

# 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

# Internet的基本通信模式

## ■ Internet支持2种基本通信模式

流模式	报文模式
<ul style="list-style-type: none"><li>1. 面向连接的</li><li>2. 一对一通信</li><li>3. 一个个字节的序列</li><li>4. 任意长度传输</li><li>5. 被大多数应用使用</li><li>6. 构建在TCP之上</li></ul>	<ul style="list-style-type: none"><li>1. 无连接的</li><li>2. 多对多通信</li><li>3. 一个个报文的序列</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>

提供可靠的服务

提供不可靠的服务

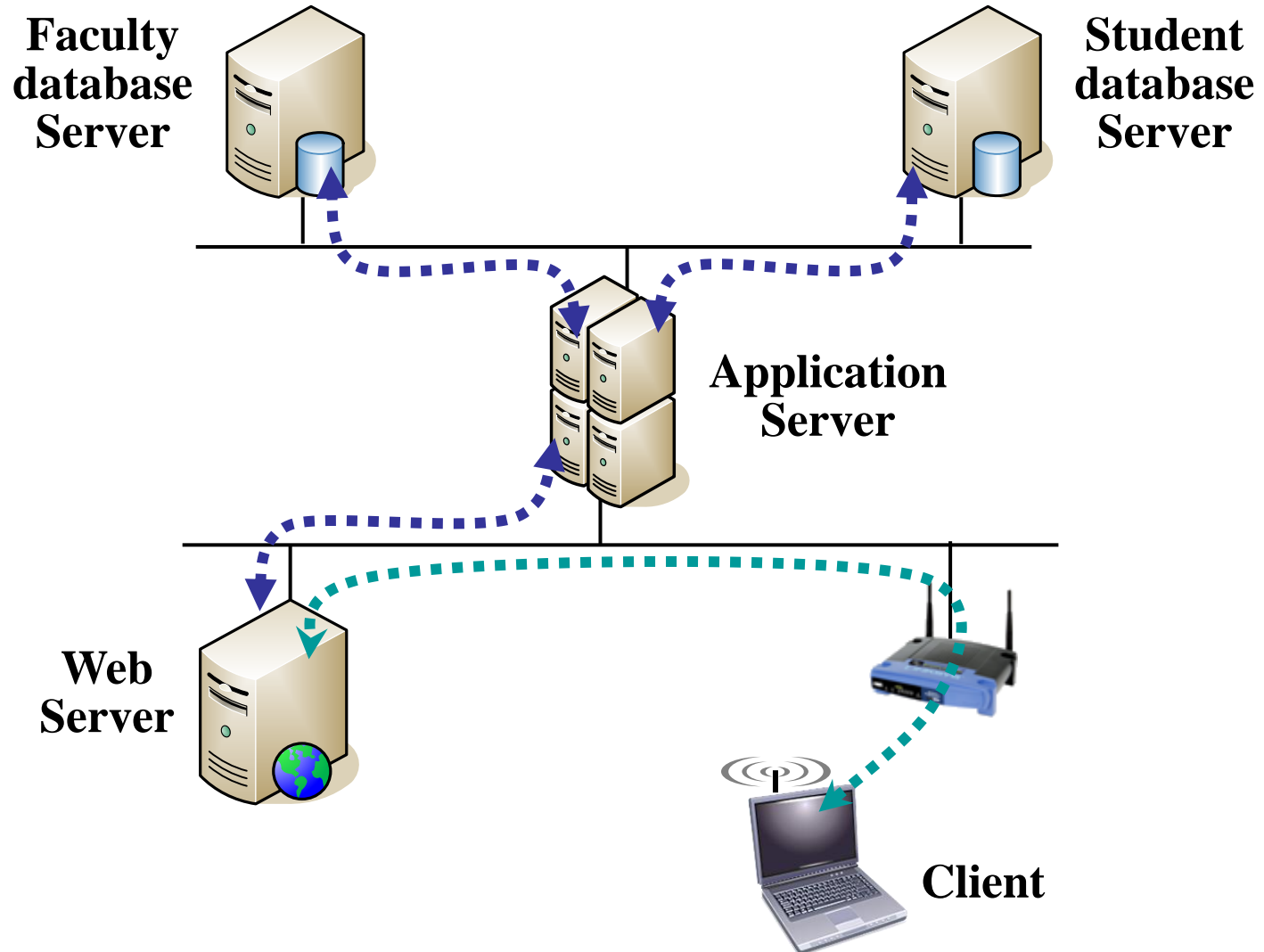
# 应用层协议的特点

- 应用层协议是为了解决某一类应用问题，而问题的解决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)和服务器(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。

# 7.1 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. 并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>

# A Client/Server Model Architecture



## 7.2 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。
- 由于域名具有文字表达的意义，比IP地址更容易记忆。

# 7.2.1 DNS的历史

## ■ 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。

## ■ DNS的出现

- HOSTS 文件机制不具备可缩放性，限制了网络规模。
- 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。

