入控制、保障延迟
- 传输层

- 将数据从源端口发送到目的端口（进程到进程）
- 网络层定位到一台主机（host），传输层的作用域具体到主机上的某一个进程
- 两类模式：可靠的传输模式，或不可靠传输模式
- 可靠传输：可靠的端到端数据传输，适合于对通信质量有要求的应用场景，如文件传输等
- 不可靠传输：更快捷、更轻量的端到端数据传输，适合于对通信质量要求不高，对通信响应速度要求高的应用场景，如语音对话、视频会议等

TCP/IP

- TCP/IP参考模型：ARPANET所采用
  - 以其中最主要的两个协议TCP/IP命名
  - Vint Cerf和Bob Kahn于1974年提出
- 网络接口层（Host-to-network Layer）
  - 描述了为满足无连接的互联网络层需求，链路必须具备的功能
- 互联网层（Internet Layer）
  - 允许主机将数据包注入网络，让这些数据包独立的传输至目的地，并定义了数据包格式和协议（IPv4协议和IPv6协议）
- 传输层（Transport Layer）
  - 允许源主机与目标主机上的对等实体，进行端到端的数据传输：TCP，UDP
- 应用层（Application Layer）
  - 传输层之上的所有高层协议：DNS、HTTP、FTP、SMTP...

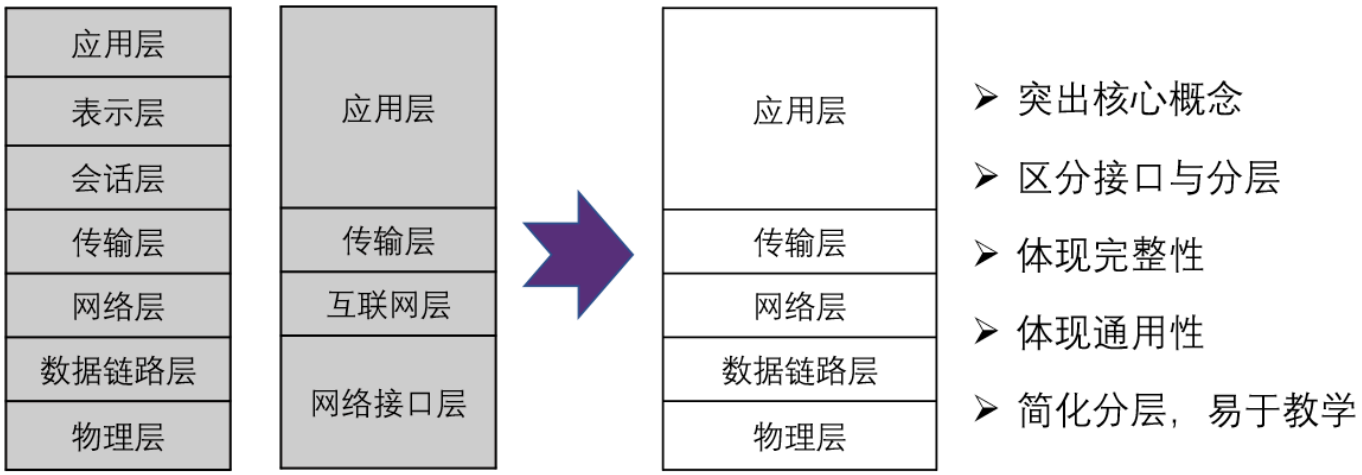
ARPNET
最终采用TCP和IP
为主要协议



- 先有TCP/IP协议栈，然后有TCP/IP参考模型
- 参考模型只是用来描述协议栈的

54

- 端对端原则：采用聪明终端&简单网络，由端系统
- TCP/IP模型的网络层仅支持无连接通信（IP）
- OSI模型网络层能够支持无连接和面向连接通信



OSI 7层模型　TCP/IP 4层模型　本教程的分层组织

第二章

物理层功能

功能：如何在连接各计算机的传输媒体上传输数据比特流

• 数据链路层将数据比特流传送给物理层

• 物理层将比特流按照传输媒体的需要进行编码

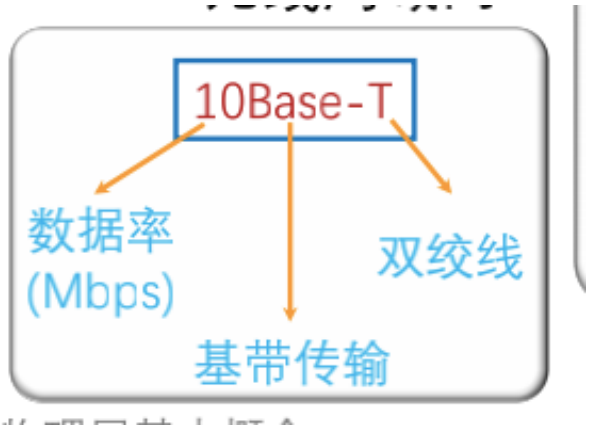
• 然后将信号通过传输媒体传输到下一个节点的物理层

作用：尽可能地屏蔽掉不同传输媒体和通信手段的差异

接口特性

- 机械特性
  - 定义接线器的形状和尺寸、引线数目和排列、固定和锁定装置等
- 电气特性
  - 发送器和接收器的电路特性、负载要求、传输速率和连接距离等
  - 如发送信号电平、发送器和接收器的输出阻抗、平衡特性等
- 功能特性
  - 描述接口执行的功能，定义接线器的每一引脚(针，Pin)的作用
- 过程特性
  - 指明对于不同功能的各种可能事件的出现顺序

物理层协议是DTE和DCE间的约定，规定了两者之间的接口特性



传输介质

导引型传输介质

• 双绞线、同轴电缆、电力线和光纤等

非导引型传输介质

• 短波传输、地面微波 、卫星微波、光波传输等

复用

基本概念

复用 (multiplexing) 技术的目的是：允许用户使用一个共享信道进行通信，避免相互干扰，降低成本，提高利用率。

- 频分复用: 频分复用将整个带宽分为多份，用户在分配到一定的频带后，在通信过程中自始至终都占用这个频带. **频分复用**的所有用户在同样的时间占用不同的带宽资源（请注意，这里的“带宽”是**频率带宽**而不是数据的发送速率）
- 时分复用: 将时间划分为一段段等长的时分复用帧（TDM帧）每一个时分复用的用户在每一个 TDM 帧中占用固定序号的时隙**,周期性、等时信号**
- 统计时分复用: 是指**动态地按需分配**共用信道的时隙，只将需要传送数据的终端接入共用信道，以提高信道利用率的多路复用技术。
- 波分复用: 是利用多个激光器在单条**光纤**上同时发送多束不同**波长**激光的技术
- 码分复用: 是指利用**码序列**相关性实现的多址通信 , 基本思想是靠不同的**地址码**来区分的地址
    - 相乘得1为发送1、为0为未发送、为-1为发送了0

