

# 第七章 应用层

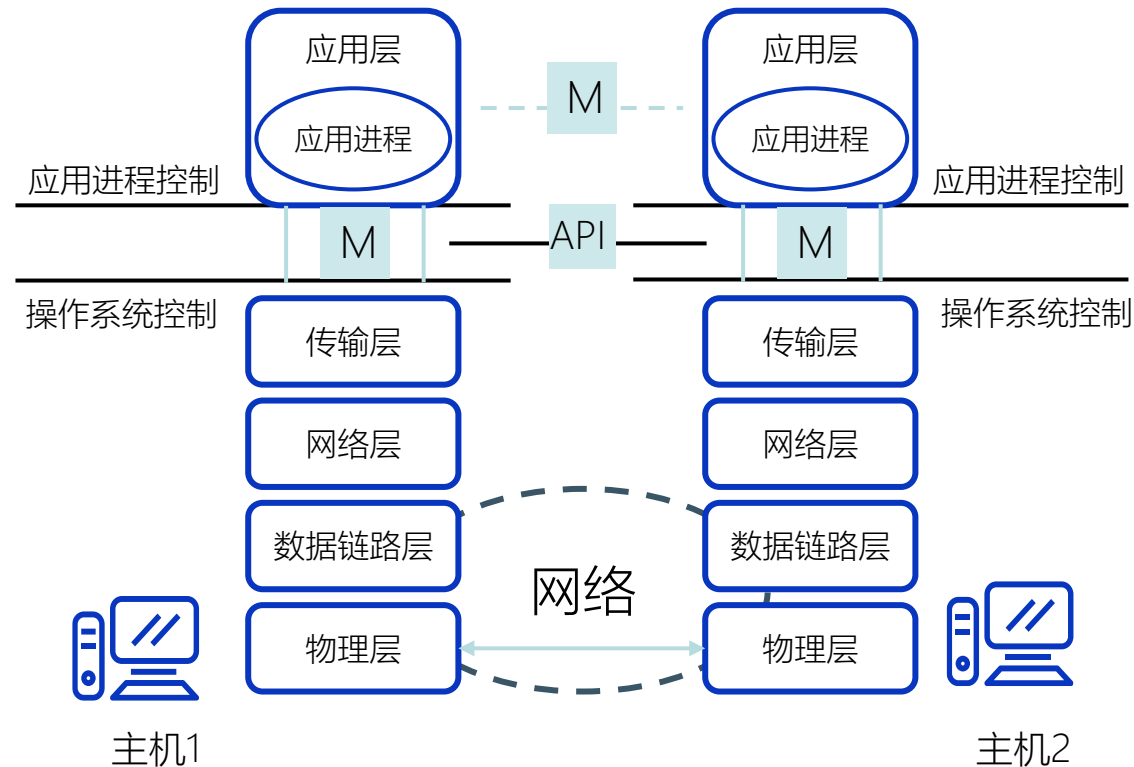
- 7.1 客户服务器模型
- 7.2 DNS服务
- 7.3 电子邮件服务
- 7.4 FTP服务
- 7.5 Web服务

# 客户与服务器模型

| 服务器应用进程  | 客户应用进程   |
|--|--|
| <ol style="list-style-type: none"><li>1. 首先运行</li><li>2. 不需要知道哪个客户将连接它</li><li>3. 被动等待来自客户的连接请求，且等待时间任意长</li><li>4. 通过发送和接收数据来与客户进行通信</li><li>5. 在实现对一个客户的服务后，维持运行并等待另一个请求</li></ol> | <ol style="list-style-type: none"><li>1. 随后运行</li><li>2. 必须知道想要连接的服务器</li><li>3. 在需要通信的任何时候，发起连接请求</li><li>4. 通过发送和接收数据来与服务器进行通信</li><li>5. 在完成与服务器的交互后，可以终止运行</li></ol> |

# 应用层协议的特点

- 应用层协议是为了解决某一类具体应用问题
  - 往往是通过位于不同主机中的多个应用进程之间的通信和协同工作来完成的
- 应用层的具体内容就是规定应用进程在通信时所遵循的协议
  - 专用通信、标准化服务
- 应用层的许多协议都是基于客户服务器方式
  - 客户服务器方式所描述的是进程之间服务和被服务的关系。
- 客户(client)和服务端(server)都是指通信中所涉及的两个应用进程
  - 客户是服务请求方，服务器是服务提供方



# 基本通信模式

## ■ 2种基本通信模式

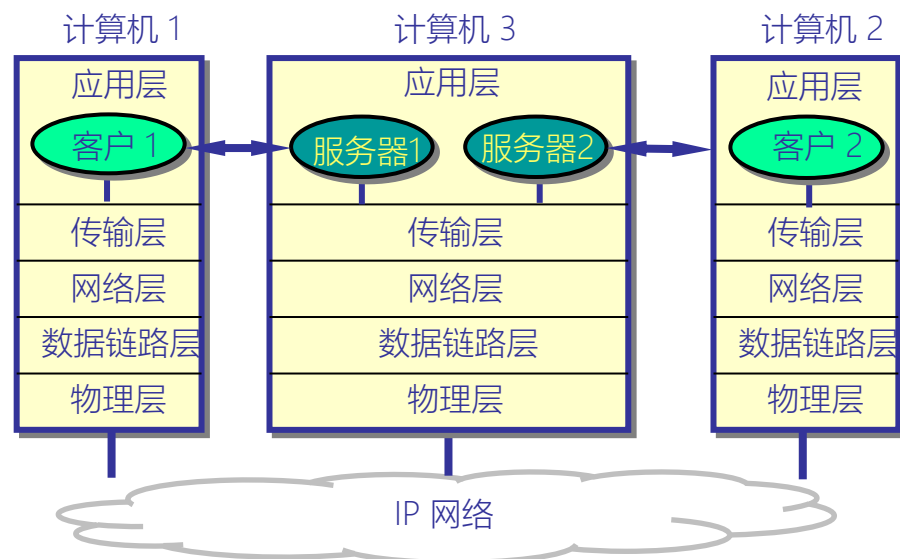
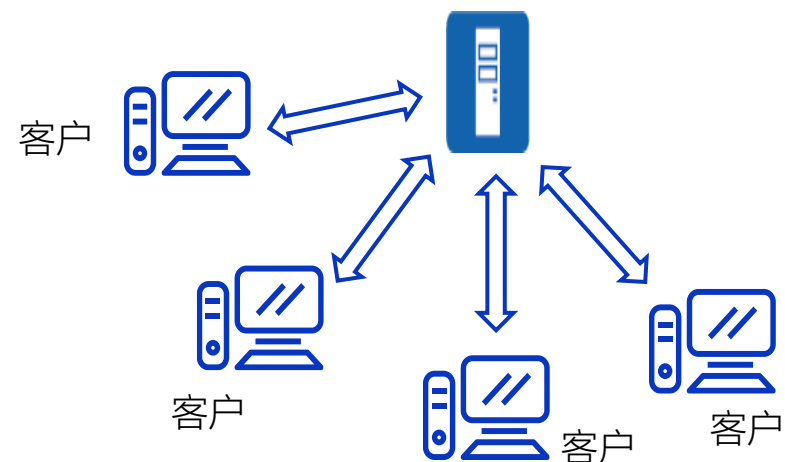
| 流模式   | 报文模式   |
|---|--|
| <ul style="list-style-type: none"><li>1. 面向连接</li><li>2. 一对一通信</li><li>3. 字节序列</li><li>4. 任意长度传输</li><li>5. 大多数应用使用</li><li>6. 构建在TCP之上</li></ul> | <ul style="list-style-type: none"><li>1. 无连接</li><li>2. 多对多通信</li><li>3. 报文序列</li><li>4. 每个报文限制最多64KB</li><li>5. 用于多媒体应用</li><li>6. 构建在UDP之上</li></ul> |

# 应用进程间通信方式分类

- 位于不同主机上的进程间的通信有三种通信方式：
  - C/S: Client/Server, 客户/服务器方式
  - B/S: Browser/Server, 浏览器/服务器方式
  - P2P: Peer to Peer, 对等方式

# C/S 方式架构

- 客户/服务器（Client/Server）模式：  
一种主从式架构
  - 服务器是主，客户端是从
  - 服务器软件以多进程方式运行，可以同时为多个客户服务
  - 在移动互联网环境下，每个应用APP都是一个客户端
- C/S 是一个可缩放的架构
- 服务器软件通常运行在性能强大的专用计算机上，承担复杂的业务逻辑处理、数据计算与存储等
- 客户端一般运行于普通个人计算机上，为用户提供操作与显示等



# C/S方式应用

- 应用层的许多协议是基于C/S方式，如传统的 Internet 的文件传输、电子邮件、远程登录等。
  - 客户(client)和服务端(server)是指通信中所涉及的2个应用进程
  - 客户/服务器方式描述的是应用进程之间服务和被服务的关系
  - 客户是服务请求方（主动请求服务，被服务）
  - 服务器是服务提供方（被动接受服务请求，提供服务）
- 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。

# C/S方式特点

## ■ 客户端进程的特点

- 在进行通信时临时成为客户，它也可在本地进行其它的计算
- 用户计算机上运行，在打算通信时主动向远地服务器发起通信
- 客户方必须知道服务器进程所在主机的IP地址才能发出服务请求
- 需要时可以与多个服务器进行通信

## ■ 服务器进程的特点

- 专门用来提供某种服务的程序，可“同时”处理多个远地或本地客户的请求
- 必须始终处于运行状态才有可能提供服务
- 通信开始之前服务器进程不需要知道客户进程所在主机的IP地址，无论客户请求来自哪里，服务器进程被动等待服务请求的到来即可
- 当系统启动时即自动调用并一直运行着，也可以由用户或其它进程在通信前启动
- 被动等待并接受来自多个客户的通信请求



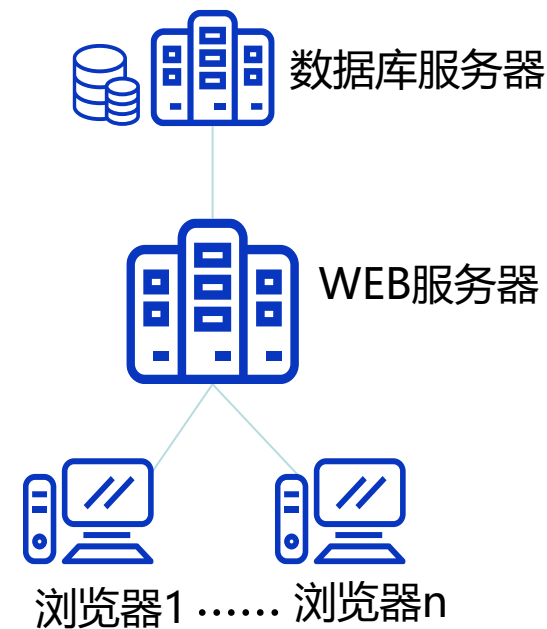
# 应用进程间的通信方式：B/S方式

- 浏览器/服务器（ Browser/Server ）模式：可看做是C/S方式的特例，客户端软件改为浏览器
- B/S方式采取浏览器请求、服务器响应的工作模式
- 在B/S方式下，用户界面通过Web浏览器实现
  - 一部分简单的事务逻辑可以在客户端实现
  - 主要的、复杂的事务逻辑在服务器端实现



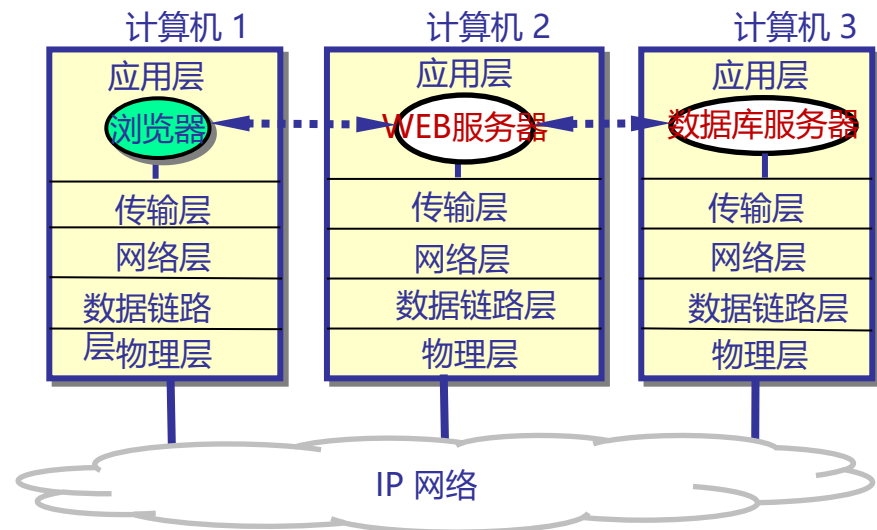
# B/S方式的架构

- 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，其任务是接受Web服务器对数据库服务器提出的数据操作请求，然后由数据库服务器进行数据处理并把处理结果返回给web服务器
- 处理层：由Web服务器承担业务处理逻辑和页面存储管理，接受客户浏览器的任务请求，执行相应的事务处理
- 展现层：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
- 实际部署时也可以把数据库服务器和web服务器部署在同一台设备上

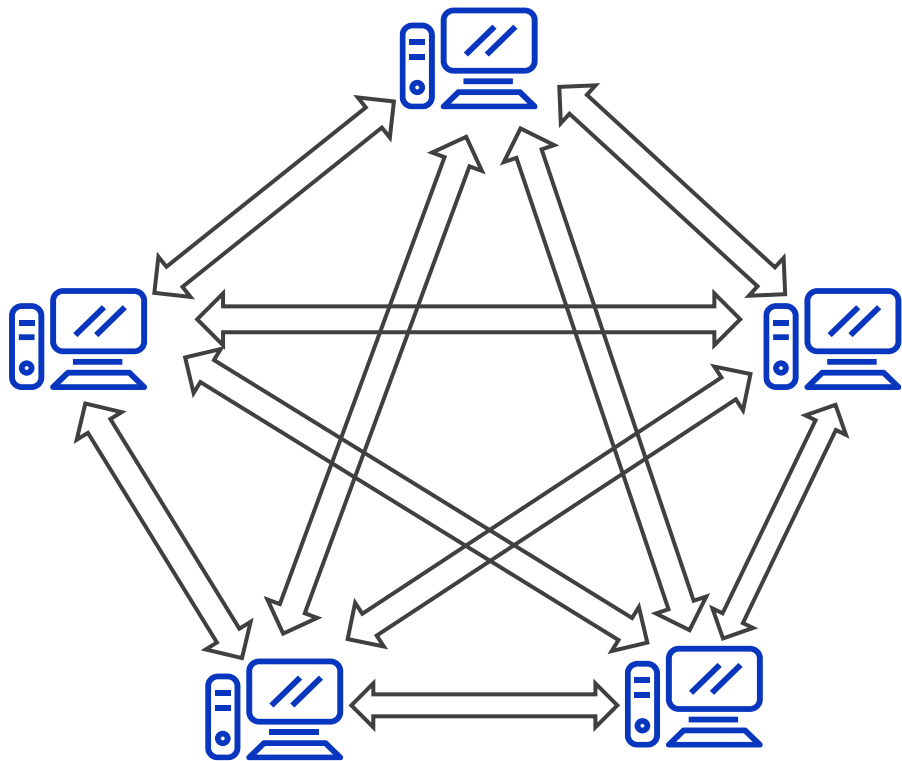


# B/S方式的特点

- 界面统一，使用简单：客户端只需要安装浏览器软件
- 易于维护：对应用系统升级时，只需更新服务器端的软件，方便了系统维护和升级
- 可扩展性好：采用标准的TCP/IP和HTTP协议，具有良好的扩展性
- 信息共享度高：HTML是数据格式的一个开放标准，目前大多数软件均支持HTML
- 注意：在一种浏览器环境下开发的界面在另一种浏览器环境下可能有不完全适配的情况，这时需要安装对应的浏览器