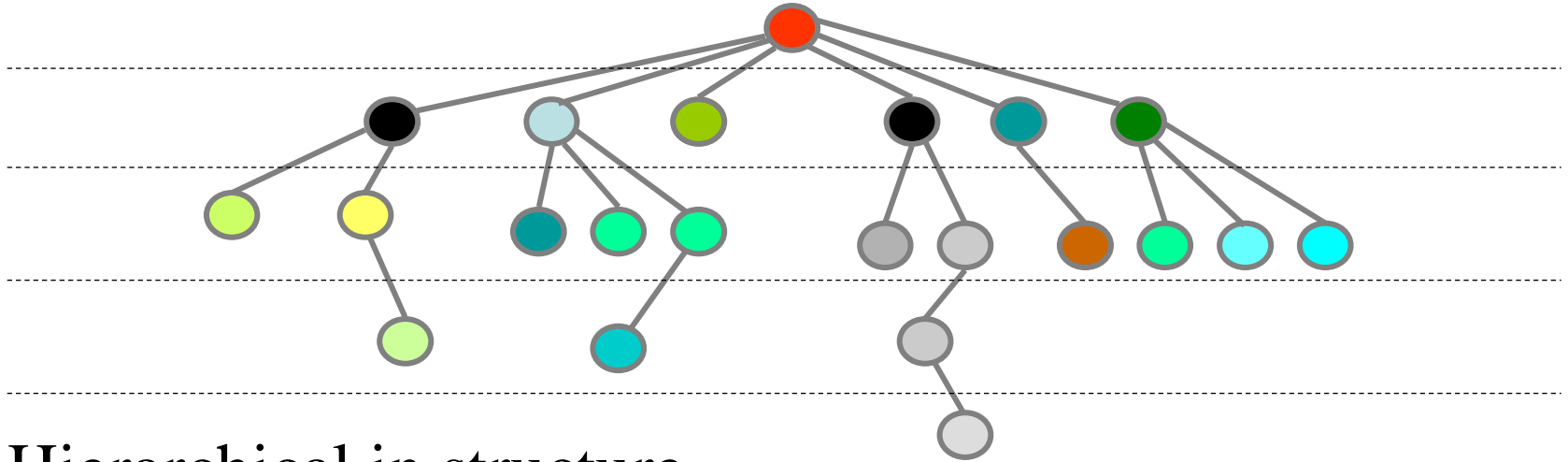
# DNS Histoty

---

- DNS was invented in 1983
- The original specifications appear in RFC882 and RFC883
- RFC 1034 and RFC 1035 updated the DNS specification in Nov. 1987
- In 1984, four Berkeley students wrote the first UNIX implementation
- In 1985, Kevin Dunlap of DEC significantly rewrote the DNS implementation and renamed it BIND (Berkeley Internet Name Domain)
- BIND was ported to the Windows NT platform in the early 1990s



# DNS structure

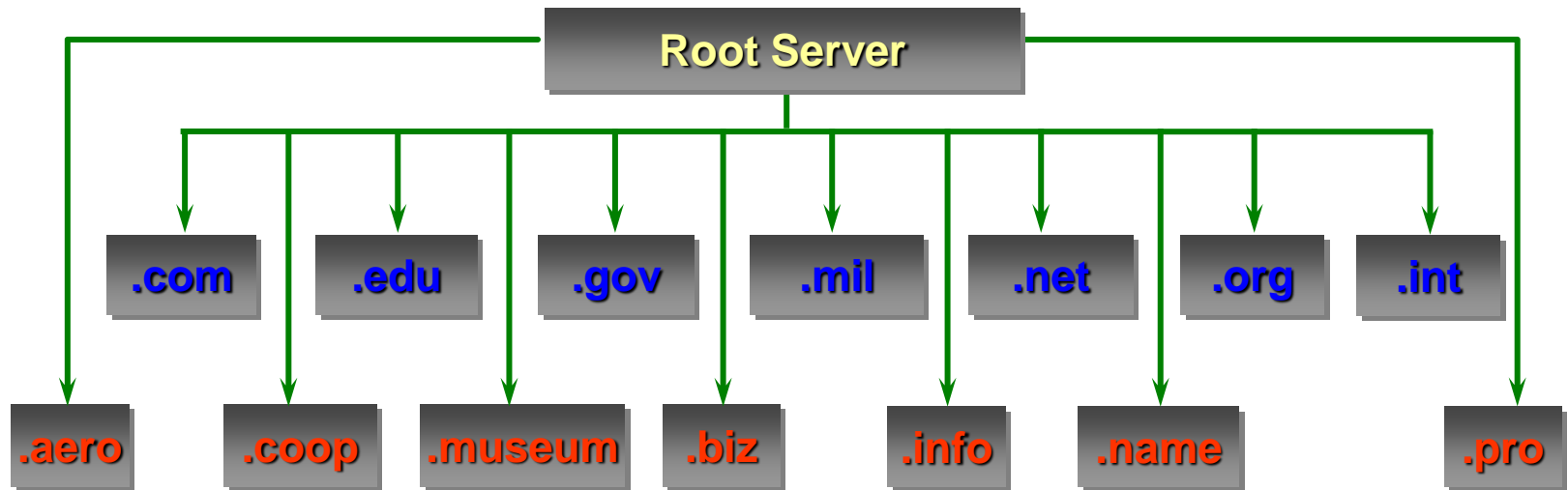


- Hierarchical in structure.
- Each level provides further definition.
- Each branch is called a lable (63 characters in length).
- Internet Registry provides uniqueness in names.
- A single domain is assigned and may be further defined by the local site.