第三章 数据链路层

逻辑链路控制(LLC)

介质访问控制(MAC)

功能

- 成帧 （Framing）

• 将比特流划分成“帧”的主要目的是为了检测和纠正物理层在比特传输中可能出现的错误，数据链路层功能需借助“帧”的各个域来实现

- 差错控制 （Error Control）

• 处理传输中出现的差错，如位错误、丢失等

- 流量控制 （Flow Control）

• 确保发送方的发送速率，不大于接收方的处理速率

差错检测

- 奇偶检验 (Parity Check)：1位奇偶校验是最简单、最基础的检错码
- 校验和 (Checksum)：主要用于TCP/IP体系中的网络层和传输层
- 循环冗余校验 (Cyclic Redundancy Check，CRC)：数据链路层广泛使用的校验方法

海明距离

奇偶校验

CRC校验码

实现

- 物理层进程和某些数据链路层进程运行在**专用硬件**上（网络接口卡）
- 数据链路层进程的其他部分和网络层进程作为操作系统的一部分运行在CPU上，数据链路层进程的软件通常以**设备驱动**的形式存在

滑动窗口

Ø 目的

• 对可以连续发出的最多帧数（已发出但未确认的帧）作限制

Ø 序号使用

• 循环**重复使用**有限的帧序号

Ø 流量控制：接收窗口驱动发送窗口的转动

Ø 累计确认：不必对收到的分组逐个发送确认，而是对按序到达的最后一个分组发送确认

回退N协议

Ø 出错全部重发

• 当接收端收到一个出错帧或乱序帧时，丢弃所有的后继帧，并且不为这些帧发送确认

• 发送端超时后，**重传所有未被确认的帧**

Ø 适用场景

• 该策略**对应接收窗口为1**的情况，即只能按顺序接收帧

Ø 优缺点

• 优点：连续发送提高了信道利用率

• 缺点：按序接收，出错后即便有正确帧到达也丢弃重传

选择重传协议

Ø设计思想

• 若发送方发出连续的若干帧后，收到对其中某一帧的否认帧，或某一帧的定时器超时， 则**只重传该出错帧或计时器超时的数据帧**

Ø 适用场景

• 该策略对**应接收窗口大于1**的情况，即暂存接收窗口中序号在出错帧之后的数据帧

Ø 优缺点

• 优点：避免重传已正确传送的帧

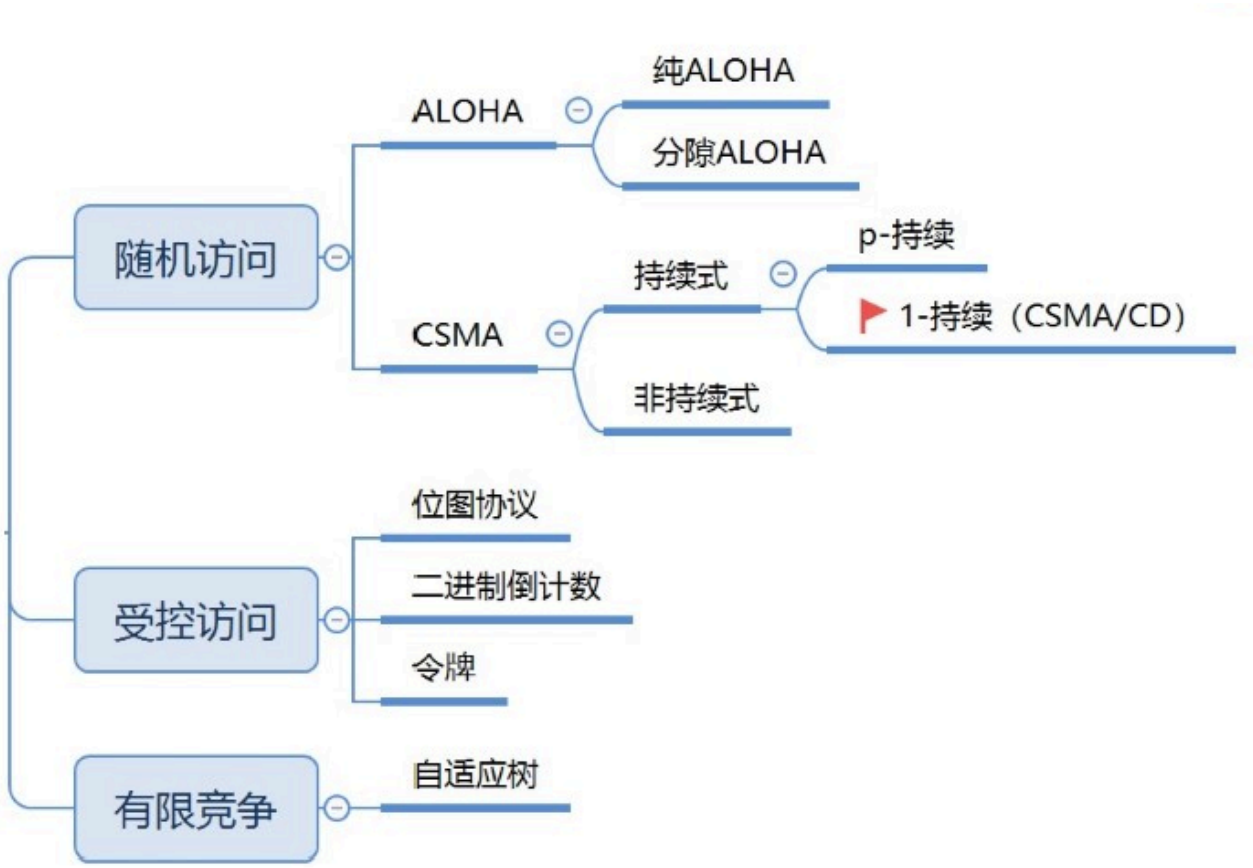
• 缺点：在接收端需要占用一定容量的缓存

协议举例: PPP协议

第四章

常见的局域网拓扑

- 总线拓扑、星型拓扑、环型拓扑
- 早期星型拓扑是**集线器**，现在几乎都是**交换机**



- 纯ALOHA 想发就发
- 分隙ALOHA 把时间分成时隙.帧的发送必须在时隙的起点,冲突只发生在时隙的起点
- 非持续式CSMA :如果介质忙就等待一个随机分布的时间
- 1-持续 :• ①经侦听，如介质空闲，则发送。
  - • ②如介质忙，持续侦听，一旦空闲立即发送。
  - • ③如果发生冲突，等待一个随机分布的时间再重复步骤①
- p-持续: • ①经侦听，如介质空闲，那么以 p的概率 发送，以(1–p)的概率延迟一个时间单元发送
  - • ②如介质忙，持续侦听，一旦空闲重复①