# 应用进程间的通信方式：P2P方式



P2P 模型

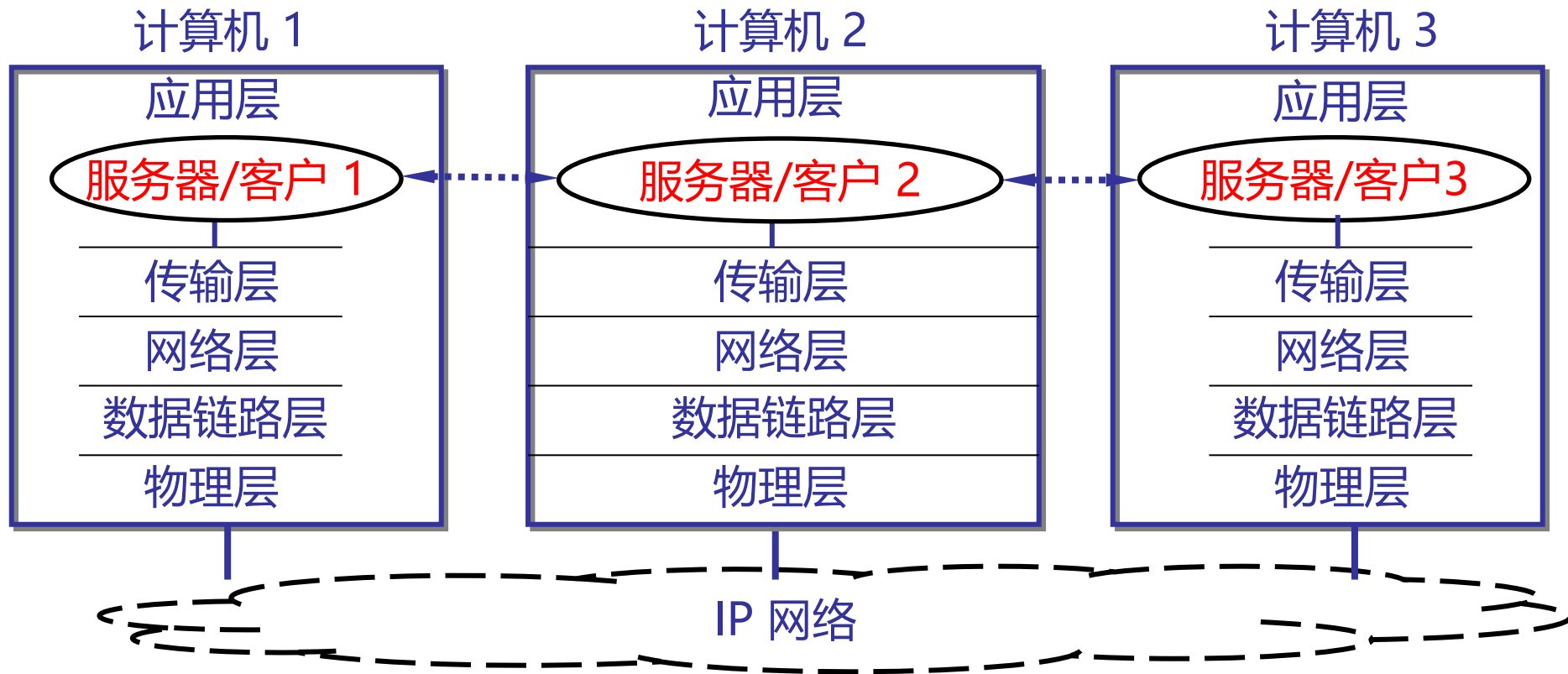
- P2P是Peer to Peer对等方式，属于分布式模型
- 每个主机既可以提供服务，也可以请求服务
- 任意端系统/结点之间可以直接通信。
- 结点间歇性接入网络，不能保证结点永远在线
- 结点的 IP 地址可能会发生变化

# P2P方式

- 处于不同主机上的两个 P2P 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，如果权限允许的话
- 音频/视频应用推动了 P2P 对等通信方式的发展
- 因特网上流量，音频/视频流量已占主要比例
- 音视频采用 P2P对等通信方式，减轻了服务器负载，也影响了网络上的流量模型，从垂直方向流量改为很大一部分是水平方向流量

# P2P方式

- P2P方式从本质上看仍然是使用了C/S方式，但强调的是通信过程中的对等
  - 每一个P2P进程既是客户同时也是服务器

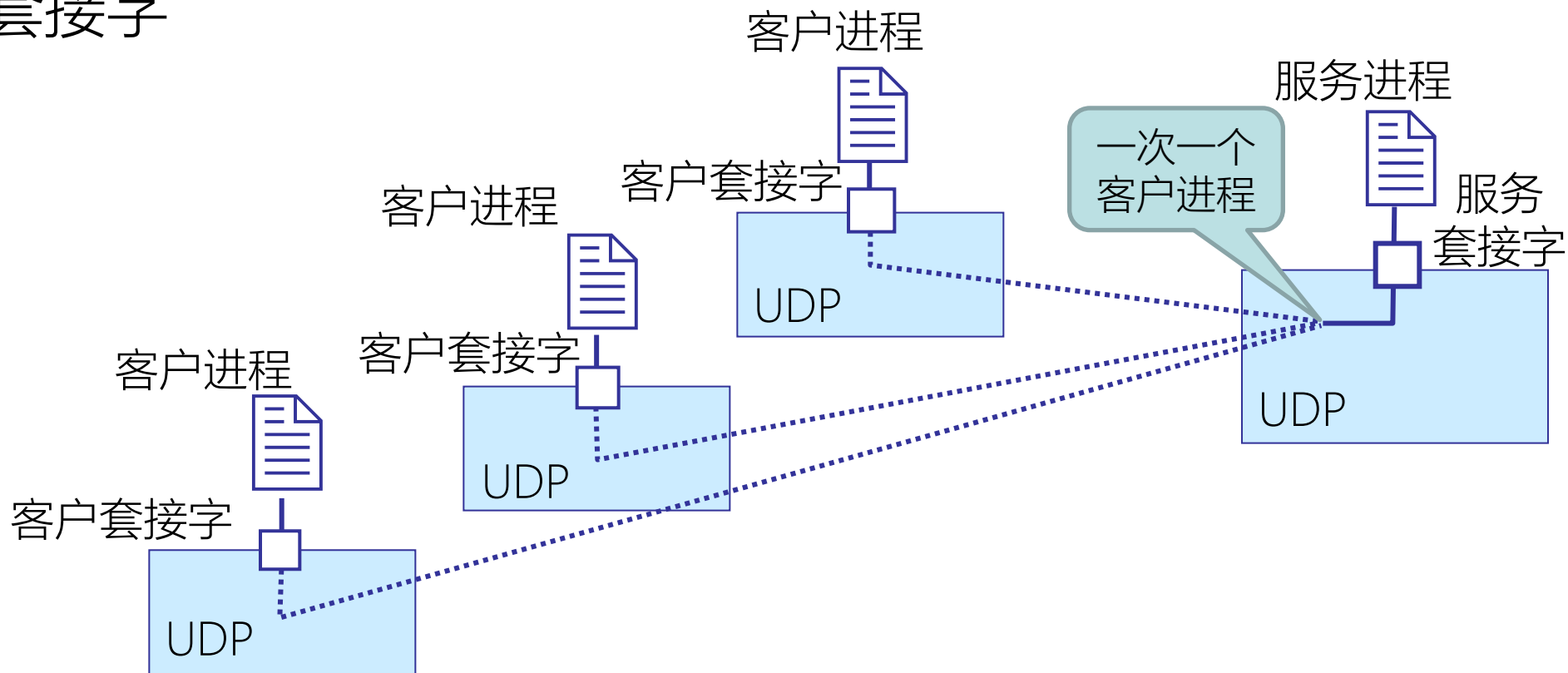


# 服务器进程工作方式

- 服务器上的进程有两种工作方式
- 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 (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 (非阻塞方式)

# 无连接循环方式服务

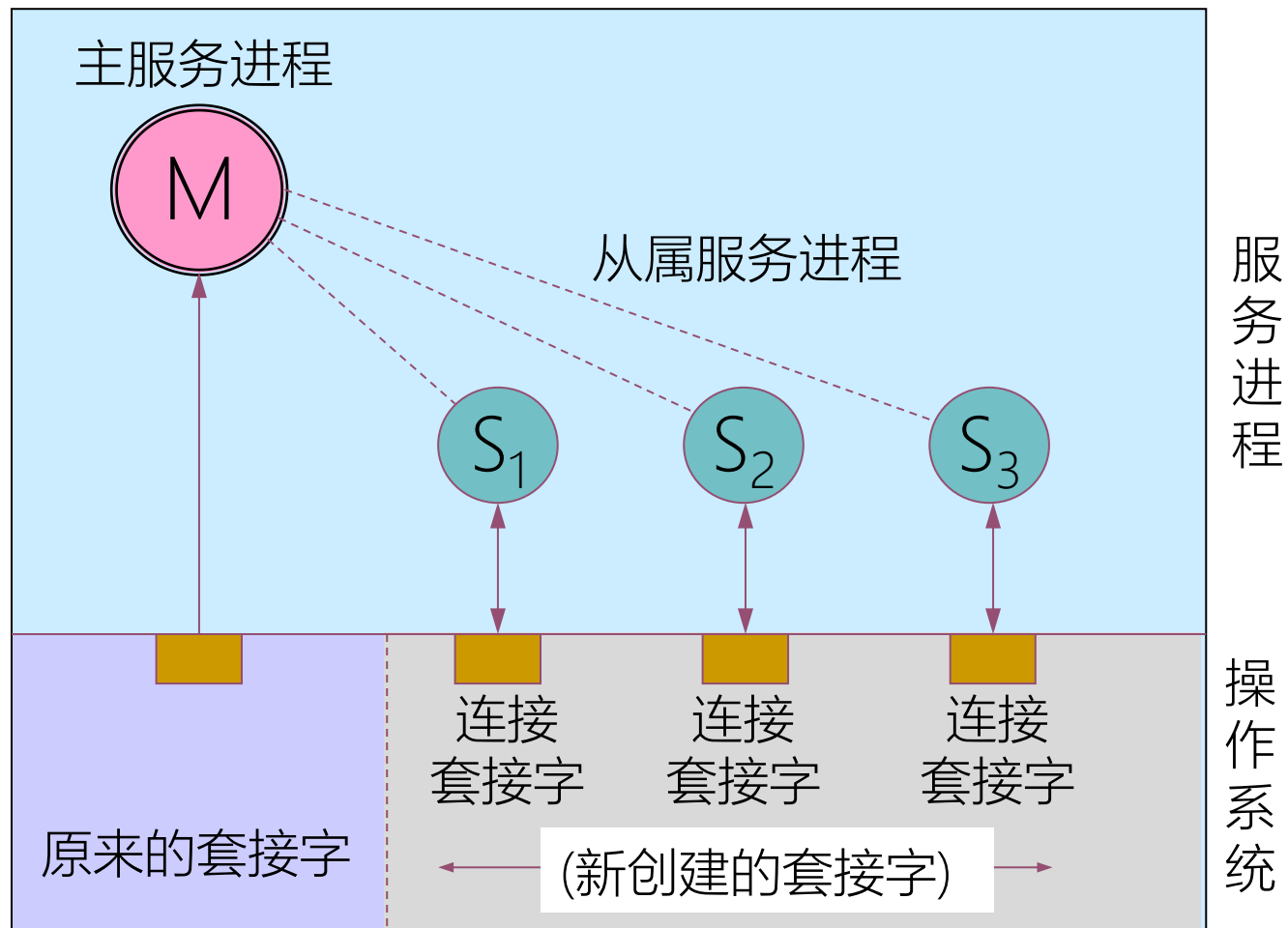
- 使用无连接的UDP服务进程通常都工作在循环方式，即一个服务进程在同一时间只能向一个客户进程提供服务。(顺序服务)
- 服务进程只使用一个服务套接字。每个客户则使用自己设定端口号的客户套接字