# Top-level domain(TLD)

- There are several types of TLDs
  - country-code TLDs (ccTLDs)
  - generic TLDs (gTLDs)
  - .arpa (special TLD): used for technical infrastructure purposes
- gTLDs
  - 7 gTLDs in the 1985
    - .com , .net, .org, .edu, .gov, .mil and .int
  - 7 new gTLDs in Nov. 2000
    - .aero, .coop, and .museum, .biz, .info, .name, and .pro
  - 6 new gTLDs in 2003
    - .asia, .cat, .jobs, .mobi, .tel and .travel

# DNS structure



- com: Commercial entry
- edu: Education body
- gov: Government body
- mil: Military
- net: network providers
- org: any other organization not previously listed
- Any country using the ISO standard 3166 for names of countries as stated in RFC 1591. E.g.
  - cn → China
  - jp → Japan

# DNS components

- DNS does much more than the name-to-address translation. It also allows for:
  - Domain Name Space and resource records
    - This is the database of grouped names and addresses that are strictly formatted using a tree-structured name space and data associated with the names.
    - The domain system consists of separate sets of local information called zones.
  - Name servers
    - These are workstations that contain a database of information about host in zones.
  - Resolvers
    - These are program that generally reside on user's workstations and send requests over the network to servers on behalf of the users.

## 7.2.2 域名系统(DNS)的工作原理

- 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。
- 域名系统(DNS)采用客户/服务器工作模式。
- 地址到名称的映射和反向查询
  - 从IP地址映射到主机名称称为反向解析

# 域名系统(DNS)的工作原理

## ■ 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2层域名小于12个字符且独立。
- 第3层域名通常表示组织部门或分支。

# 域名系统(DNS)的工作原理

- 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
- 域名服务器
  - 存储有关域名空间信息的程序称为名称域名服务器。它是一个在后台的守候程序，监听来自客户机的请求。

# 域名系统(DNS)的工作原理

## ■ 解释过程:

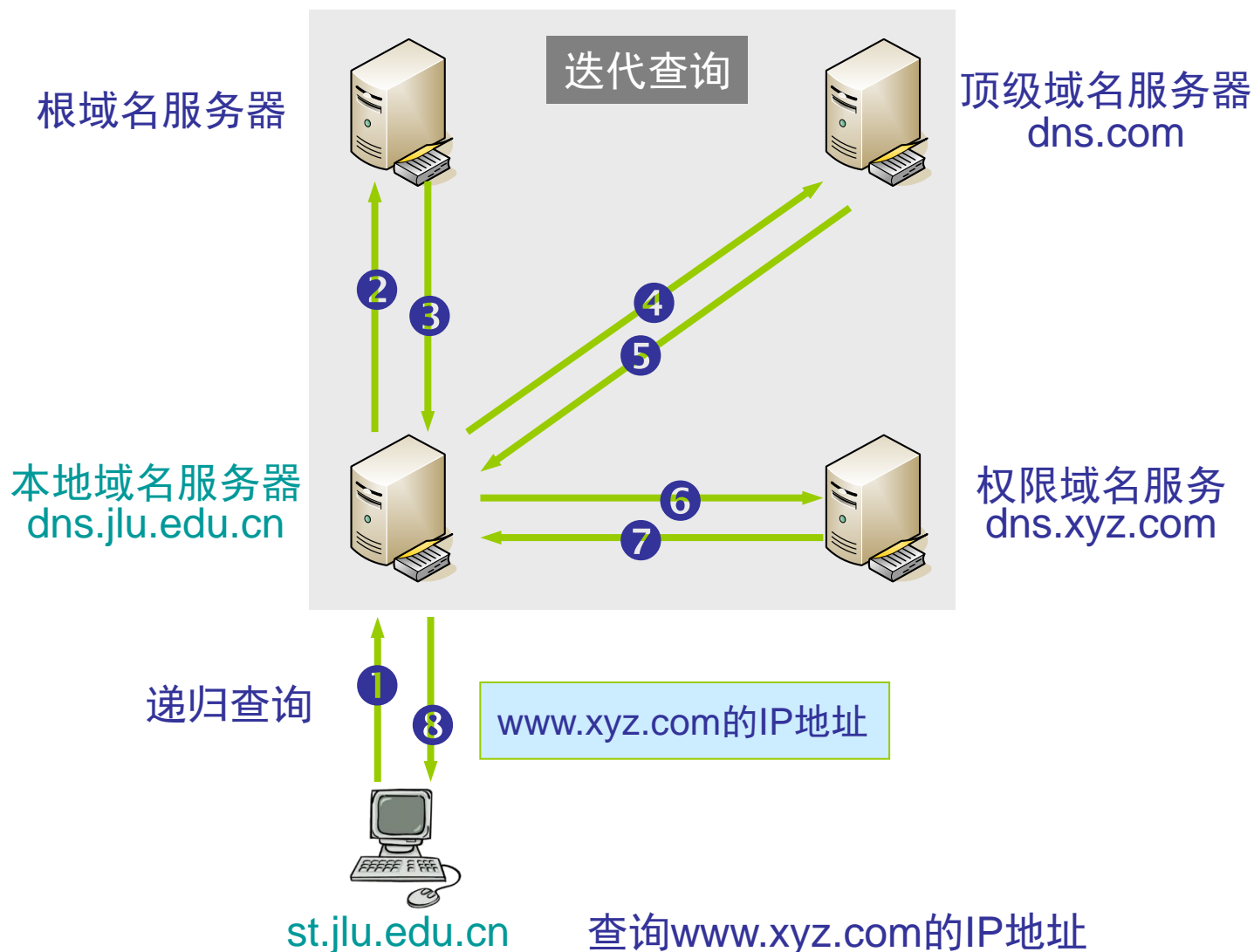
- 域名解析器首先查询本地主机的缓冲区，查看主机是否以前解析过主机名。
- 如果主机缓冲区中没有与其IP地址的映射关系，解析器将向本地域名服务器发出请求。
- 本地域名服务器首先检查域名与其IP地址的映射关系是否存储在它的数据库中，如果是，本地服务器将该映射关系传送给请求者。