• ③如果发送已推迟一个时间单元，再重复步骤①

Ø 以太网采用了**CSMA/CD(1-持续CSMA)**

• 吞吐量：比ALOHA高，比P-持续式CSMA低

• 冲突：比ALOHA少，比P-持续式高

• P-持续式付出了**高延迟的代价**

经典以太网

物理层

Ø最高速率10Mbps

Ø使用曼彻斯特编码

Ø使用同轴电缆和中继器连接

MAC子层协议

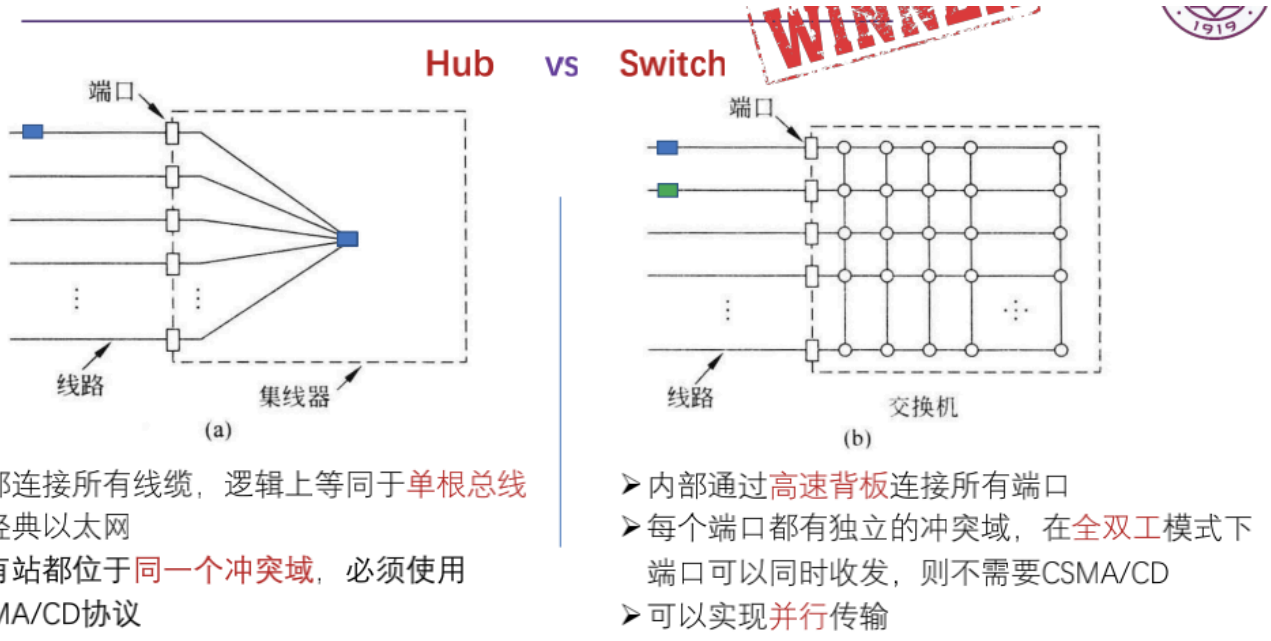
Ø 主机运行CSMA/CD协议

MAC帧中的源地址和目的地址长度均为6字节

交换式以太网

交换式以太网的核心是**交换机**

工作在数据链路层，检查MAC 帧的**目的地址**对收到的帧进行转发



- 内部连接所有线缆，逻辑上等同于单根总线的经典以太网
- 所有站都位于同一个冲突域，必须使用CSMA/CD协议

- 内部通过高速背板连接所有端口
- 每个端口都有独立的冲突域，在全双工模式下端口可以同时收发，则不需要CSMA/CD
- 可以实现并行传输

万兆以太网

Ø 10-Gigabit Ethernet(IEEE 802.3ae, 2002)

• 1Gbps —-> 10Gbps

• 常记为10GE, 10GbE 或 10 GigE

• **只支持全双工，不再使用CSMA/CD**

• 保持兼容性

• 重点是**超高速的物理层**

数据链路层交换原理

数据链路层设备扩充网络

• 网桥或交换机

• 分隔了冲突域

- 转发
- 过滤
- 泛洪(广播帧、未知单播帧)

逆向学习

根据帧的源地址在MAC地址表查找匹配表项，

ü如果没有，则增加一个新表项（源地址、入境端口、帧到达时间），

ü如果有，则更新原表项的帧到达时间，重置老化时间。

• 对入境帧的转发过程（三选一），查帧的目的地址是否在MAC地址表中

ü如果有，且入境端口≠出境端口，则从对应的出境端口(转发帧)；

ü如果有，且入境端口=出境端口，则丢弃帧（过滤帧）；

ü如果没有，则向除入境端口以外的其它所有端口(泛洪帧)。

第五章

网络层服务的实现

Ø网络层实现端系统间**多跳传输**可达

Ø网络层功能存在每台主机和路由器中

• 发送端：将传输层数据单元封装在数据包中

• 接收端：解析接收的数据包中，取出传输层数据单元，交付给传输层

• 路由器：检查数据包首部，转发数据包

网络层关键功能

- 路由（控制面）

• 选择数据包从源端到目的端的路径

• 核心：路由算法与协议

- 转发（数据面）

• 将数据包从路由器的输入接口传送到正确的输出接口

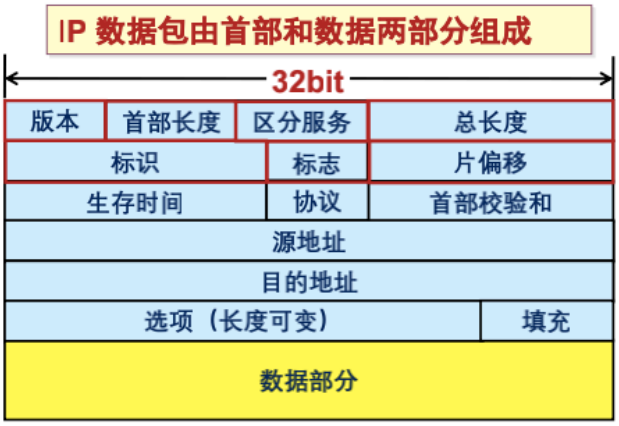
IPv4协议

Internet协议执行两个基本功能,一种**无连接**的协议，是互联网的核心

• 寻址(addressing)

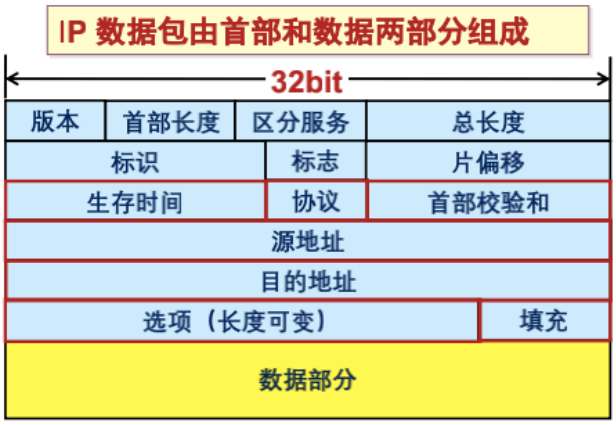
• 分片(fragmentation)