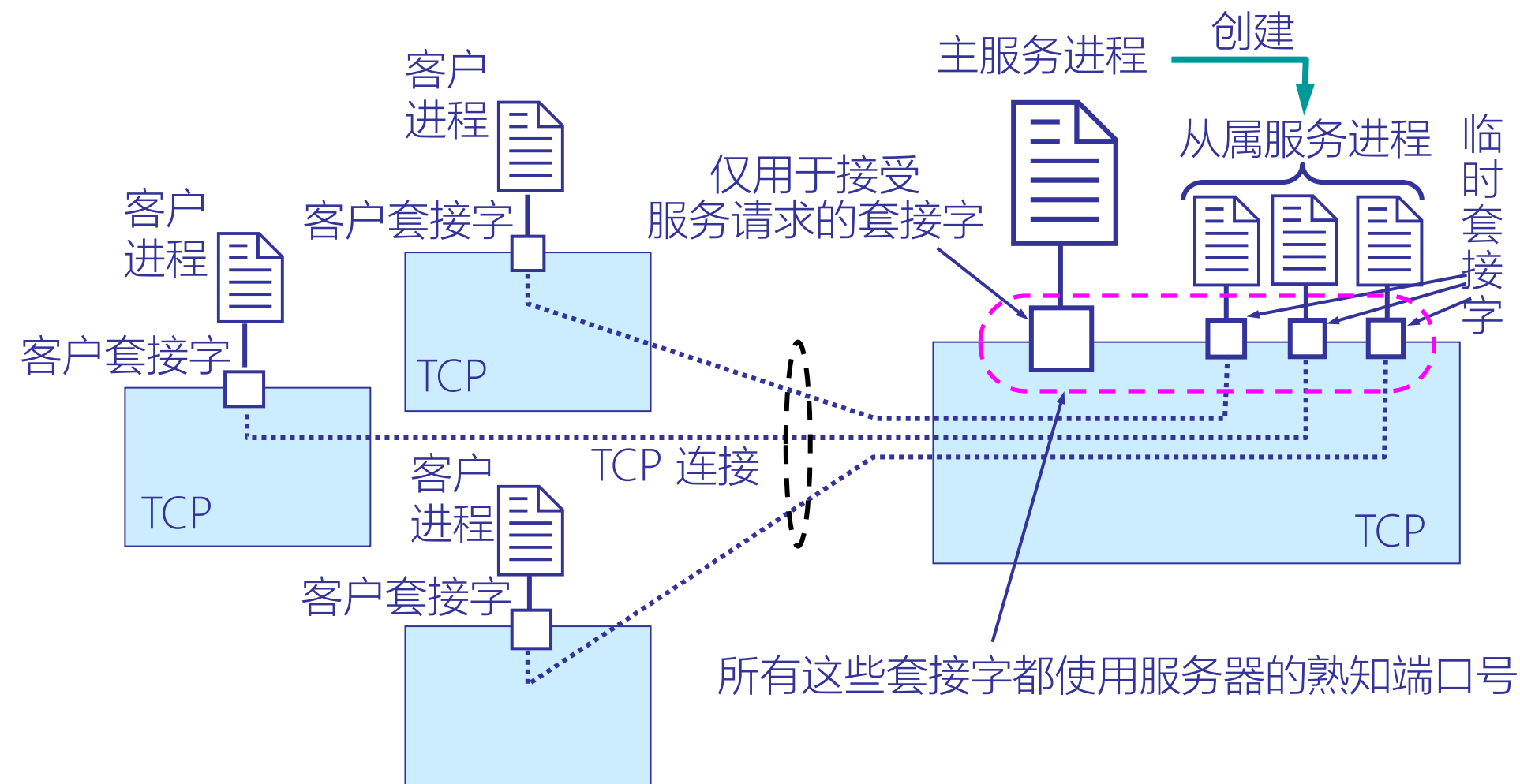


# 面向连接的并发方式服务

- 面向连接的TCP服务进程通常都工作在并发服务方式，服务进程在同一时间可同时向多个客户进程提供服务。(并发服务)
- 在TCP服务进程与多个客户进程间需要建立多条TCP连接时，每条TCP连接在其数据传送完毕后释放
- 一个TCP连接对应一个（熟知）服务端口
- 主服务进程在熟知端口等待客户进程发出的请求



# 面向连接的并发方式服务



主服务进程也称为父服务进程，而从属服务进程又称为子服务进程。

# DNS服务

- DNS(域名系统、域名服务、域名服务器)
  - 一种能够完成从主机名字到IP地址或从IP地址到主机名字的映射
  - 给计算机取一个有语义的名字，容易记忆
  - 通信时，仍然需要用IP地址来通信
- 由于域名具有文字表达的意义，比IP地址更容易记忆。



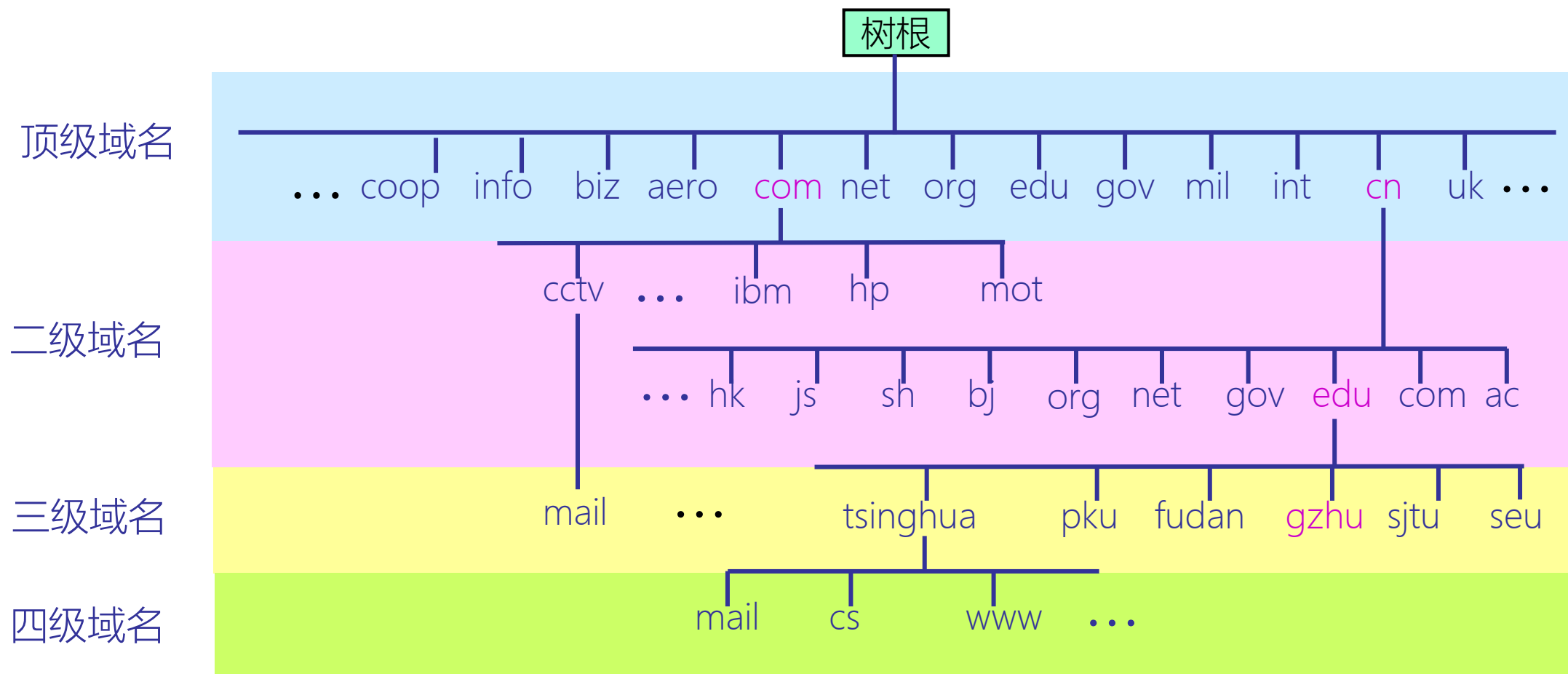
# DNS的历史

- ARPANET时期：HOSTS方式
  - ARPANET的域名与IP地址的映射是依靠一个文本文件（HOSTS）来实现  
HOSTS文件包含了主机的名称到地址的映射
- 互联网的规模发展
  - HOSTS 文件机制不具备可缩放性，限制了网络规模
  - 需要集中管理来防止主机名冲突
- DNS的出现
  - 1984年USC（University of Southern California）发布了RFC882-883，定义了DNS，形成了现在的域名系统标准
- 采用了一种层次的、基于域的命名模式，并使用分布式数据库系统实现

# 域名系统概述

- 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
- Internet采用层次结构的命名树作为主机的名字, 并使用分布式的联机分布式数据库系统
- 域名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结点上运行
  - 相应的结点称为名字服务器(Name Server)或域名服务器(Domain Name Server)

# 域名系统名字空间的层次化结构



域名系统中的名字空间采用层次化的树状结构，按级划分

# 命名方法

- Internet的域名结构采用了层次化的树状结构的命名方法
- 域名的结构由若干个分量组成，各分量之间用小数点(.)隔开，总长不超过255个字符
- 各分量分别代表不同级别的域名(每个分量域名长度 $\leq 63$ 字符)
- 合法域名中，点“.”的个数至少为一个
- 通常，点“.”对应的英文单词为dot，也可以读为point

... .三级域名.二级域名.顶级域名

# 顶级域名分类

- 顶级域名TLD (Top Level Domain) 有三类
- 国家或地区顶级域nTLD, 也记为ccTLD (cc: country code)
  - 例如.cn 表示中国, .us 表示美国, .uk 表示英国。目前有300多个
- 基础设施域.arpa (Address and Routing Parameter Area)
  - 专用于Internet基础设施目的
  - 目前有二级域ip6.arpa; iris.arpa; in-addr.arpa; uri.arpa; urn.arpa; home.arpa; as112.arpa; in-addr-servers.arpa; ipv4only.arpa等
- 通用顶级域gTLD
  - 早期规定了20个通用顶级域名, 2011年批准新通用顶级域名(New Generic Top-level Domain, New gTLD)
  - 截至2020年, 已注册有1200多个通用顶级域名
- 可以注册 .中国、.公司、.网络 等顶级域名(多语种域名国际标准 RFC3454、RFC3490、RFC3491、RFC3492)

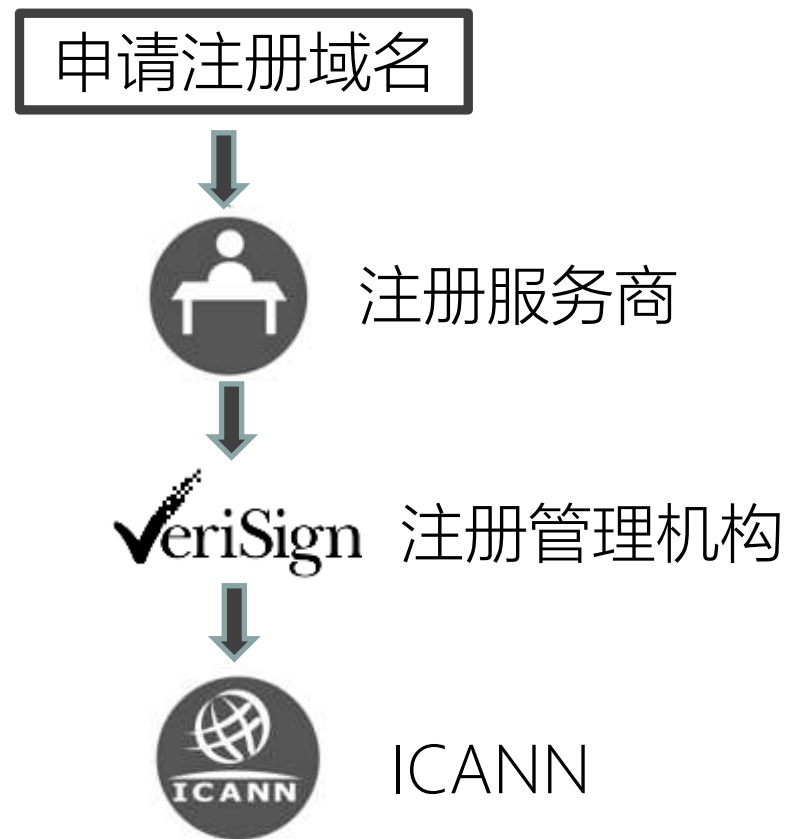


# 国家顶级域名

- 国家顶级域名 .cn下的二级域名分为三类
- 类别域名7个：
  - .edu.cn 教育
  - .gov.cn 政府
  - .org.cn 非营利组织
  - .net.cn 网络服务
  - .com.cn 工商金融等企业
  - .ac.cn 科研
  - .mil.cn 国防机构
- 行政区域名34个：省、直辖市、自治区、特区等行政区域名，每个行政区域名为两个字母，例如：北京-bj、吉林-jl等
- 无类别域名：例如 www.google.cn、www.tianya.cn等

# 域名管理方法

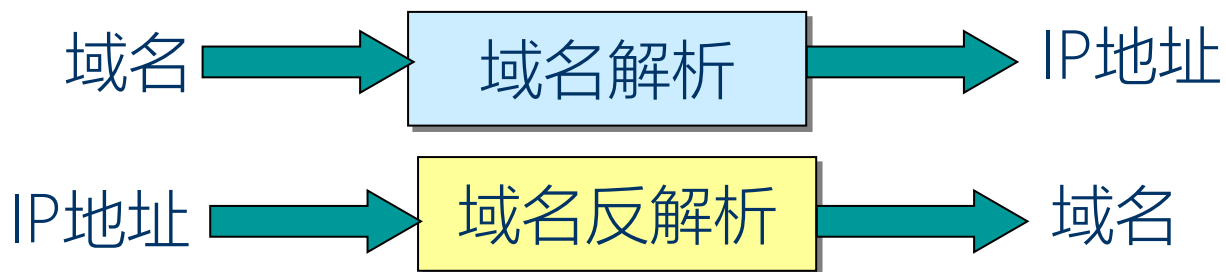
- 域名管理机构分级负责域名注册
  - Internet的域名管理机构：ICANN  
(Internet Corporation for Assigned Names and Numbers) [www.icann.org](http://www.icann.org)
  - ccTLD下的二级域名该国自行确定
  - 三级域名注册由其所属二级域名机构负责，  
以此类推
- .edu.cn下三级域名注册由CERNET负责
- 我国的其它二级域名注册由中国互联  
网络信息中心(CNNIC)负责