## ■ 解释方法有两种:

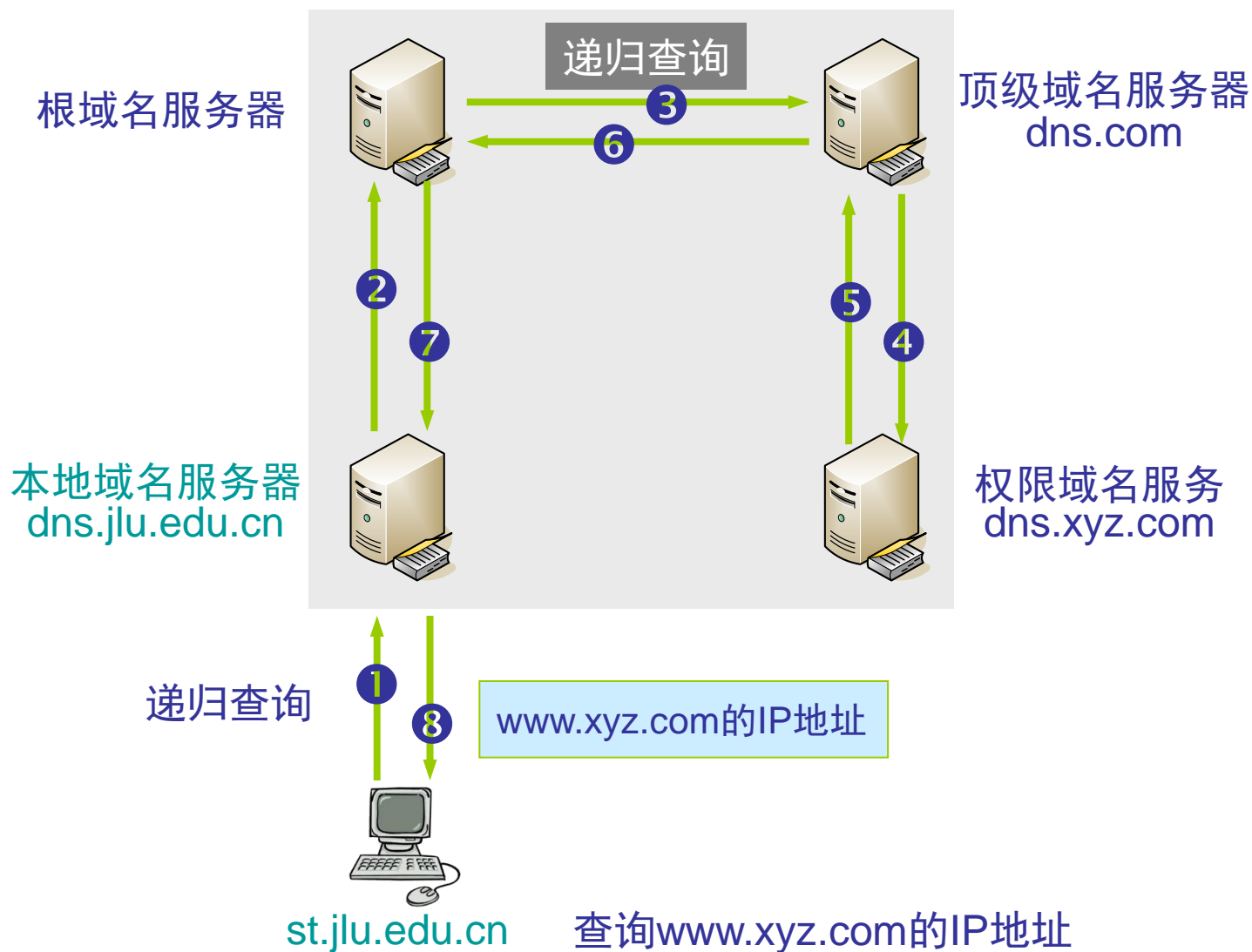
- 重复解释
- 递归解释



# 迭代查询(重复解释)



# 递归查询



# 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

## 7.2.3 域名系统(DNS)报文格式

标识	标志	↑ 头部 12字节 ↓
询问记录数	应答记录数	
授权记录数	附加记录数	
查询		
应答（数目可变）		
授权（数目可变）		
附加（数目可变）		

标识：报文ID，用于匹配报文

标志：报文类型（查询/响应）  
          响应类型（授权/非授权）  
          解析类型（迭代/递归）

## 7.2.4 DNS与ARP的比较

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

# ICANN

- ICANN is the Internet Corporation for Assigned Names and Numbers.
- ICANN is a non-profit corporation that was created on September 18, 1998 in order to oversee a number of Internet-related tasks