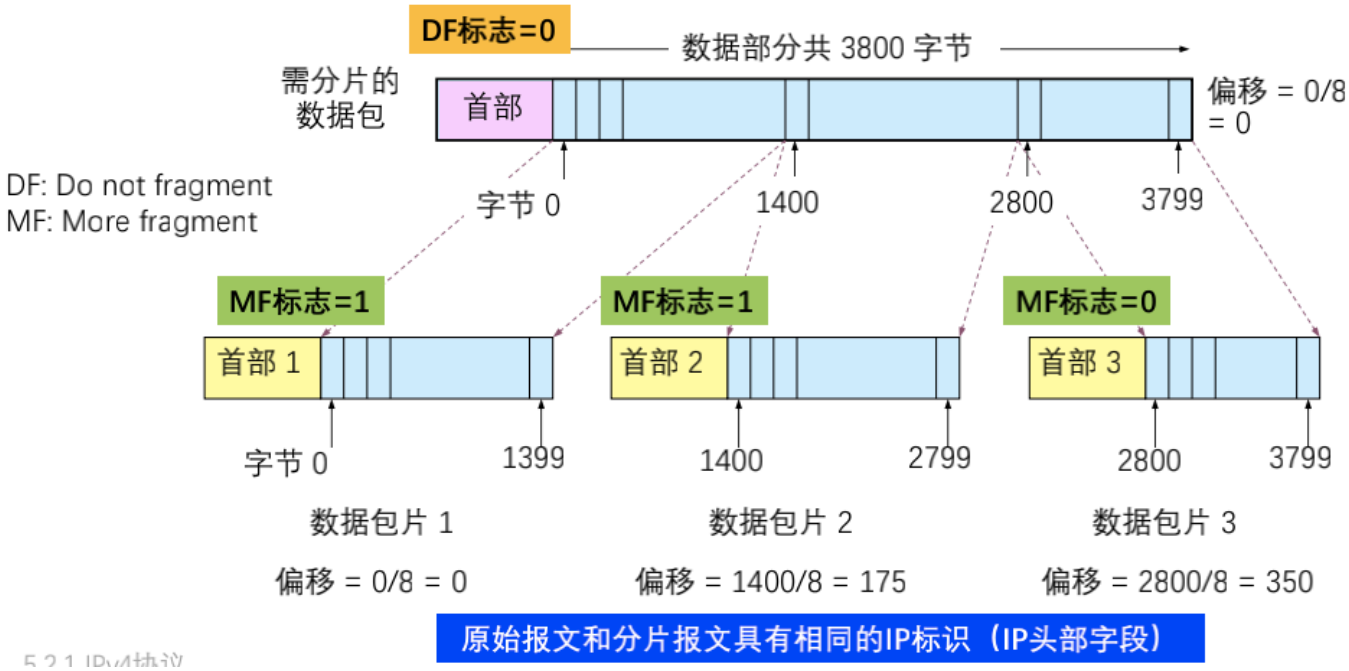
- 版本： 4bit， 表示采用的IP协议版本
- 首部长度： 4bit，表示整个IP数据包首部的长度
- 区分服务： 8bit， 该字段一般情况下不使用
- 总长度： 16bit ， 表示整个IP报文的长度,能表示的最大字节为2^16-1=65535字节
- 标识： 16bit ， IP软件通过计数器自动产生，每产生1个数据包计数器加1；在IP分片以后，用来标识同一片分片
- 标志： 3bit，目前只有两位有意义； MF，置1表示后面还有分片，置0表示这是数据包片的最后1个；DF，不能分片标志，置0时表示允许分片
- 片偏移： 13bit，表示IP分片后，相应的IP片在总的IP片的相对位置



DF: Do not fragment
MF: More fragment

- 生存时间TTL(Time To Live) ：8bit,表示数据包在网络中的生命周期，用通过路由器的数量来计量，即跳数（每经过一个路由器会减1）
- 协议：8bit，标识上层协议（TCP/UDP/ICMP…）
- 首部校验和：16bit， 对数据包首部进行校验，不包括数据部分
- 源地址：32bit，标识IP片的发送源IP地址
- 目的地址：32bit，标识IP片的目的地IP地址
- 选项：可扩充部分，具有可变长度，定义了安全性、严格源路由、松散源路由、记录路由、时间戳等选项
- 填充：用全0的填充字段补齐为4字节的整数倍





Ø IPv4分组在传输途中可以多次分片

• 源端系统，中间路由器（可通过标志位设定是否允许路由器分片）

Ø IPv4分片只在目的IP对应的目的端系统进行重组

Ø IPv4分片、重组字段在基本IP头部

• 标识、标志、片偏移

Ø 支持多跳寻路将IP数据包送达目的端：*目的IP地址*

Ø 表明发送端身份：*源IP地址*

Ø 根据IP头部协议类型，提交给不同上层协议处理：*协议*

Ø 数据包长度大于传输链路的MTU的问题，通过分片机制解决：*标识*、 *标志*、*片偏移*

Ø 防止循环转发浪费网络资源（路由错误、设备故障...），通过跳数限制解决：*生存时间TTL*

Ø IP报头错误导致无效传输，通过头部机校验解决：*首部校验和*

IP特殊地址

地址	用途
全0网络地址	只在系统启动时有效，用于启动时临时通信，又叫主机地址
127.X.X.X	指本地节点(一般为127.0.0.1)，用于测试网卡及TCP/IP软件，这样浪费了1700万个地址
全0主机地址	用于指定网络本身，称之为网络地址或者网络号
全1主机地址	用于广播，也称定向广播，需要指定目标网络
0.0.0.0	指任意地址
255.255.255.255	用于本地广播，也称有限/受限广播，无须知道本地网络地址

IP数据包经过不同链路时，IP 数据包中封装的IP地址. ARP 不发生改变，而Mac帧中的硬件地址是发生改变的

ARP协议工作过程

主机A的IP地址为192.168.1.1，MAC地址为0A-11-22-33-44-01；
主机B的IP地址为192.168.1.2，MAC地址为0A-11-22-33-44-02；
当主机A要与主机B通信时，地址解析协议可以将主机B的IP地址（192.168.1.2）解析成主机B的MAC地址。工作流程如下：
1、根据主机A上的路由表内容，IP确定用于访问主机B的转发IP地址是192.168.1.2。
2、如果主机A在ARP缓存中没有找到映射，它将询问192.168.1.2的硬件地址，从而将ARP请求帧广播到本地网络上的所有主机。
3、主机B确定ARP请求中的IP地址与自己的IP地址匹配，则将主机A的IP地址和MAC地址映射添加到本地ARP缓存中。
4、主机B将包含其MAC地址的ARP回复消息直接发送回主机A。
5、当主机A收到从主机B发来的ARP回复消息时，会用主机B的IP和MAC地址映射更新ARP缓存。