注册管理机构与注册服务商之间的关系

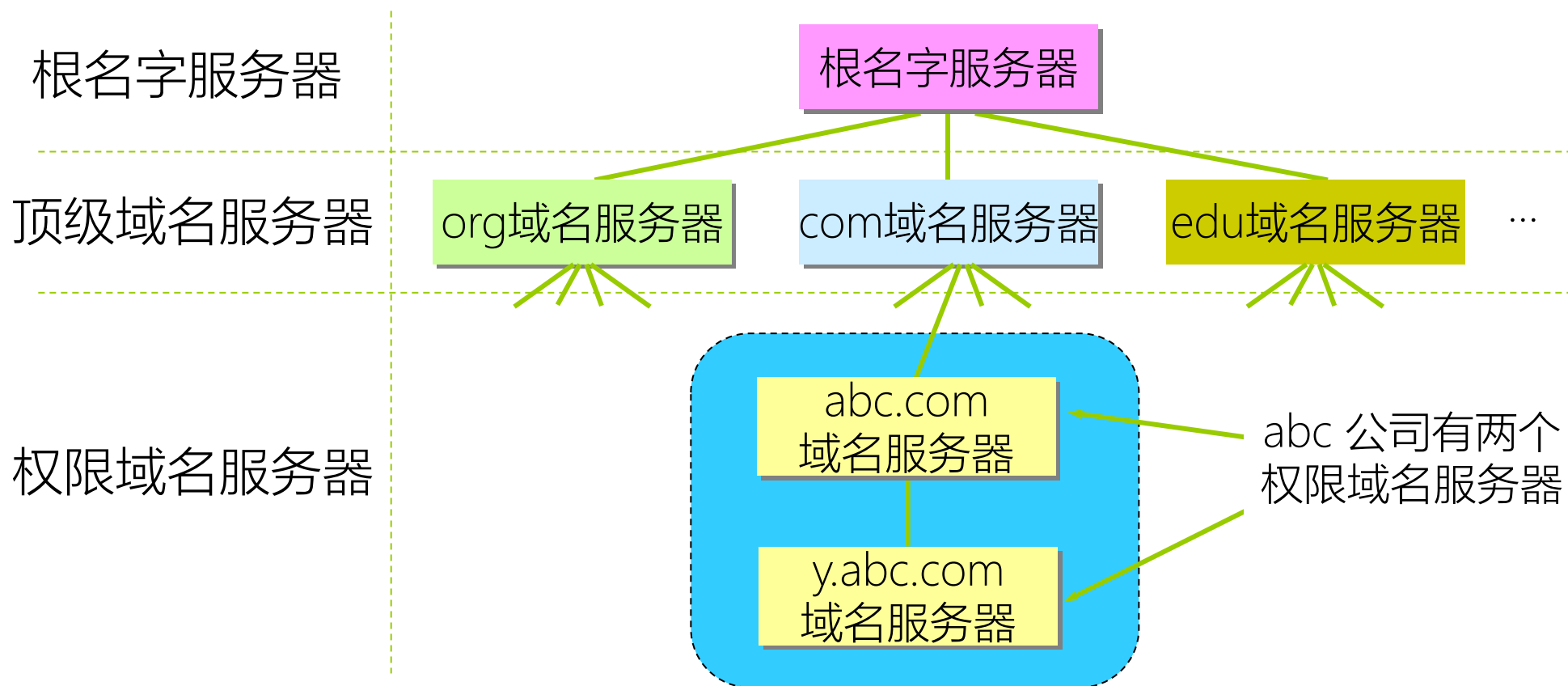
# 域名系统

- 域名系统是一个分布式的数据库，每个子域负责维护整个数据库的一个分段。
- DNS采用客户/服务器工作模式
  - 负责进行名字解析的服务器统称为名字服务器或域名服务器
  - 每个域名服务器必须具有连向其它有关域名服务器的信息
  - 当自己不能完成域名与IP地址的转换时，能够知道或使其它域名服务器知道到哪里去找别的域名服务器，使域名解析过程能够完成
- 域名与IP地址可以是一对一、一对多或者多对一的关系
- 域名解析过程对用户透明



# 域名服务器分类

## ■ 域名服务器按层次化树状结构的部署方式



# 根服务器

<https://root-servers.org/>



As of 2019-12-15, the root server system consists of 1031 instances operated by the 12 independent root server operators.

As of 2023-10-18T08:59:07Z, the root server system consists of 1755 instances operated by the 12 independent root server operators.

# 根服务器列表

<https://www.iana.org/domains/root/servers>

| HOSTNAME           | IP ADDRESSES                      | MANAGER                                 |
|--------------------|-----------------------------------|---|
| a.root-servers.net | 198.41.0.4, 2001:503:ba3e::2:30   | VeriSign, Inc.                          |
| b.root-servers.net | 199.9.14.201, 2001:500:200::b     | University of Southern California (ISI) |
| c.root-servers.net | 192.33.4.12, 2001:500:2::c        | Cogent Communications                   |
| d.root-servers.net | 199.7.91.13, 2001:500:2d::d       | University of Maryland                  |
| e.root-servers.net | 192.203.230.10, 2001:500:a8::e    | NASA (Ames Research Center)             |
| f.root-servers.net | 192.5.5.241, 2001:500:2f::f       | Internet Systems Consortium, Inc.       |
| g.root-servers.net | 192.112.36.4, 2001:500:12::d0d    | US Department of Defense (NIC)          |
| h.root-servers.net | 198.97.190.53, 2001:500:1::53     | US Army (Research Lab)                  |
| i.root-servers.net | 192.36.148.17, 2001:7fe::53       | Netnod                                  |
| j.root-servers.net | 192.58.128.30, 2001:503:c27::2:30 | VeriSign, Inc.                          |
| k.root-servers.net | 193.0.14.129, 2001:7fd::1         | RIPE NCC                                |
| l.root-servers.net | 199.7.83.42, 2001:500:9f::42      | ICANN                                   |
| m.root-servers.net | 202.12.27.33, 2001:dc3::35        | WIDE Project                            |

# 权限域名服务器

- 每个区设置相应的权限域名字服务器，用来保存该区中的所有主机的域名到IP地址的映射
- 权限域名字服务器一般只解析本辖域的域名
  - 每一个主机都必须在某个权限域名字服务器处注册登记
  - 权限域名字服务器知道其管辖的主机名应当转换成什么IP地址
  - 各个单位根据自己的具体情况把本单位的域名划为若干个域名管辖区(zone)，也可简称为区
- 当一个权限域名服务器还不能给出最后的查询回答时，就会告诉发出查询请求的 DNS 客户，下一步应当找哪一个权限域名服务器

# 本地域名服务器

- 每一个Internet服务提供者ISP(Internet Service Provider), 都至少有一个本地域名服务器, 称为默认域名服务器
- 本地域名服务器离用户较近, 一般不超过几个路由器的距离
- 当主机发出DNS查询报文时, 这个查询报文就首先被送往该主机所在区域的本地域名字服务器
- 如果所要查询的主机也处在本地 ISP 的管辖范围, 则本地域名字服务器就立即能进行域名解析, 返回IP。否则就需要再以此字去询问其他的域名服务器
- 实际部署中, 解析请求路径上的本地域名服务器可能有多



# 域名系统组成

## ■ 域名空间

- 域名全称是一个从该域到根的标签序列，以“.”分离这些标签
- 域名即可以是叶结点的名称，也可是子树的根结点的名称

## ■ 命名规则

- 最高层的域名由网络信息中心指定
- 第3层域名通常表示组织部门或分支

## ■ 代理技术：代理包含2个意思

- 数据存储分散化
- 管理权分散化

# DNS的工作原理

## ■ 解释过程：

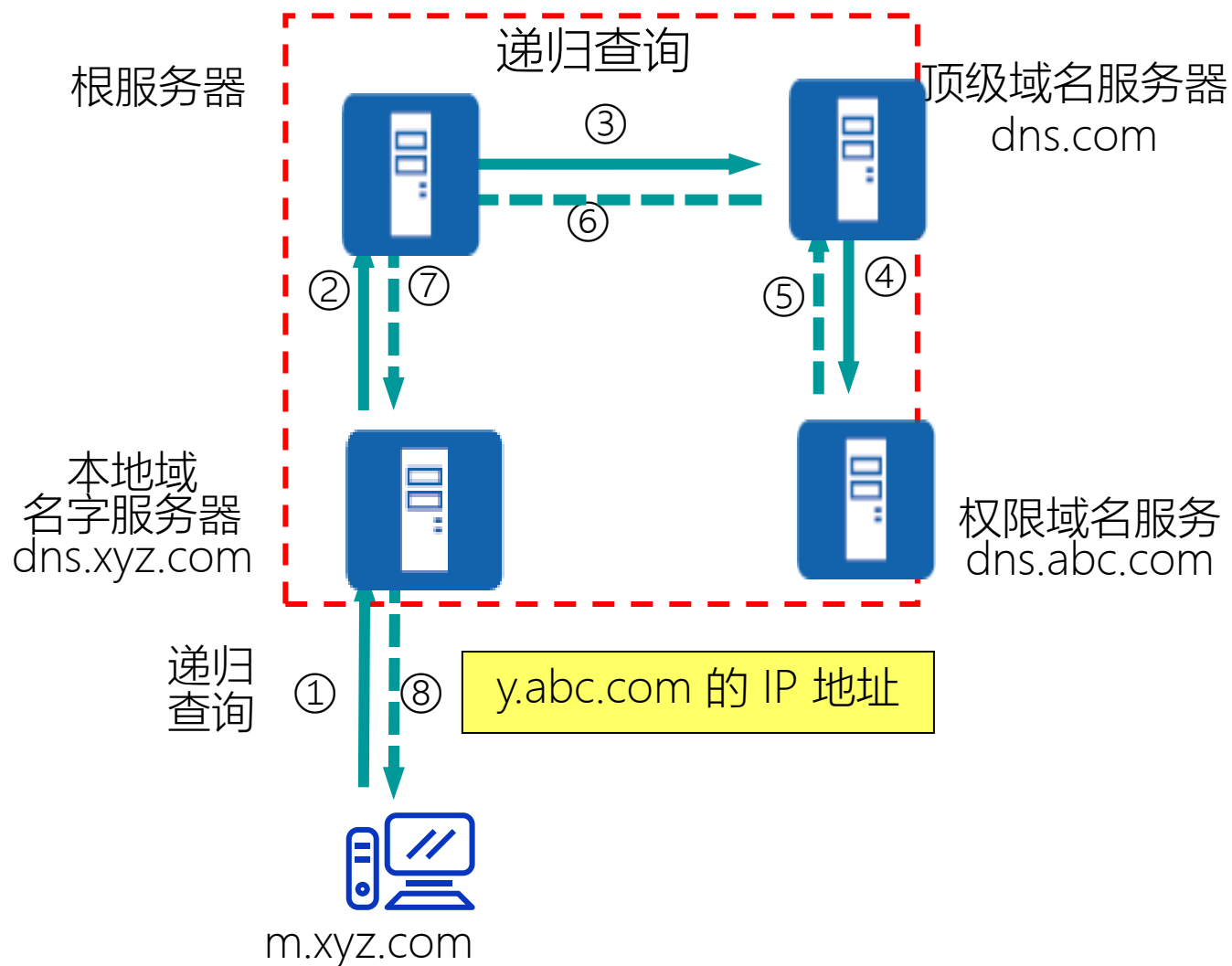
- 域名解析器首先查询本地主机的缓冲区，查看主机是否以前解析过主机名。
- 如果主机缓冲区中没有与其IP地址的映射关系，解析器将向本地域名服务器发出请求。
- 本地域名服务器首先检查域名与其IP地址的映射关系是否存储在它的数据库中，如果是，本地服务器将该映射关系传送给请求者。