Founded	September 18, 1998
Headquarters	Marina Del Rey, CA
Focus	Manage Internet Protocol numbers and Domain Name System root
Website	<a href="http://www.icann.org">www.icann.org</a>



ICANN headquarters at USC ISI

# ICANN's tasks

---

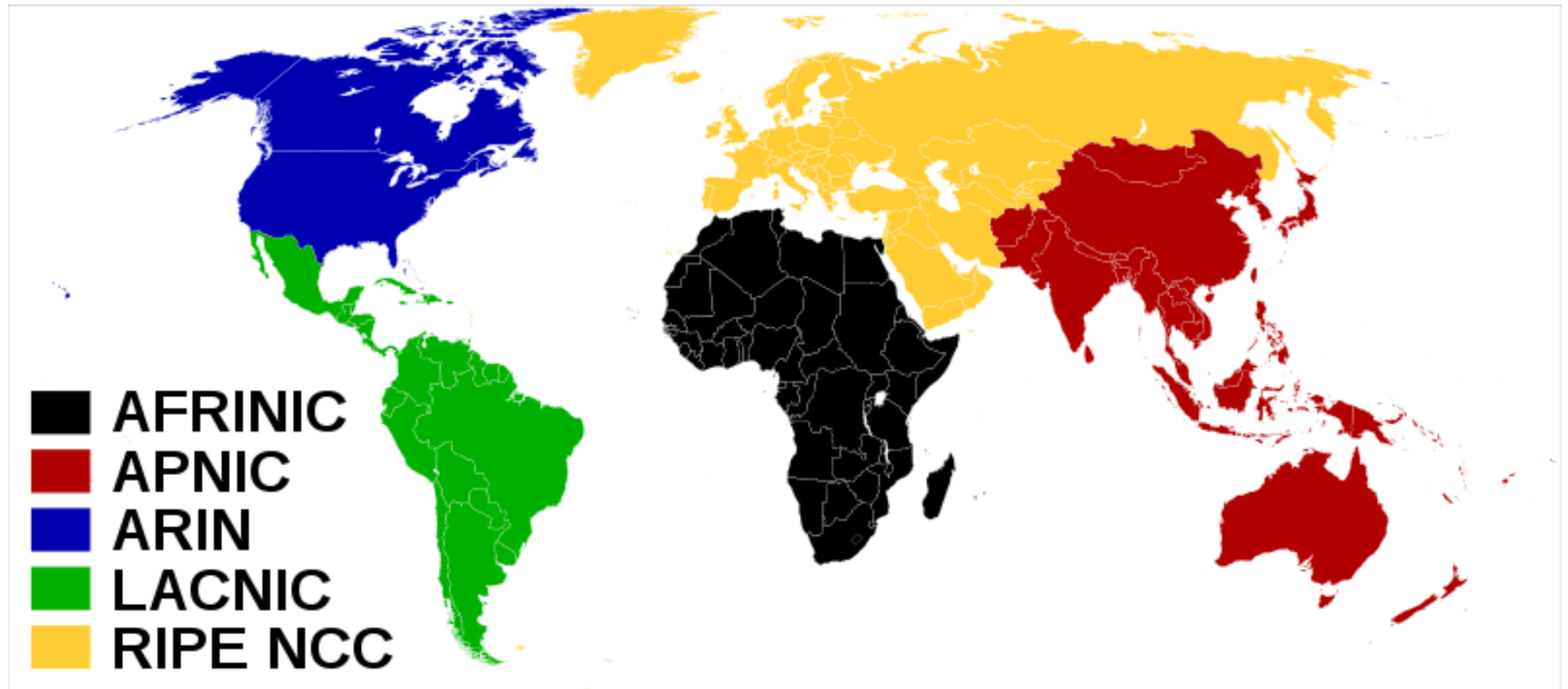
- Internet Protocol (IP) address space allocation,
- Protocol identifier assignment,
- Generic (gTLD) and country code (ccTLD) top-level domain name system management
- Root server system management functions
- More generically, ICANN is responsible for managing the assignment of domain names and IP addresses.