NAT技术

• A类地址：10.0.0.0--10.255.255.255

• B类地址：172.16.0.0--172.31.255.555

• C类地址：192.168.0.0--192.168.255.255

路由算法

DHCP 工作过程

DHCP (Dynamic Host Configuration Protocol)：动态主机配置协议

• 当主机加入IP网络，允许主机从DHCP服务器动态获取IP地址

• 可以有效利用IP地址，方便移动主机的地址获取

工作模式：客服/服务器模式（C/S）

• 基于 UDP 工作，服务器运行在 67 号端口， 客户端运行在 68 号端口

DHCP服务不只返回客户机所需的IP地址，还包括:

• 缺省路由器IP地址

• DNS服务器IP地址

• 网络掩码

工作流程

Ø DHCP 客户从UDP端口68以广播形式向服务器发送发现报文

Ø DHCP 服务器以广播形式发出提供报文

Ø DHCP 客户从多个DHCP服务器中选择一个，并向其以广播形式发送DHCP请求报文

Ø 被选择的DHCP服务器以广播形式发送确认报文（DHCPACK）

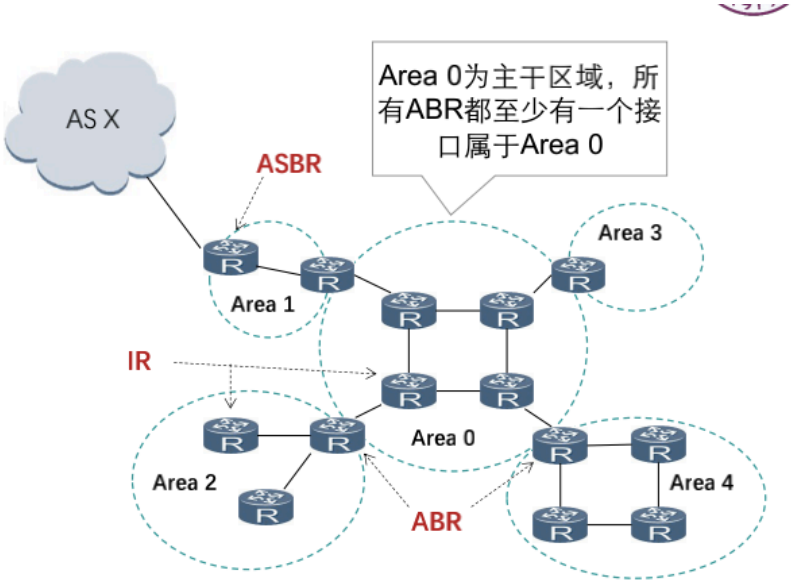
ICMP和Traceroute工作流程

• ICMP 报文携带在IP 数据包中： IP上层协议号为1

算法

OSPF-区域的概念

- OSPF的区域
  - 主干区域
  - 非主干区域
- 路由器角色
  - 内部路由器(Internal Router，IR)
  - 区域边界路由器(Area Bounder Router，ABR)
  - 自治系统边界路由器(AS Bounder Router, ASBR)