## ■ 解释方法有两种：

- 重复解释
- 递归解释

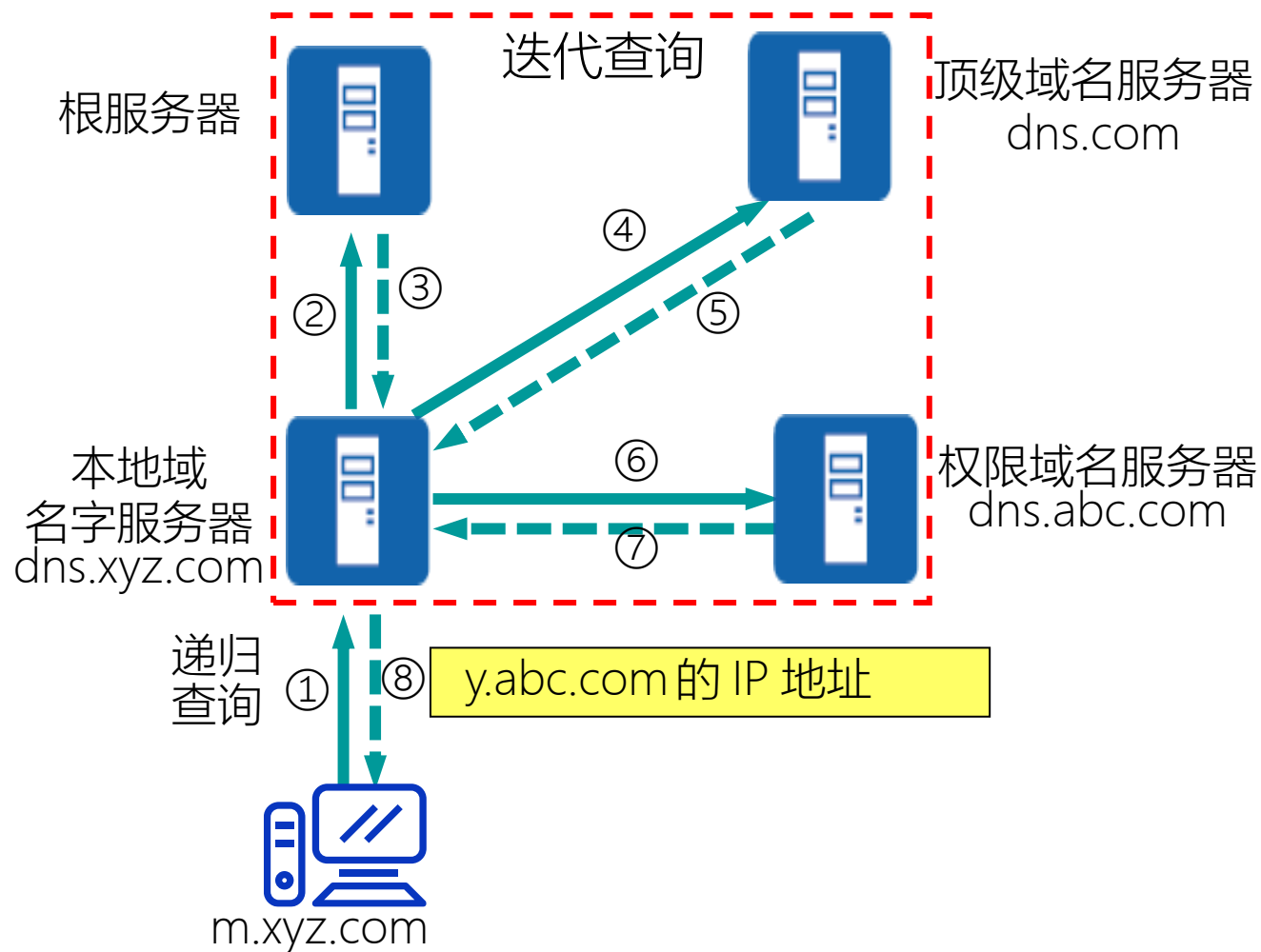
# 递归查询

- 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，即替本地域名服务器继续查询。
- 较少使用

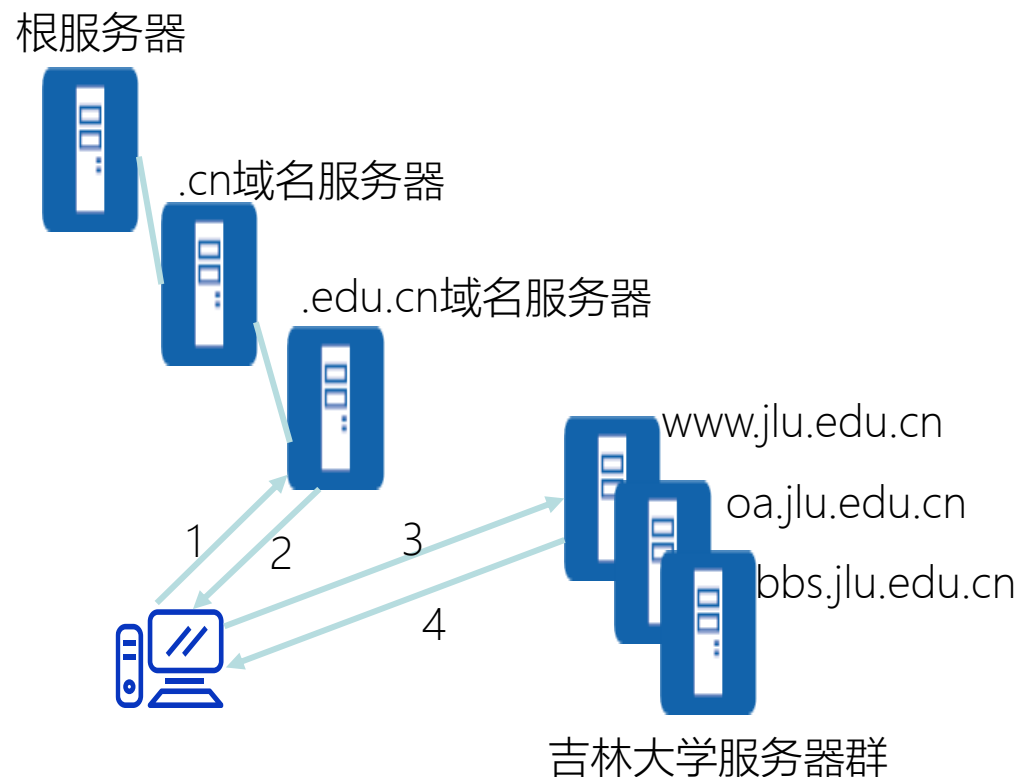


# 迭代查询(重复解释)

- 被查询的域名服务器检索其本地数据（包括缓存中的数据）找寻所要求的数据。
- 域名服务器未找到所需的数据，它将尽力返回可以帮助查询者继续解析过程的答案。
- 通常使用

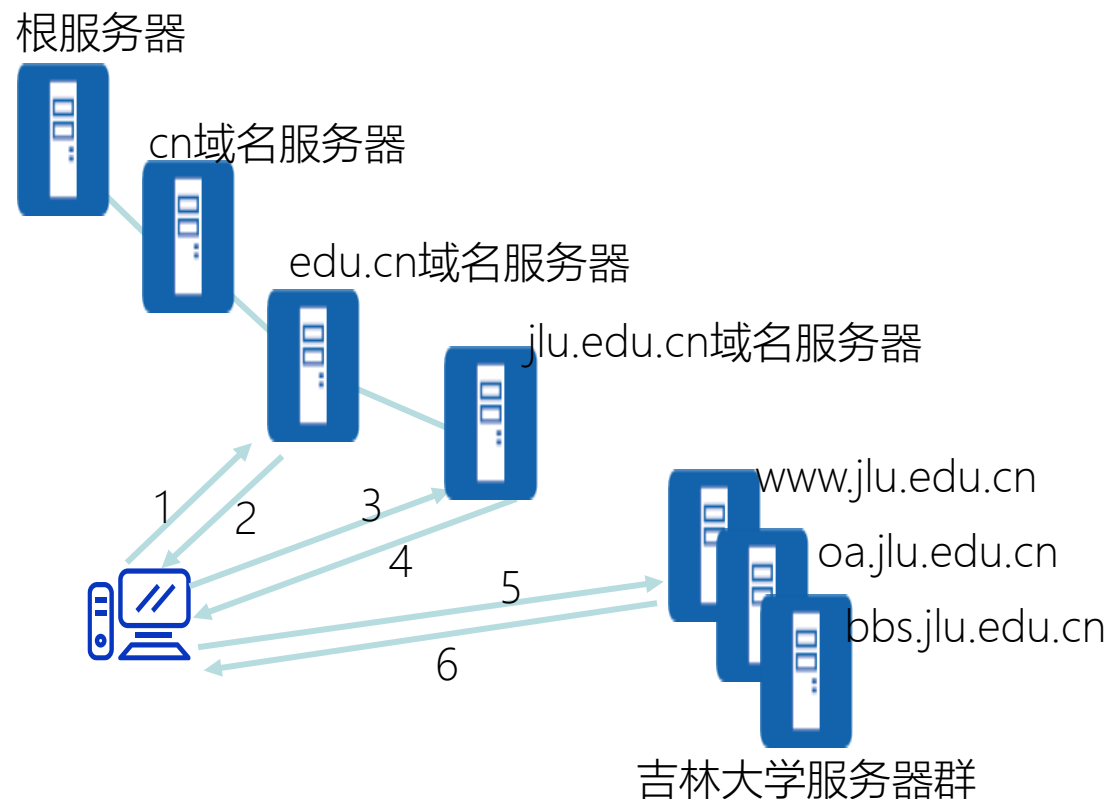


# 域名解析过程示例：无子域代理情形



1. 客户机向edu.cn域名服务器发出域名解析请求，以获得域名www.jlu.edu.cn的IP地址；
2. 服务器向客户机返回www.jlu.edu.cn的IP地址202.198.16.80；
3. 客户机用查询而得的IP地址向www.jlu.edu.cn发出http请求；
4. 吉林大学www服务器接受请求，并返回结果给客户机。

# 域名解析过程示例：有子域代理情形



1. 客户机向edu.cn域名服务器发出域名解析请求，以获得域名www.jlu.edu.cn的IP地址；
2. edu.cn域名服务器发觉所请求的地址已经代理出去，于是返回jlu域名服务器的IP的地址；
3. 客户机向jlu.edu.cn域名服务器查询www.jlu.edu.cn的IP地址；
4. jlu.edu.cn域名服务器返回www.jlu.edu.cn的IP地址202.198.16.80；
5. 客户机用查询而得的IP地址向www.jlu.edu.cn发出http请求；
6. 吉林大学www服务器接受请求，并返回结果给客户机。

# DNS相关工具

- 用于查询域名相关信息
- nslookup
  - nslookup domain-name
- dig (domain information groper)
- host
- 例：Windows系统查看DNS缓存：
  - ipconfig /displaydns

# 域名系统高速缓存

- 每个域名服务器都维护一个高速缓存，存放最近用过的名字以及从何处获得名字映射信息的记录
  - 可大大减轻根域名服务器负荷，使互联网上 DNS 查询请求和回答报文的数量大为减少
  - 能够使Internet上的DNS查询请求和回答报文的数量大为减少
- 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，并处理超过合理时间的项（例如，每个项目只存放两天）。
- 当权限域名服务器回答一个查询请求时，在响应中都指明绑定有效存在的时间值
  - 增加此时间值可减少网络开销
  - 减少此时间值可提高域名转换的准确性



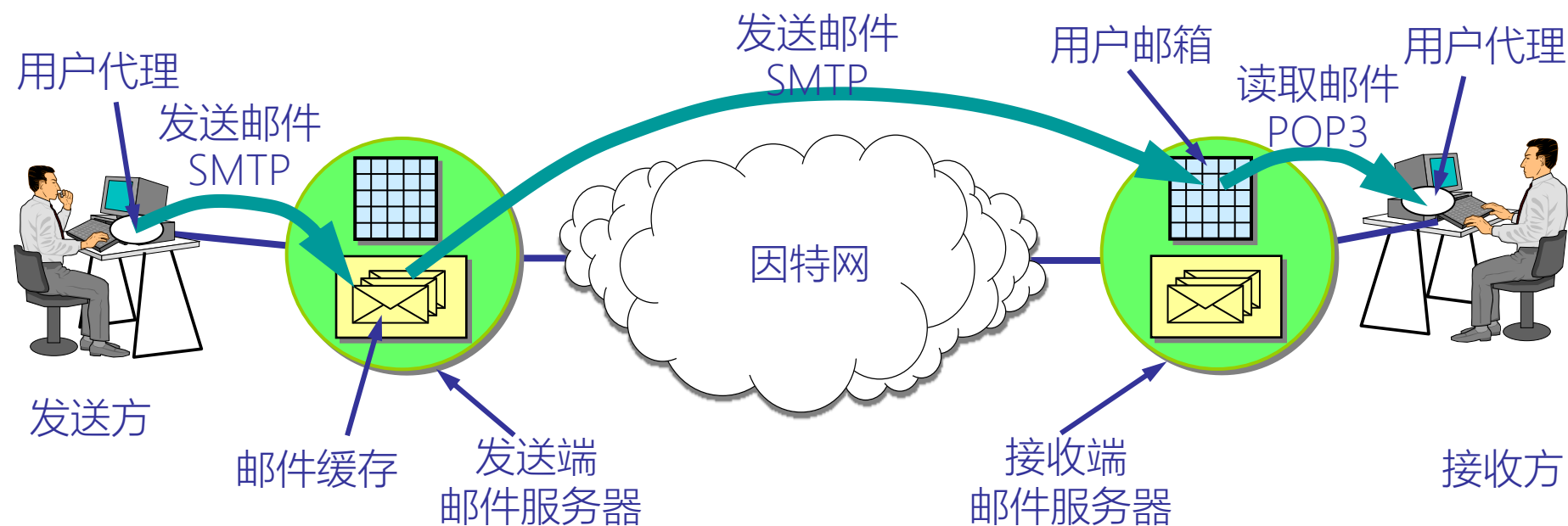
# DNS vs ARP

- DNS是应用层地址与网络层地址之间映射关系的翻译
- ARP是网络层地址与数据链路层地址之间映射关系的翻译
- 网络体系结构层次的独立性决定两者存在的必要性

# 电子邮件服务

- Email是Internet上使用最多的一种应用。
- 一个电子邮件系统由3部分组成：
  - 用户代理：用户与电子邮件系统的接口，是用户机上运行的程序。
  - 邮件服务器：收发邮件，运行邮件服务程序。
  - 电子邮件协议：SMTP、POP3
- 电子邮件不是一种“终端到终端”的服务，而是一种“存贮转发式”服务。

# 电子邮件的主要组成



# 简单邮件传输协议

- 简单邮件传输协议--SMTP(Simple Mail Transfer Protocol)
- SMTP使用客户服务器模式，负责发送邮件的SMTP进程是SMTP客户，负责接受邮件的SMTP进程是SMTP服务器。
- SMTP规定了14条命令和21种响应信息。