# Regional Internet Registry (RIR)

- A **regional Internet registry (RIR)** is an organization overseeing the allocation and registration of Internet number resources within a particular region of the world.
  - Resources include IP addresses (both IPv4 and IPv6) and autonomous system numbers (for use in BGP routing).
- There are currently five RIRs in operation:
  - American Registry for Internet Numbers (**ARIN**) for North America and parts of the Caribbean
  - RIPE Network Coordination Centre (**RIPE NCC**) for Europe, the Middle East and Central Asia
  - Asia-Pacific Network Information Centre (**APNIC**) for Asia and the Pacific region
  - Latin American and Caribbean Internet Addresses Registry (**LACNIC**) for Latin America and parts of the Caribbean region
  - African Network Information Centre (**AfriNIC**) for Africa



# 5 RIRs

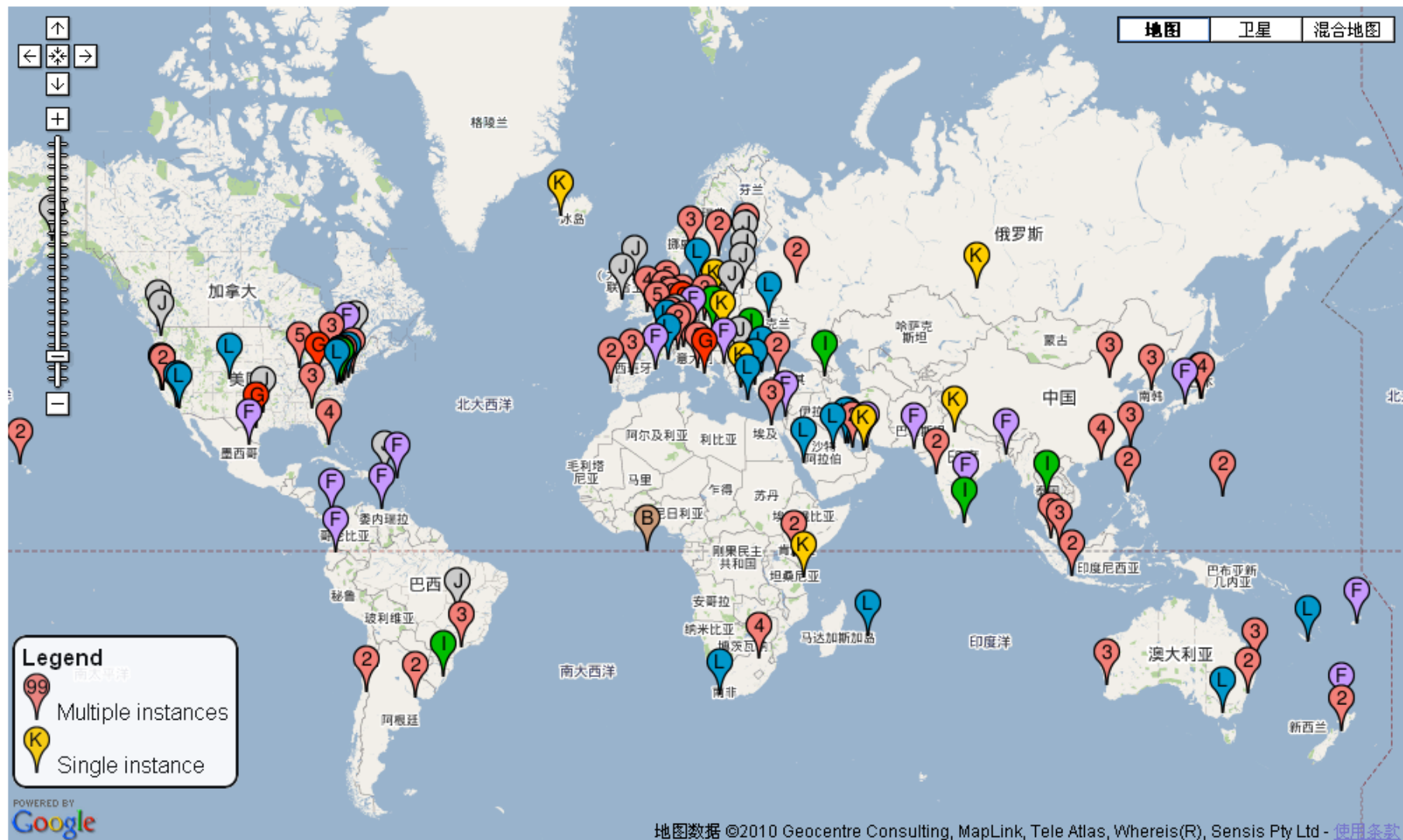


# Root Server List

<http://www.root-servers.org/>

Server	Operator	IPv4 Addresses	IPv6 Addresses
A	VeriSign, Inc.	198.41.0.4	2001:503:BA3E::2:30
B	Information Sciences Institute	192.228.79.201	2001:478:65::53
C	Cogent Communications	192.33.4.12	
D	University of Maryland	128.8.10.90	
E	NASA Ames Research Center	192.203.230.10	
F	Internet Systems Consortium, Inc.	192.5.5.241	2001:500:2f::f
G	U.S. DOD Network Information Center	192.112.36.4	
H	U.S. Army Research Lab	128.63.2.53	2001:500:1::803f:235
I	Autonomica	192.36.148.17	2001:7fe::53
J	VeriSign, Inc.	192.58.128.30	2001:503:C27::2:30
K	RIPE NCC	193.0.14.129	2001:7fd::1
L	ICANN	199.7.83.42	2001:500:3::42
M	WIDE Project	202.12.27.33	2001:dc3::35

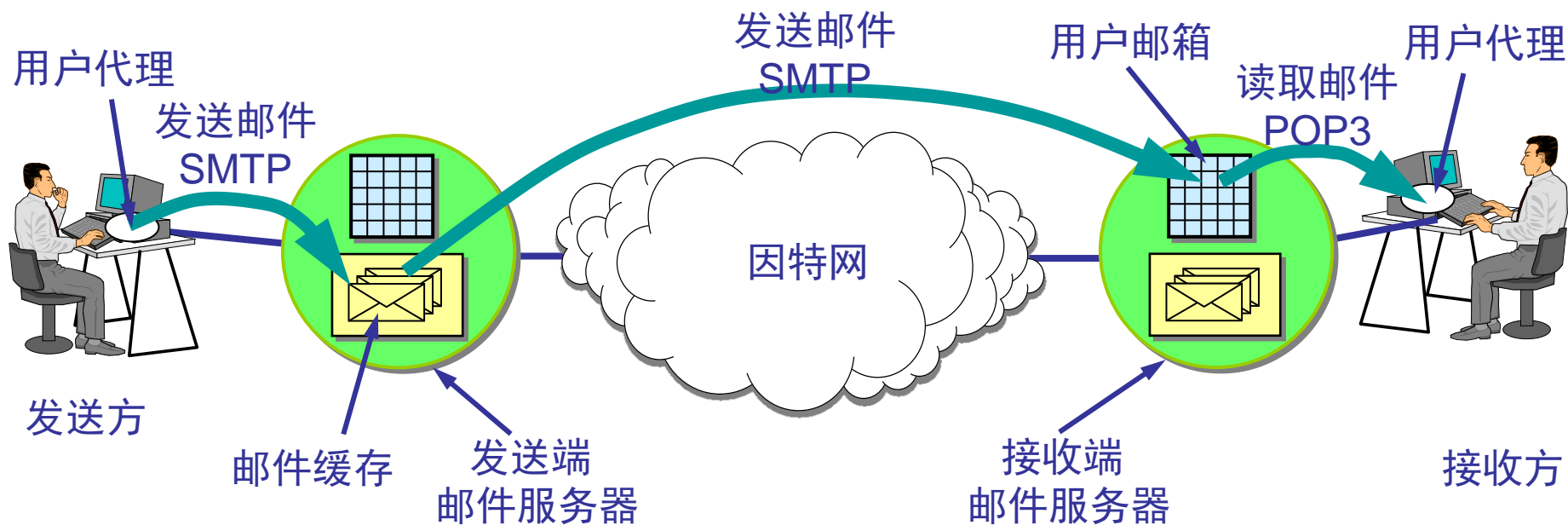
# Root Server Location (total: 243)