RIP协议工作流程

使用**跳数**衡量到达目的网络的距离

BGP功能

外部网关协议,目前互联网中唯一实际运行的自治域间的路由协议

• eBGP：从相邻的AS获得网络可达信息

• iBGP： 将网络可达信息传播给AS内的路由器

• 基于网络可达信息和策略决定到其他网络的“最优”路由

BGP协议的特点

Ø BGP 协议交换路由信息的节点数量级是**自治系统数的量级**

Ø 每一个自治系统边界路由器的数目是很少的

Ø 在 BGP 刚刚运行时，BGP 的邻站交换整个的 BGP 路由表；以后只需要在发生变化时**更新有变化的部分**

BGP通过TCP的179端口交换报文

第六章

套接字编程

源IP地址 = 客户IP地址，源端口号 = 客户套接字端口号

目的IP地址 = 服务器IP地址，目的端口号 = 服务器监听套接字的端口号

UDP

校验和字段的作用: 对传输的报文段进行检错

Ø UDP需要实现的功能：

• 复用和分用

• 报文检错

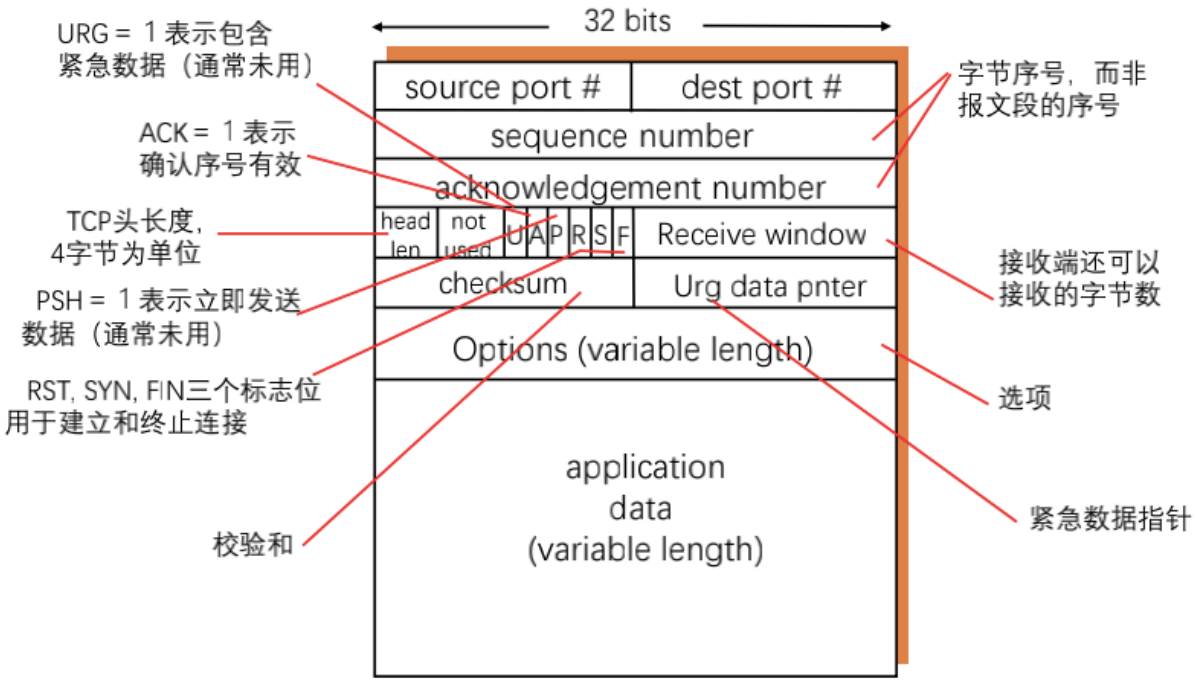
UDP校验和的使用是可选的，若不计算校验和，该字段填入0

计算UDP校验和时，要包括**伪头**、**UDP头和数据**三个部分

Ø UDP伪头是一个虚拟数据结构，取自IP报头，并非UDP报文的实际有效成分

接收方对UDP报文（包括校验和）及伪头求和，若结果为0xFFFF，认为没有错误

TCP段报文格式



重要的TCP选项

Ø 最大段长度（MSS）：

• TCP段中可以携带的最大数据字节数

• 建立连接时，每个主机可声明自己能够接受的MSS，缺省为536字节

Ø 选择确认（SACK）：

• 最初的TCP协议只使用累积确认

• 改进的TCP协议引入选择确认，允许接收端指出缺失的数据字节

TCP协议

Ø 接收方：

• 确认方式：采用累积确认，仅在正确、按序收到报文段后，更新确认序号；其余情况，**重复前一次的确认序号（与回退N协议类似）**

• 失序报文段处理：缓存失序的报文段（与选择重传协议类似）

Ø 发送方：

• 发送策略：流水线式发送报文段

• 定时器的使用：仅对最早未确认的报文段使用一个重传定时器（与回退N协议类似）

• 重发策略：仅在超时后重发最早未确认的报文段（与选择重传协议类似，因为接收端缓存了失序的报文段）

Ø 收到应用数据：

• 创建并发送TCP报文段

• 若当前没有定时器在运行（没有已发送、未确认的报文段），启动定时器

Ø 超时:

• 重传**包含最小序号的、未确认的报文段**

• 重启定时器

Ø 收到ACK：

• 如果确认序号大于基序号（已发送未确认的最小序号）：

• 推进发送窗口（更新基序号）

• 如果发送窗口中还有未确认的报文段，启动定时器，否则终止定时器

TCP收端的事件和处理

接收端事件	接收端动作
收到一个期待的报文段，且之前的报文段均已发送过确认	推迟发送确认，在500ms时间内若无下一个报文段到来，发送确认
收到一个期待的报文段，且前一个报文段被推迟确认	立即发送确认（估计RTT的需要）
收到一个失序的报文段（序号大于期待的序号），检测到序号间隙	立即发送确认（快速重传的需要），重复当前的确认序号
收到部分或全部填充间隙的报文段	若报文段始于间隙的低端，立即发送确认（推进发送窗口），更新确认序号

所谓快速重传，就是在定时器到期前重发丢失的报文段

流量控制

为什么回退N、选择重传和UDP不需要流量控制

Ø 回退N协议和选择重传协议均假设：

• 正确、按序到达的分组被立即交付 给上层

• 其占用的缓冲区被立即释放

Ø 发送方根据确认序号即可知道：

• 哪些分组已被移出接收窗口

• 接收窗口还可以接受多少分组

ØUDP不保证交付：

• 接收端UDP将收到的报文载荷放入接收缓存

• 应用进程每次从接收缓存中读取一个完整的报文载荷

• 当应用进程消费数据不够快时，

接收缓存溢出，报文数据丢失， UDP不负责任

TCP触发条件

在TCP协议中，触发一次TCP传输

需要满足以下三个条件之一：

• 应用程序调用

• 超时

• 收到数据或确认

小结

ØNagle算法的解决方法：

• 在新建连接上，当应用数据到来时，组成一个TCP段发送（那怕只有一个字节）

• 在收到确认之前，后续到来的数据放在发送缓存中

• 当数据量达到一个MSS或上一次传输的确认到来（取两者的较小时间），用一个TCP段将缓存的字节全部发走

Ø TCP接收端

• 使用显式的窗口通告，告知发送方可用的缓存空间大小

• 在接收窗口较小时，推迟发送确认

• 仅当接收窗口显著增加时，通告新的窗口大小

Ø TCP发送端

• 使用Nagle算法确定发送时机

• 使用接收窗口限制发送的数据量，已发送未确认的字节数不超过接收窗口的大小

TCP三次握手

1. 客户TCP发送SYN 报文段（SYN=1, ACK=0）

• 给出客户选择的起始序号

• 不包含数据

2. 服务器TCP发送SYNACK报文段（SYN=ACK=1）（服务器端分配缓存和变量）

• 给出服务器选择的起始序号

• 确认客户的起始序号

• 不包含数据

3. 客户发送ACK报文段（SYN=0，ACK=1）（客户端分配缓存和变量）

• 确认服务器的起始序号

• 可能包含数据

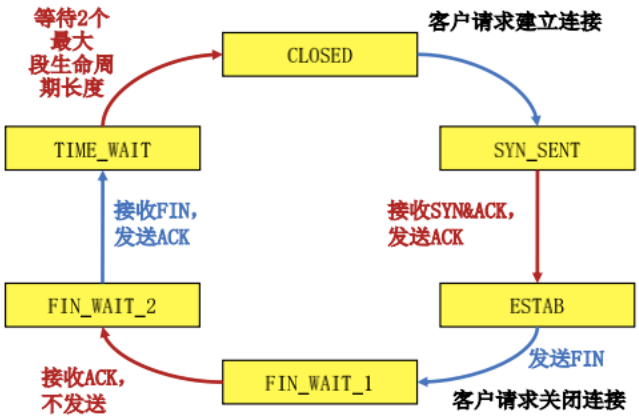
为什么起始序号不从0开始?

• 若在不同的时间、在同一对套接字之间建立了连接，则新、旧连接上的序号有重叠，旧连接上重传的报文段会被误以为是新连接上的报文段

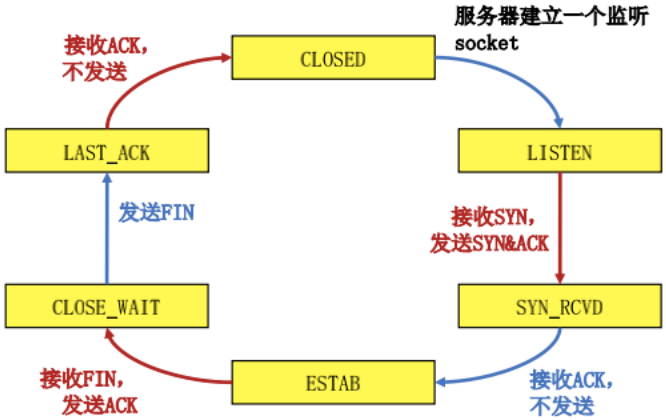
Ø 可以随机选取起始序号吗?