# SMTP 协议：基于 TCP 通信的三个阶段

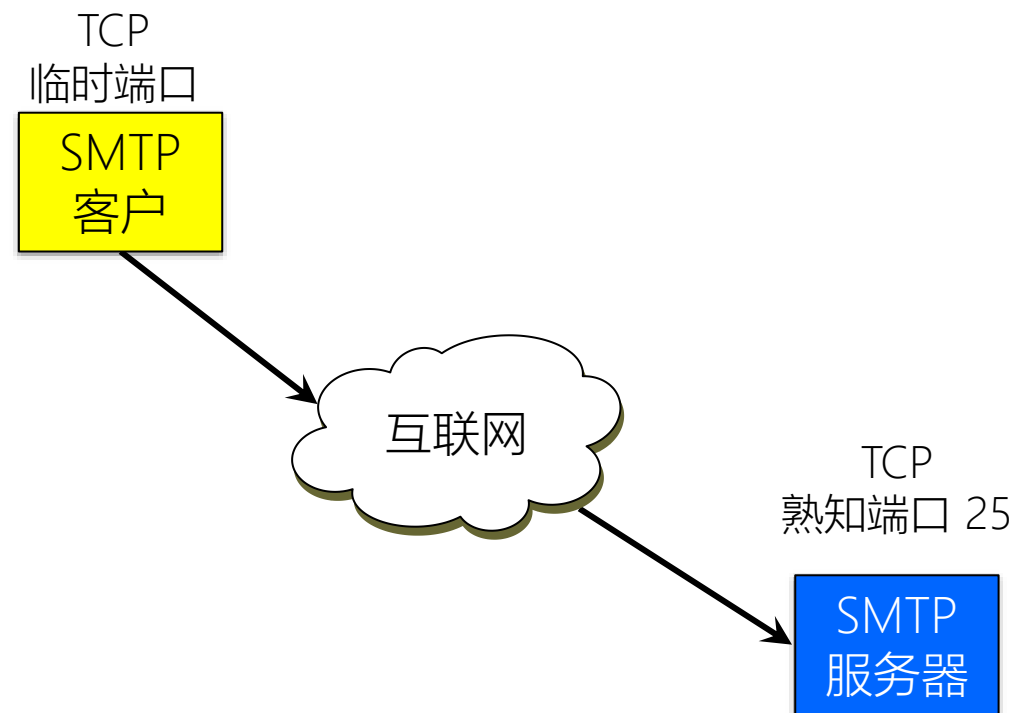
## ■ 连接建立

- 连接是在发送主机的 SMTP 客户和接收主机的 SMTP 服务器之间建立的。SMTP 不使用中间的邮件服务器

## ■ 邮件传送

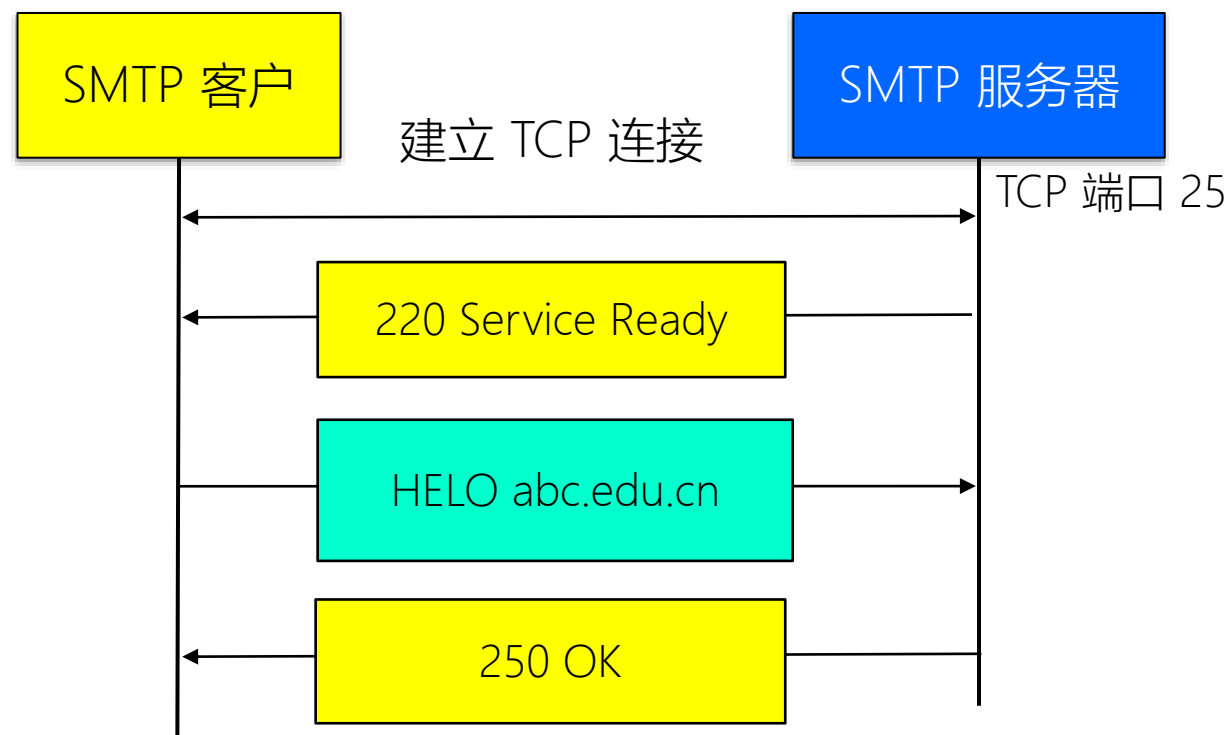
## ■ 连接释放

- 邮件发送完毕后，SMTP 应释放 TCP 连接



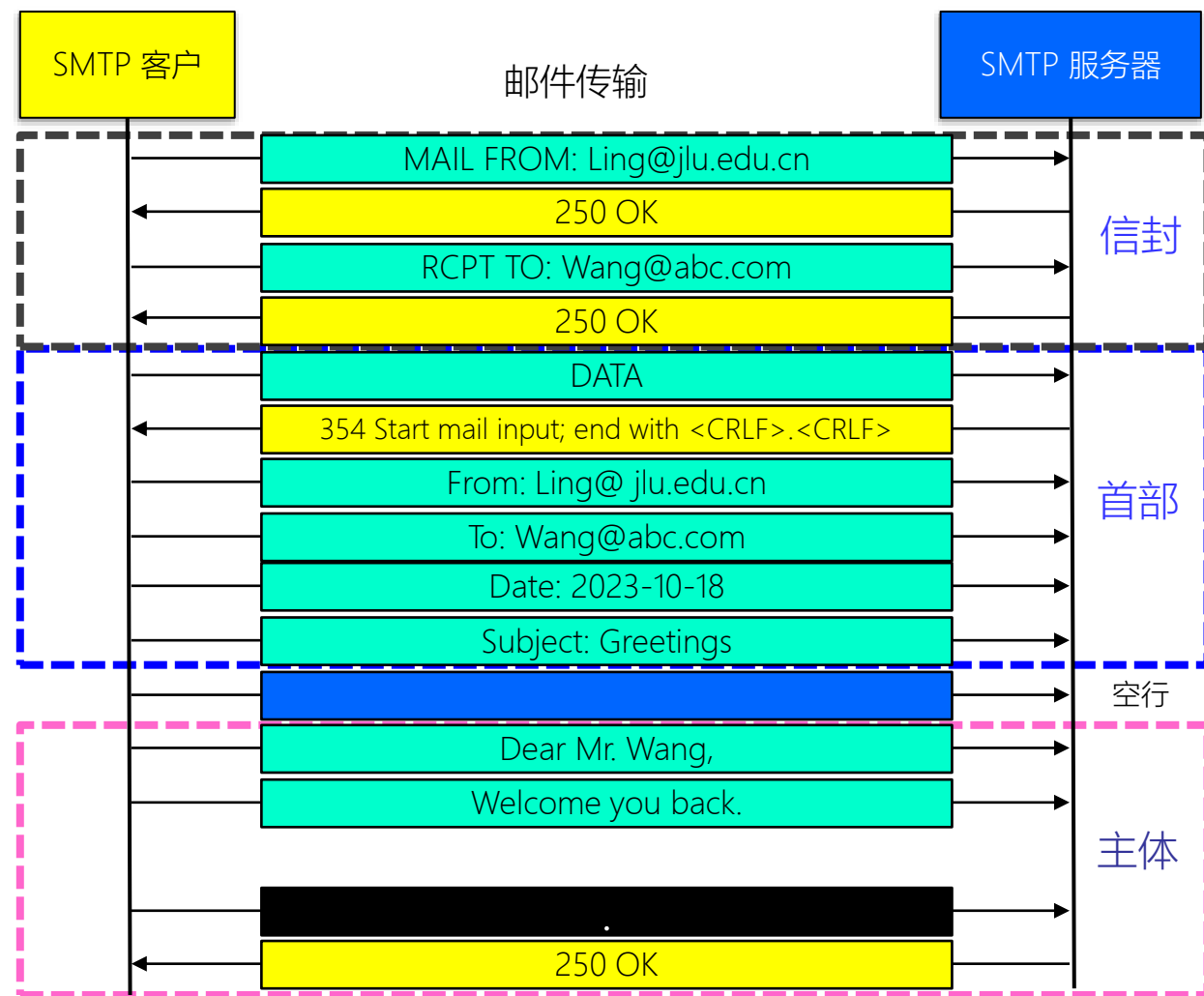
# SMTP 通信的三个阶段——建立阶段

- 使用SMTP的默认端口（25）与目的主机的SMTP服务器建立TCP连接
- SMTP服务器发出“220 SMTP ready”
- SMTP客户向SMTP服务器发送HELO命令，附上发送方的主机名
- SMTP服务器若有能力接收邮件，则回答：“250 OK”，表示已准备好接收



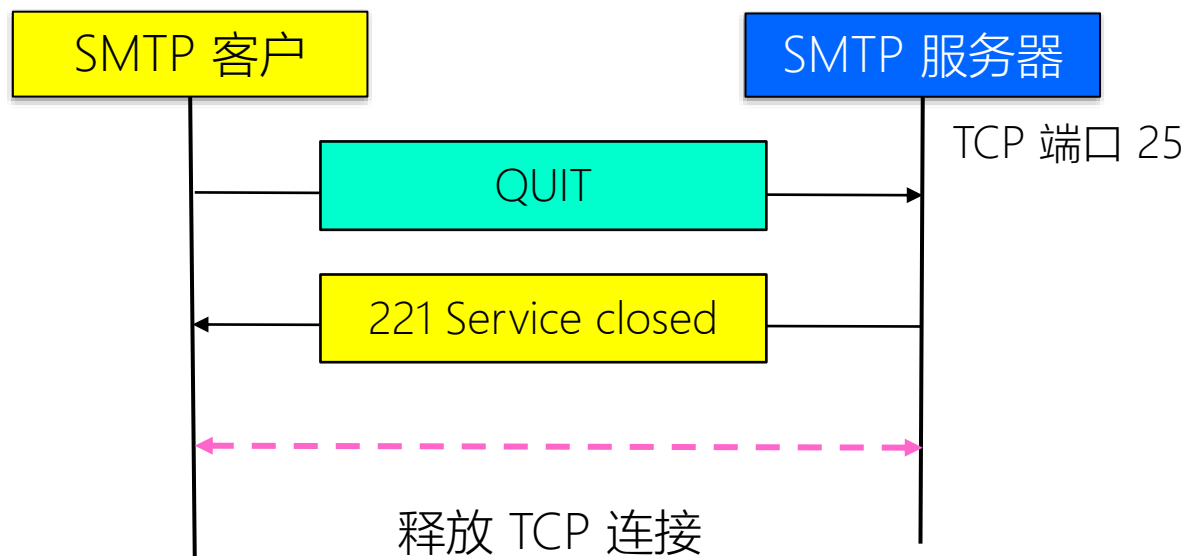
# SMTP 通信的三个阶段——邮件传输

- 邮件的传输首先从MAIL命令开始
- 根据收件人数量，发送一个或多个RCPT命令
- 先确认接收端系统是否已做好接收邮件的准备，然后才能发送邮件
- 每发送一个命令，都应当有相应的信息从SMTP服务器返回
- 接下来是DATA命令，表示要开始传输邮件的内容了



# SMTP 通信的三个阶段——释放连接

- 邮件发送完毕后，SMTP客户应发送QUIT命令；
- SMTP服务器返回的信息是“221 close connection”，表示SMTP同意释放TCP连接





# 接收邮件协议

- POP (Post Office Protocol)
  - POP使用客户服务器模式，接收邮件的计算机运行POP客户程序，其ISP的邮件服务器中运行POP服务程序
  - POP是脱机程序
  - 服务端口：110，安全端口：995
- IMAP (Internet Message Access Protocol)
  - 接收邮件协议
  - 不同于POP，IMAP是联机协议
  - 服务端口：143，安全端口：993

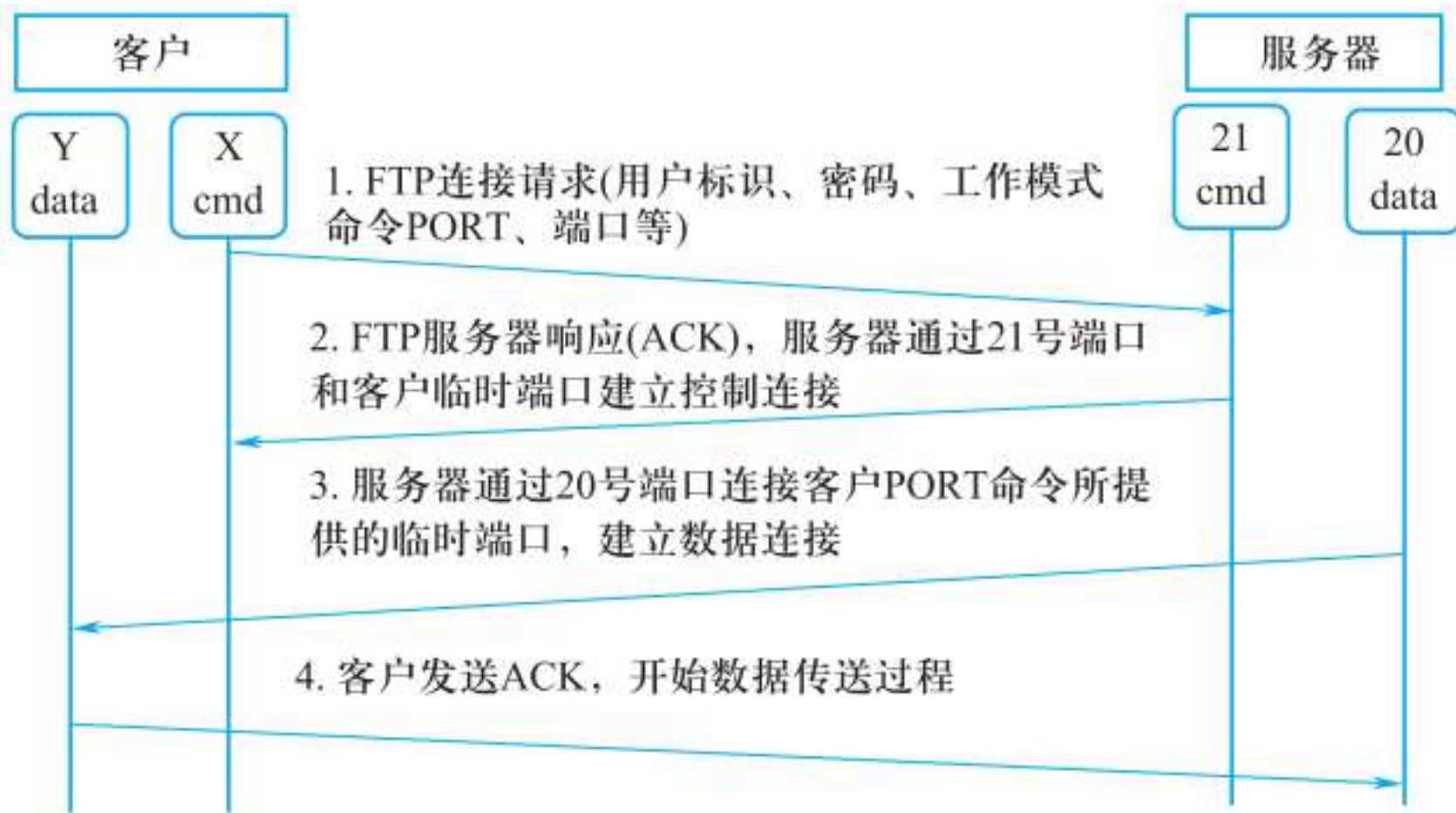
# FTP服务

- FTP(File Transfer Protocol)文件传输协议
- FTP是基于TCP的文件传输协议，用于在两台异构的主机间传输文件，可靠性由TCP保障。
- 主要功能是减少或消除不同操作系统下处理文件的不兼容性。
- 有两种类型：匿名、非匿名
- FTP与网络文件系统不同