• 若在不同的时间、在同一对套接字之间建立了连接，且新、旧连接上选择的起始序号x和y相差不大，那么新、旧连接上传输的序号仍然可能重叠

Ø 结论：必须避免新、旧连接上的序号产生重叠



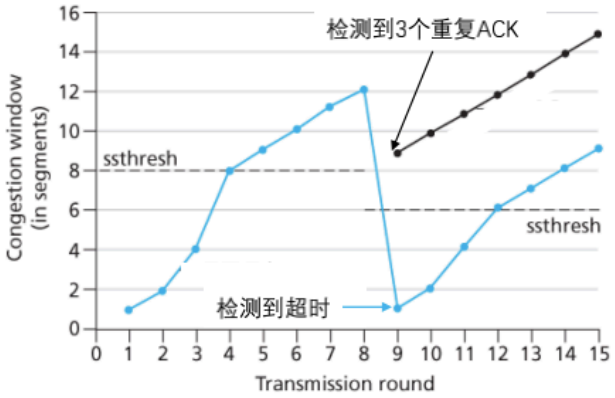
客户生命周期



服务器生命周期

网络拥塞

- 发送方维护变量ssthresh
- 发生丢包时，ssthresh=cwnd/2
- ssthresh是从慢启动转为拥塞避免的分水岭：
  - cwnd低于ssthresh时，执行慢启动
  - cwnd高于ssthresh时：执行拥塞避免
- 拥塞避免阶段，拥塞窗口线性增长：
  - 每当收到ACK， cwnd=cwnd + MSS*(MSS/cwnd)
  - 相当于每经过一个RTT，cwnd增加一个MSS



- 检测到3个重复的ACK后（黑色线）
  - cwnd= ssthresh+3，线性增长
- 检测到超时后（蓝色线）：
  - cwnd=1 MSS，慢启动

状态	事件	TCP发送方操作	说明
慢启动 (SS)	收到新的确认	cwnd = cwnd + MSS, If (cwnd > ssthresh) set state to "Congestion Avoidance"	每收到一个ACK段，cwnd增加一个MSS；即每经过一个RTT，cwnd加倍
拥塞避免 (CA)	收到新的确认	cwnd = cwnd+MSS * (MSS/cwnd)	每经过一个RTT，cwnd增加一个MSS
SS or CA	收到3个重复的确认	ssthresh = cwnd/2, cwnd = ssthresh+3, Set state to "Congestion Avoidance"	cwnd减半后加3，然后线性增长
SS or CA	超时	ssthresh = cwnd/2, cwnd = 1 MSS, Set state to "Slow Start"	cwnd降为一个MSS，进入慢启动
SS or CA	收到一个重复的确认	统计收到的重复确认数	cwnd 和 ssthresh 都不变

第七章

- 两台主机通信实际是其上对应的两个应用进程(process)在通信

三种方式

客户/服务器（C/S, Client/Server）方式

浏览器/服务器（B/S，Browser/Server） 方式

对等（P2P，Peer to Peer）方式 287

C/S方式

- C/S方式可以是面向连接的，也可以是无连接的
- 面向连接时，C/S通信关系一旦建立，通信就是双向的，双方地 位平等，都可发送和接收数据

B/S方式

Ø B/S方式可以看做**C/S方式**的特例，即客户软件改为浏览器了

Ø B/S方式采取**浏览器请求**、**服务器响应**的工作模式

Ø 在B/S方式下，用户界面完全通过Web浏览器实现，一部分事务逻辑在前端实现，但主要的事务逻辑在服务器端实现

B/S方式通常采取3层架构实现

• 数据层：由数据库服务器承担数据处理逻辑

• 处理层：由Web服务器承担业务处理逻辑和页面存储管理

• 表现层：浏览器仅承担网页信息的浏览功能, 以超文本格式实现信息的输入和浏览

对等（P2P，Peer to Peer）方式

对等方式是指两个进程在通信时并不区分服务的请求方和服务的提供方

• 只要两个主机都运行P2P软件，它们就可以进行**平等、对等的通信**

• 双方都可以下载对方存储在硬盘中的共享文档

服务器进程工作方式

- 循环方式(iterative mode)

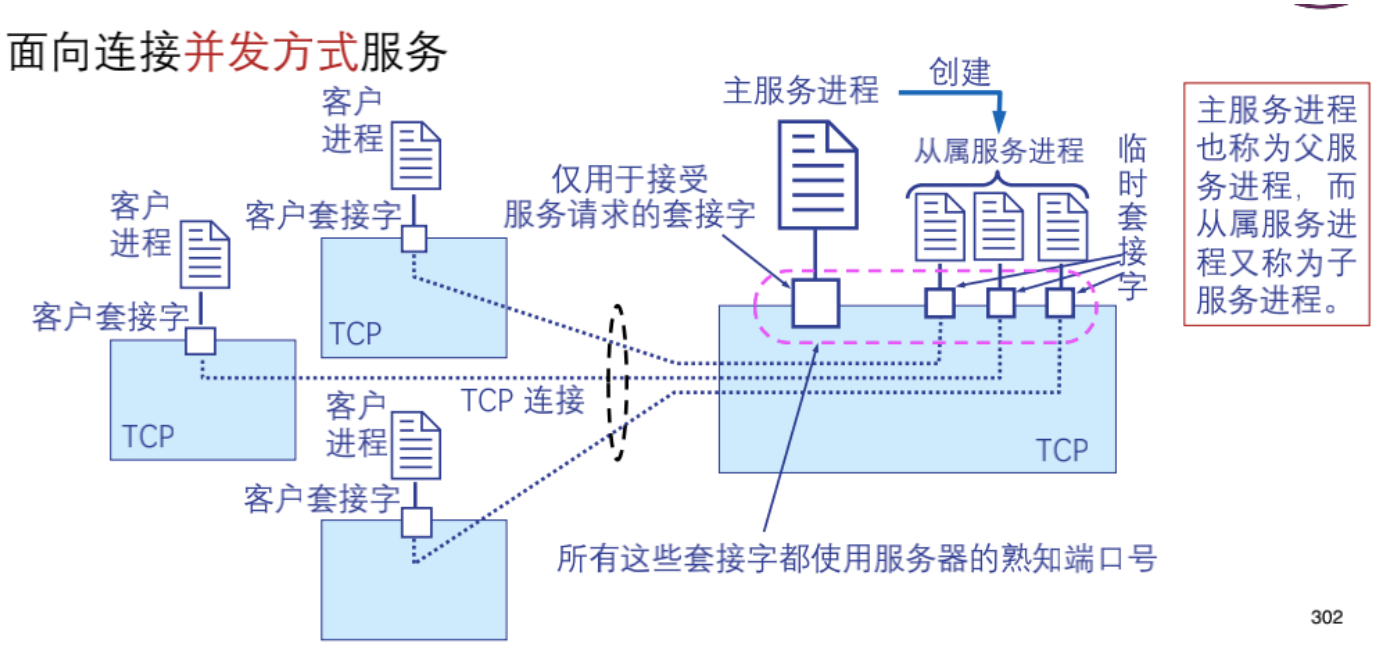
• 一次只运行一个服务进程

• 当有多个客户进程请求服务时，服务进程就按请求的先后顺序依次做出响应,UDP (**阻塞方式**)

- 并发方式(concurrent mode)

• 可以同时运行多个服务进程

• 每一个服务进程都对某个特定的客户进程做出响应,TCP (**非阻塞方式**)



域名系统

• 域名系统（DNS，Domain Name System）是互联网重要的基础设施之一，向所有需要域名解析的应用提供服务，主要负责将可读性好的域名映射成IP地址

• 名字到域名的解析是由若干个域名服务器程序完成的。域名服务器程序在专设的节点上运行，相应的节点也称为名字服务器(Name Server)或域名服务器(Domain Name Server)

- 域名与IP地址可以是**一对一、一对多或者多对一的关系**
- 域名服务器的管辖范围以“区”为单位，而不是以“域”为单位
- 管辖区是域名“域”的子集，管辖区可以小于或等于域，但不可能大于域

类别

域名系统的域名服务器分为两大类

• 权威域名服务器(authoritative name server)

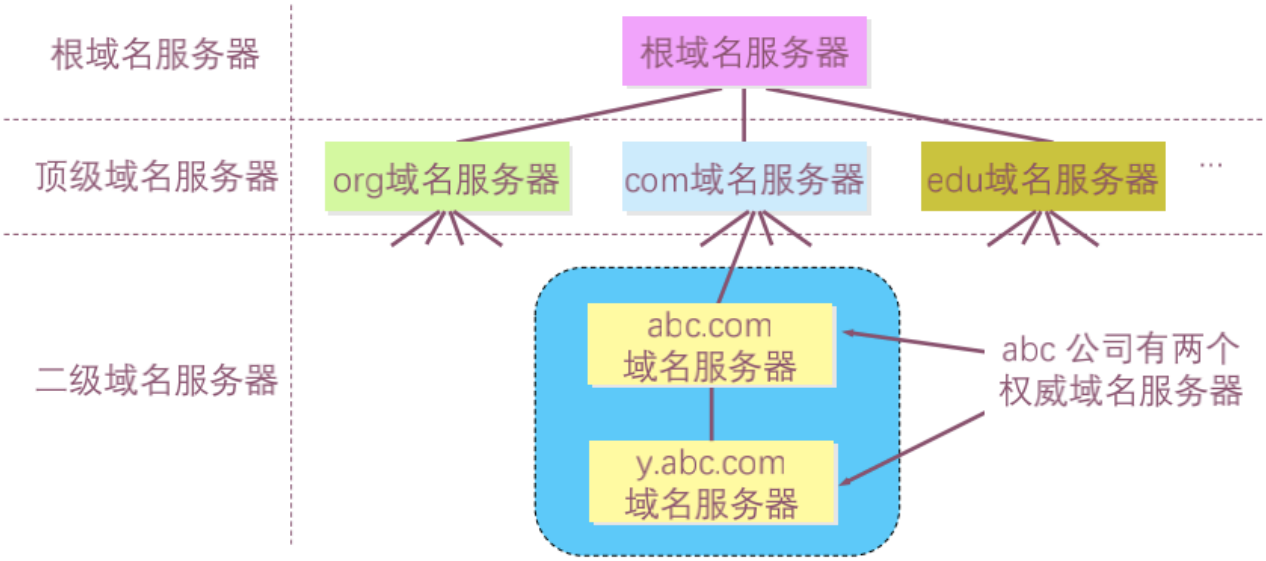
• 每个DNS区至少应有一个IPv4可访问的权威域名服务器提供服务

• 递归服务器(recursive resolver)

• 以递归方式运行的、使用户程序联系域(domain)域名服务器的程序

三级域及以下的域名服务器也统称为**本地域名服务器**

层次树状结构的权威域名服务器



域名解析过程

域名查询有**递归查询**(recursive query)和**迭代查询**(或循环查询，iterative query)两种方式

• 主机向递归解析器/本地域域名服务器的查询一般采用递归查询

• 递归解析器/本地域域名服务器向根服务器可以采用递归查询，但一般优先采用迭代查询

UDP数据报, 端口号为53

递归查询

当收到查询请求报文的域名服务器不知被查询域名的IP地址时，该域名服务器就以DNS客户的身 份向下一步应查询的域名服务器发出查询请求， 即替递归服务器继续查询

• 较少使用

迭代查询

当收到查询请求报文的域名服务器不知道被查询域名的IP地址时，就把自己知道的下一步应查询的域名服务器IP地址告诉本地域域名服务器，由本地域域名服务器继续向该域名服务器查询，直到得到所要解析的域名的IP地址，或者查询不到所要解析的域名的IP地址

• 通常使用

电子邮件

电子邮件系统采用**客户/服务器**工作模式

SMTP利用**TCP可靠地从客户向服务器传递邮件，使用端口25**

• 直接投递: 发送端直接到接收端

• SMTP的3个阶段：连接建立、邮件传送、连接关闭

POP3协议

当用户代理打开一个到端口**110上的TCP**连接后，客户/服务器开始工作

POP3使用客户/服务器工作方式

POP3的三个阶段：

• 认证(Authorization)：处理用户登录的过程

• 事务处理(Transactions)：用户收取电子邮件，并将邮件标记为删除

• 更新(Update)：将标为删除的电子邮件删除

IMAP

IMAP—Internet邮件访问协议[RFC 2060]

• 用于最终交付的主要协议

• 邮件服务器运行侦听端口为143的IMAP服务

Telent

Telnet协议使用C/S方式实现

• 在本地系统运行Telnet客户进程，在远程主机运行Telnet服务器进程

Telnet协议使用TCP连接通信

• 服务器进程默认监听TCP 23端口，使用主进程等待新的请求，并产生从属进程来处理每一个连接

特点; 使用Telnet协议在网络中传输的数据都是NVT格式，不同的用户终端与服务器进程均与本地终端格式无关

FTP

FTP使用C/S方式实现

Ø FTP工作过程

• 服务器主进程打开TCP21端口，等待客户进程发出的连接请求

• 客户可以用分配的任意**一个本地端口号与服务器进程的TCP 21端口**进行连接

• 客户请求到来时，服务器主进程启动**从属进程**来处理客户进程发来的请求

TFTP

简单文件传输协议 TFTP

• TFTP(Trivial File Transfer Protocol) 是一个很小且易于实现的文件传输协议

• 使用C/S方式和UDP协议实现

SNMP

SNMP协议实现

- SNMP使用无连接的UDP协议实现
- SNMP使用C/S方式实现