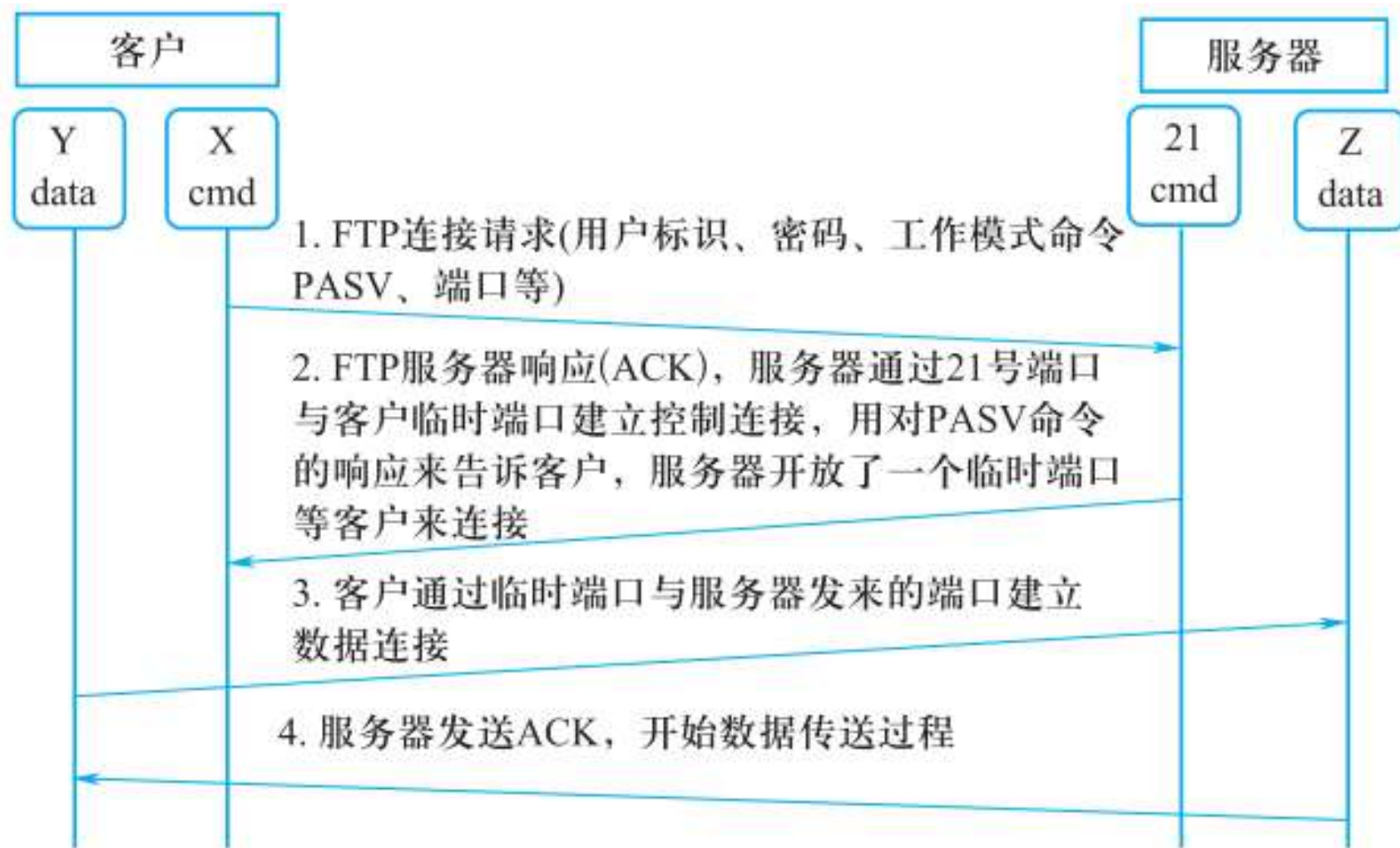
# FTP的工作原理

- FTP是一个客户机/服务器系统。
  - 用户通过一个支持FTP协议的客户机程序，连接到在远程主机上的FTP服务器程序。
  - 用户通过客户机程序向服务器程序发出命令，服务器程序执行用户所发出的命令，并将执行的结果返回到客户机。
- FTP的控制信息是带外传送的。当FTP客户端与服务器建立FTP连接时，将与服务器上的两个端口建立联系
  - 端口21用于控制连接，如用户标识、操作命令
  - 端口20用于数据连接
- 在一次会话过程中，控制始终保持，数据连接随文件的传输会不断地打开与关闭
- 两种工作方式：主动(PORT)模式与被动(PASV)模式

# 主动模式



# 被动模式



# FTP的主要命令

---

- LIST -- 目录名列出子目录或文件
- RETR -- 文件从服务器传送到客户
- USER -- 用户信息
- PASS -- 用户口令
- QUIT -- 向系统注销
- STOR -- 文件从客户传送到服务器

# 普通文件传输协议

- TFTP (Trivial File Transfer Protocol)
- 使用UDP协议，其优点：
  - 简单、代码所占的内存小
  - 因使用UDP协议，可用于分发程序或文件
  - TFTP 需要有自己的差错改正措施
- 只支持文件传输，不支持交互

# WEB服务

- 万维网(World Wide Web)是一个分布式超文本系统。使Internet中不同计算机的文件相互链接。
- WEB服务器是指服务器及运行在服务器上运行的软件。



# WEB涉及的内容

## ■ 两个问题

- 能否不按照物理地址，而是根据内容进行查询
- 能否用统一的格式对声图文等各类资源进行标注

## ■ 两个概念

- 超文本概念-Hypertext: 不是顺序关系，而是链接关系
- 通用资源定位概念-Universal Resource Locator: 对各种资源统一定位

## ■ 两项实现技术

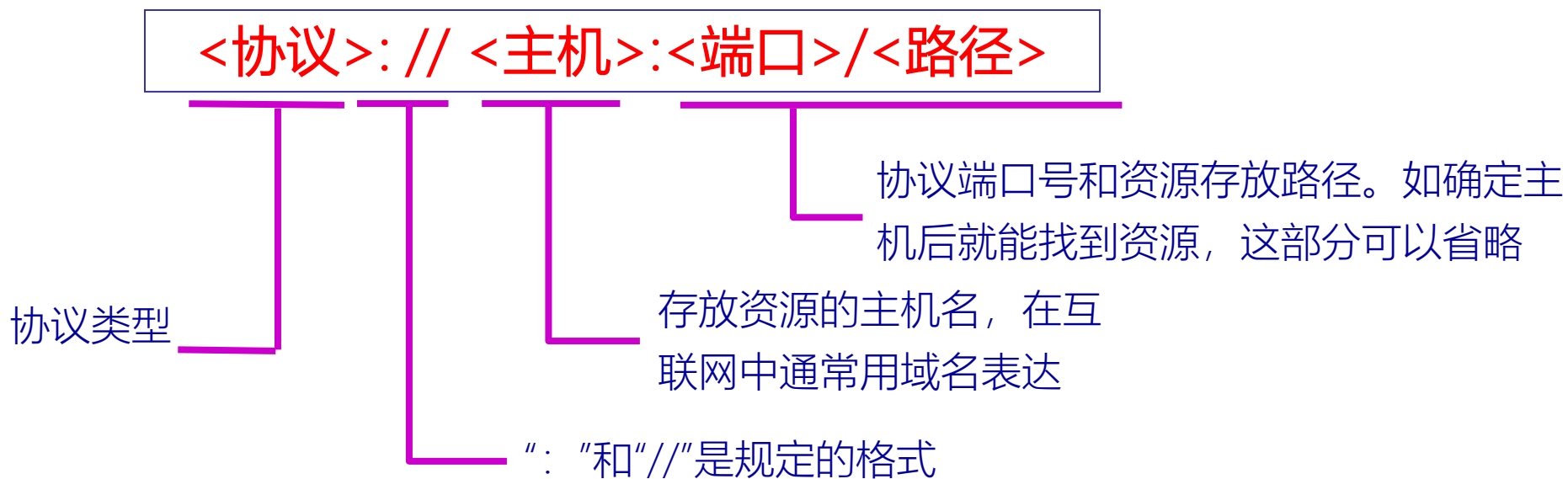
- 超文本传输协议-HTTP
- 超文本标识语言-HTML: 如何链接、是哪种资源（声图文等）

# 客户浏览网页的基本过程

1. 客户机向Web服务器发起TCP连接;
2. 客户机上的浏览器程序根据要访问页面的网址, 发出HTTP请求报文;
3. HTTP报文中包含统一资源定位器 (Uniform Resource Locator, URL), 即通常所说的网址。互联网上的所有资源都有一个唯一的URL。网页超链接中包含URL的信息, 当点击网页上的一个超链接时, HTTP的请求报文会把URL携带在报文当中, 发送给服务器。
4. 服务器根据收到URL找到相应的网页资源, 网页资源使用HTML语言编写, 网页资源作为HTTP报文的响应信息返回给浏览器;
5. 浏览器收到HTML语言的网页资源后, 进行解释, 以一定的格式显示在浏览器当中, 呈现给客户。

# URL的格式

- URL由两大部分组成，以“//”冒号隔开，且对字符大、小写没有要求。  
一般形式



# 超文本

- HTML(Hypertext Markup Language)超文本标记语言，万维网文档发布和浏览的基本文件格式。有如下特点：
  - 独立于平台的格式
  - 文本：允许文档之间漫游
  - 构化设计

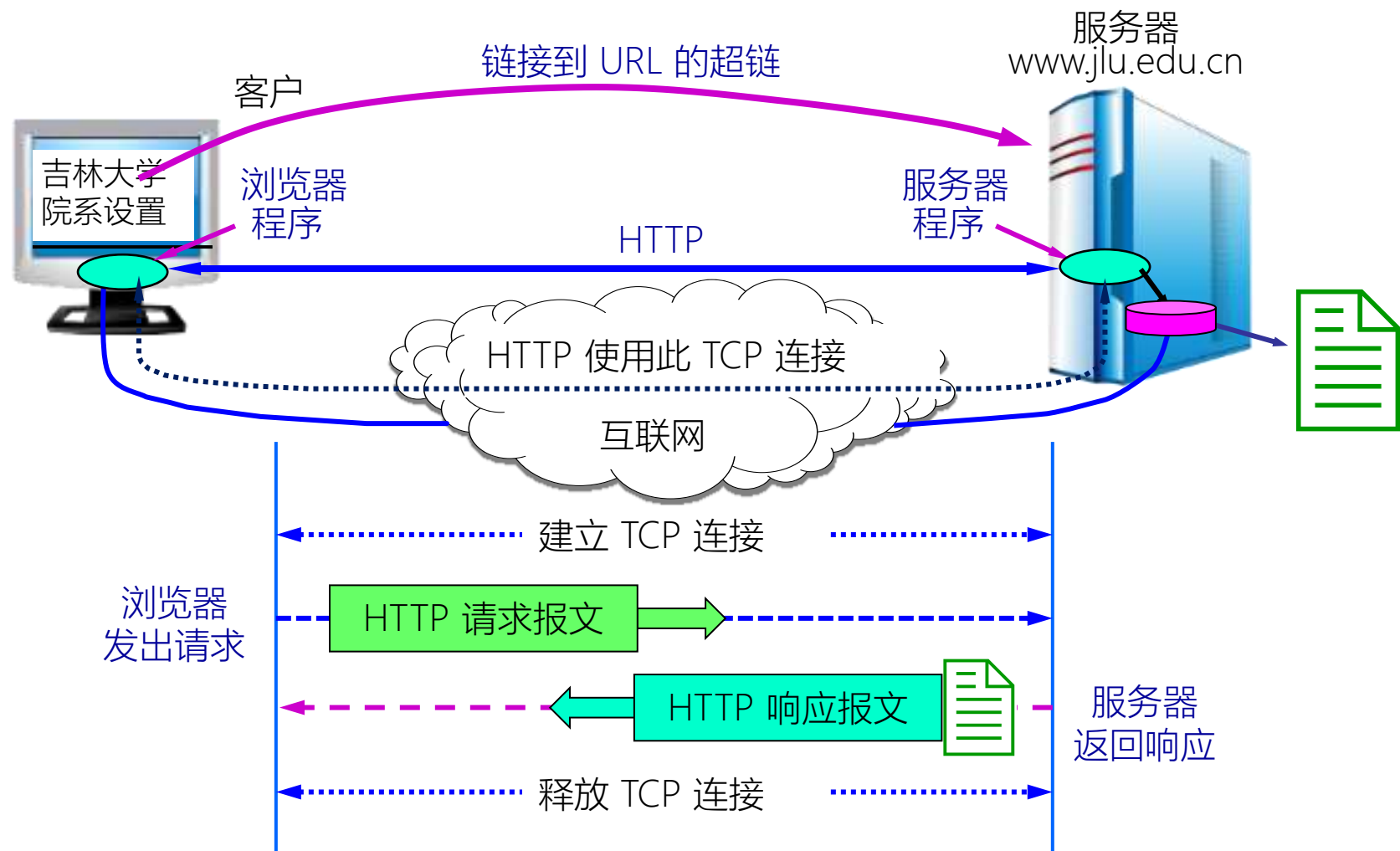
# 超文本传输协议

- HTTP(Hypertext Transfer Protocol)超文本传输协议以普通文本、超文本、音频、视频等格式传输数据。通常在端口80上使用TCP服务
- HTTP是典型的客户/服务器模式。客户是浏览器，服务是WWW服务器
- HTTP协议定义了这些报文的结构和交换的规范
- HTTP协议是无状态协议，不保留客户的状态
- HTTP协议有持续连接和非持续连接

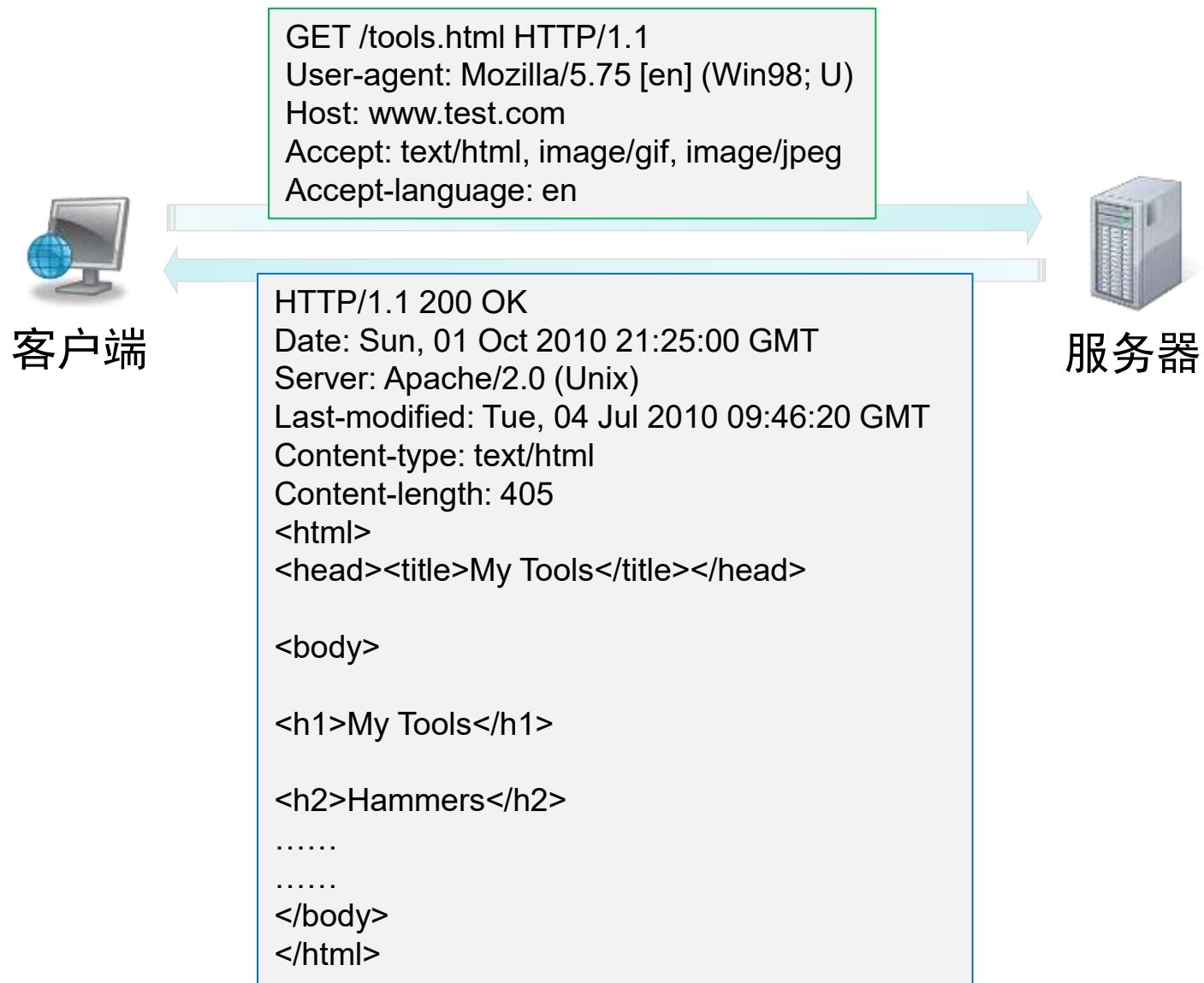
# 4种主要的HTTP请求类型

- GET：请求一个文档
  - 服务器响应：发送状态信息，紧接着发送该文档的一个副本
- HEAD：请求状态信息
  - 服务器响应：发送状态信息，但不发送文档副本
- POST：发送数据给服务器
  - 服务器将该数据添加到指定的项上
- PUT：发送数据给服务器
  - 服务器用该数据完全替代指定项

# HTTP 协议工作过程：协议流程示例



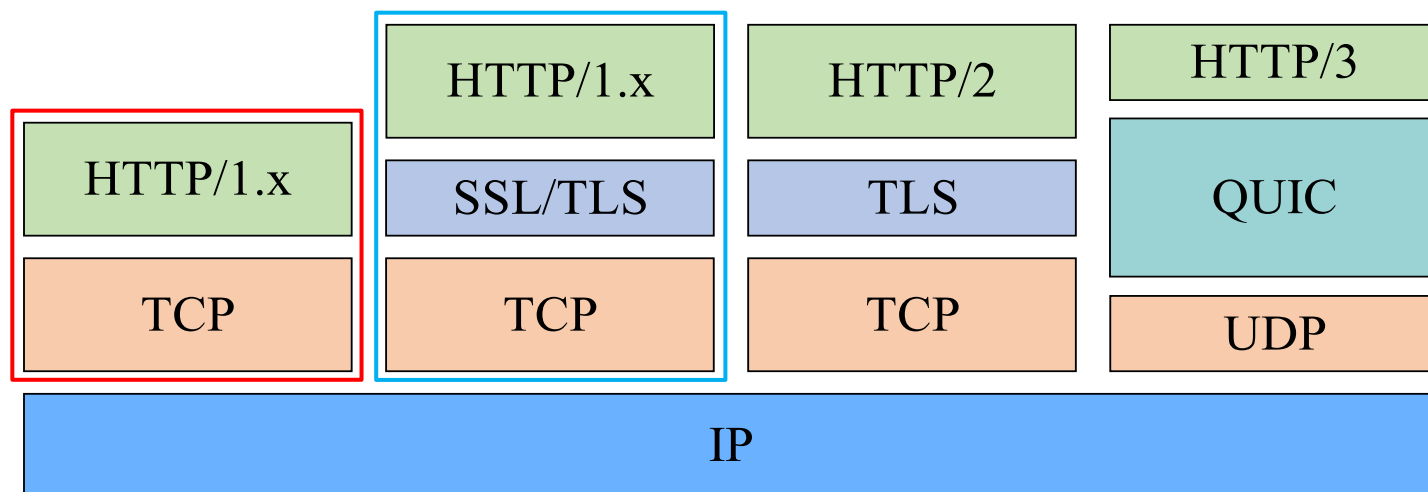
# 简单示例



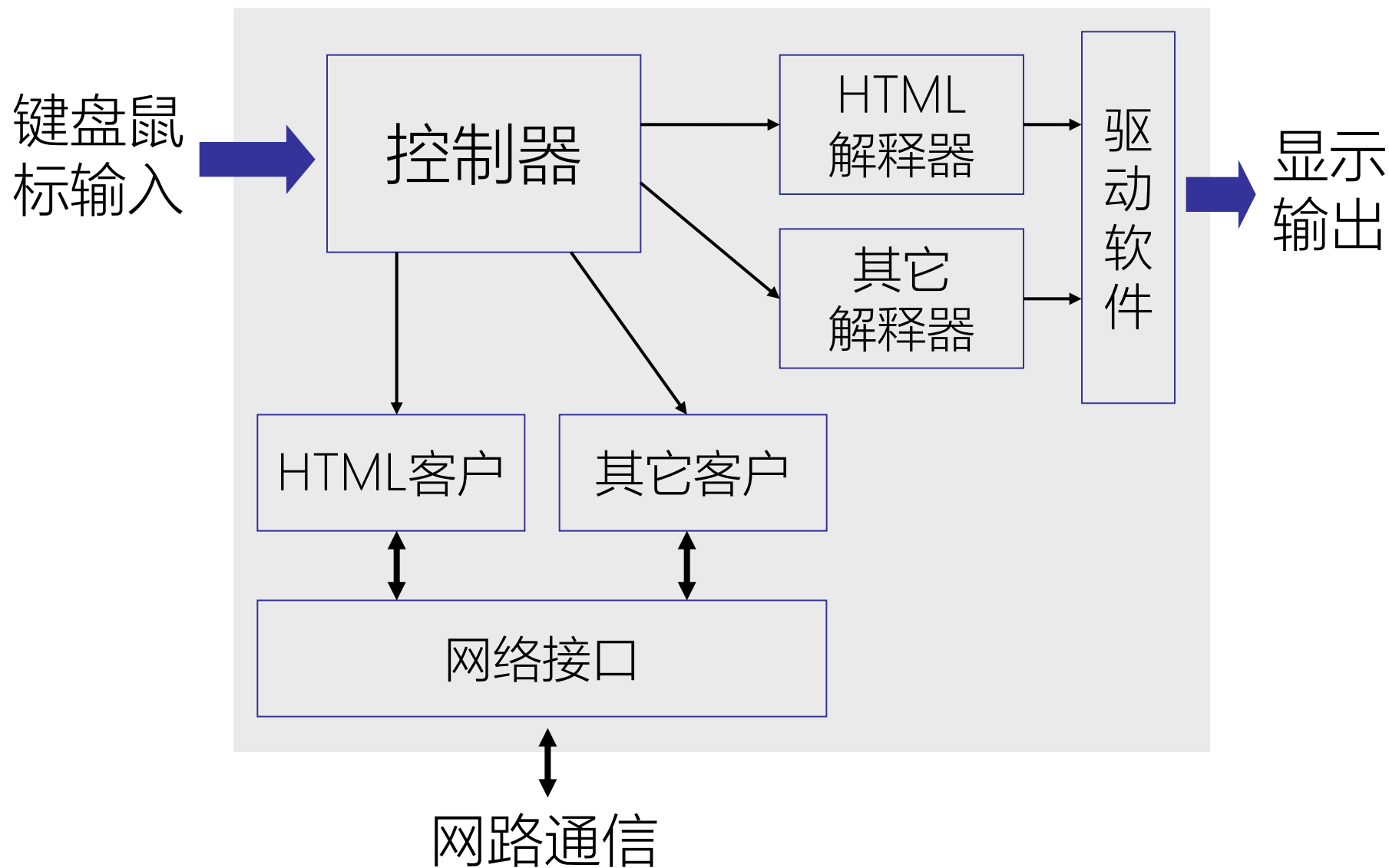


# HTTP 发展现状

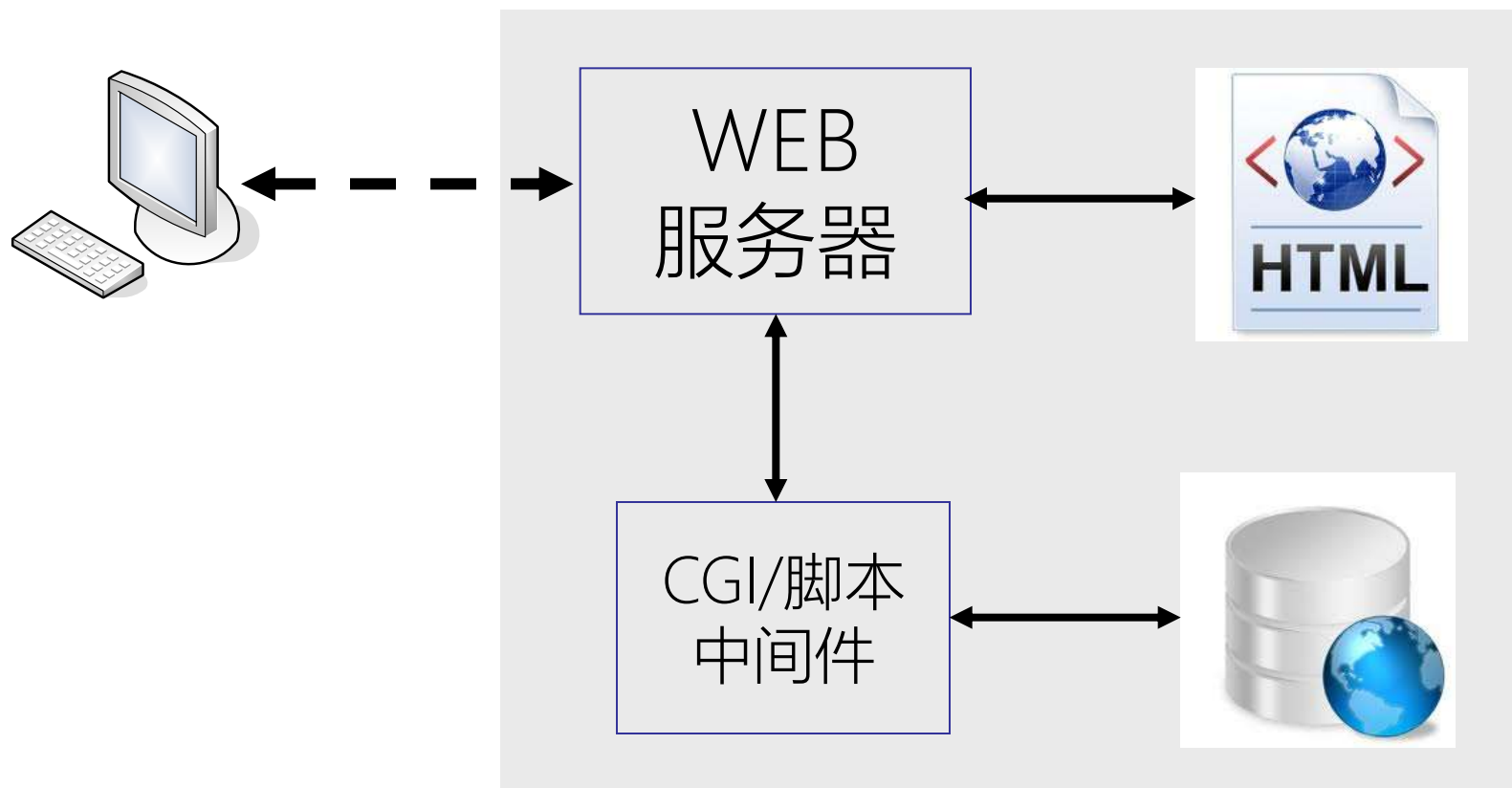
- HTTP/1.0 (1996)
  - 无状态, 非持久连接
- 与HTTP/1.1 (1999)
  - 支持长连接和流水线机制
  - 缓存策略优化、部分资源请求及断点续传
- HTTPS: HTTP+TLS (2008)
  - 增加SSL/TLS (TLS 1.2) 层, 在TCP之上提供安全机制
- HTTP/2.0 (2015、2020)
  - 目标: 提高带宽利用率、降低延迟
  - 增加二进制格式、TCP多路复用、头压缩、服务端推送等功能



# 浏览器结构



# WEB服务器结构



# 小结

- 互联网提供的网络服务都属于应用层，服务模式包括两类
  - 客户/服务器模型
  - P2P模型
- TCP/IP网络架构以IP地址来识别不同的主机，需要DNS将域名翻译成IP地址
  - DNS服务通过层次化管理
  - 根域名服务器 -- 顶级域名服务器 -- 权限域名服务器 -- 本地域名服务器
- 域名的解析请求由主机解析器向本地服务器发起，如果本地服务器查询到该域名是在本网络子域中，则从数据库中找出域名并返回数据信息。
  - 如果该域名非本网络子域域名，则解析方法有两种：重复解析和递归解析

# 小结

- 电子邮件（Email）是互联网上广泛使用的一种应用；
- 电子邮件系统应具有三个主要组成部件：
  - 用户代理（User Agent）
  - 邮件服务器(E-mail Server)
  - 电子邮件传输协议
    - 发送/传递协议（SMTP, MIME）
    - 读取协议（POP、IMAP）

# 小结

- FTP应用程序通过本地的客户端程序与远程FTP服务器进行交互，实现访问远程文件系统；
- 服务端需要开放两个独立的TCP端口：
  - 用于控制连接（Control Connection），缺省端口号是21；
  - 用于数据连接（Data Connection），缺省端口号是20；
- FTP有两种工作模式：
  - 主动模式（PORT模式）
  - 被动模式（PASV模式）
- TCP/IP协议簇中还有一个简单文件传输协议（TFTP）
  - 使用UDP，只支持文件传输，它不支持交互

# 小结

- Web服务是目前最广泛的互联网应用，由大量分布在互联网上、以超文本页面形式存在的内容组成
- Web应用结构包括三个部分
  - Web服务器
  - 浏览器
  - 超文本传输协议HTTP
- 互联网上的所有资源都有一个唯一的URL，有3部分
  - 协议
  - 页面所在主机的域名（或者IP地址）
  - 唯一指示特定页面的路径