## 7.3 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。

# 7.3.1 电子邮件的主要组成



## 7.3.2 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工作流程

telnet mailserver 25

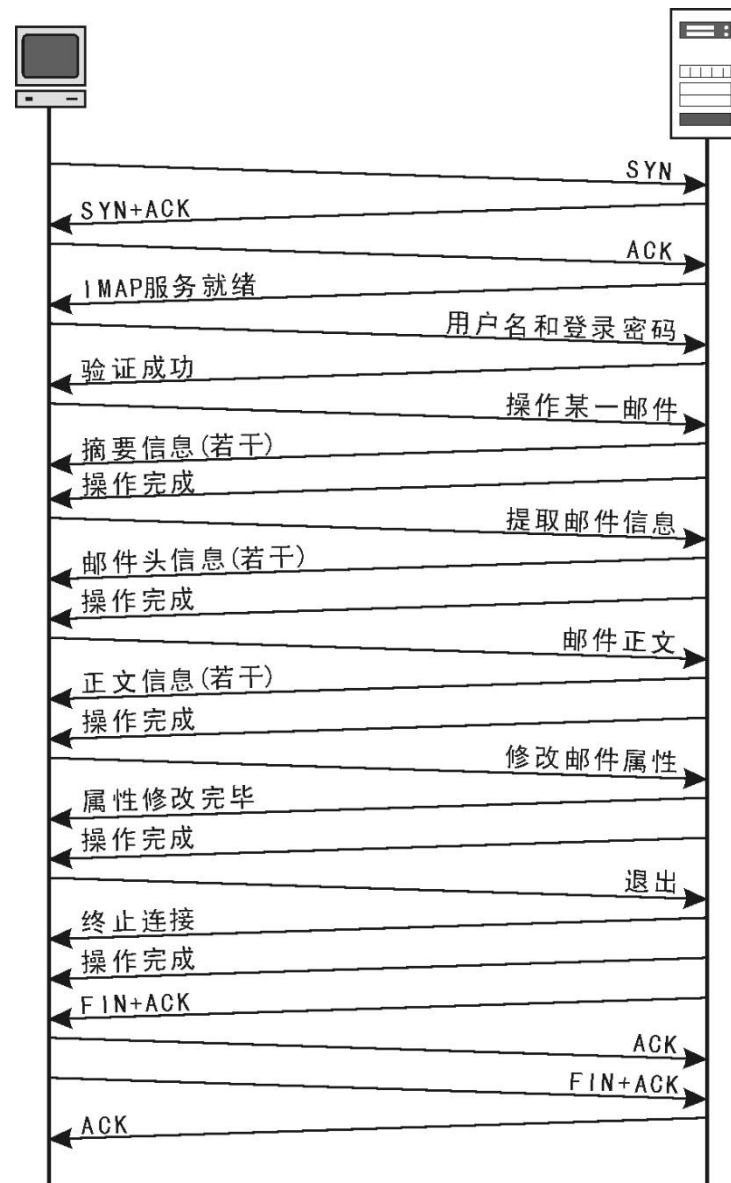
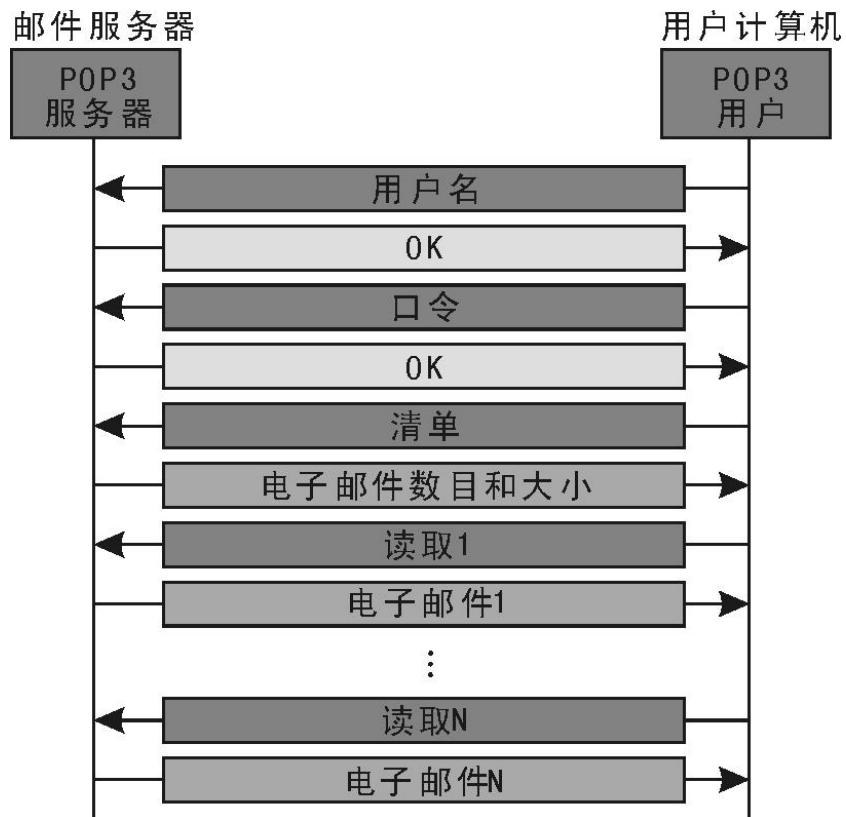
```
220 ESMTP
helo
250 HELO:
auth login
334 VXNlcm5hbWU6
qwertr= <-- user name (base64)
334 UGFzc3dvcmQ6
WERSDF34fds== <-- password (base64)
235 go ahead (eyou mta) (eyou.net _50)
MAIL FROM: zhang@jlu.edu.cn
250 OK (eyou mta)
RCPT TO: zhang@jlu.edu.cn
250 OK (eyou mta) (eyou.net _15728640)
DATA
354 go ahead (eyou mta)
Subject: Test using telnet
(blank line)
This is just a test to see if I can SMTP auth from my PC
.
250 OK:has queued (eyou mta)
quit
```

## 7.3.3 接收邮件协议

- POP(Post Office Protocol)接收邮件协议
  - 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。
- IMAP：也是接收邮件协议，不同于POP，IMAP是联机协议。



# POP3/IMAP工作流程



## 7.4 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

## 7.4.1 文件传输协议(FTP)的工作原理

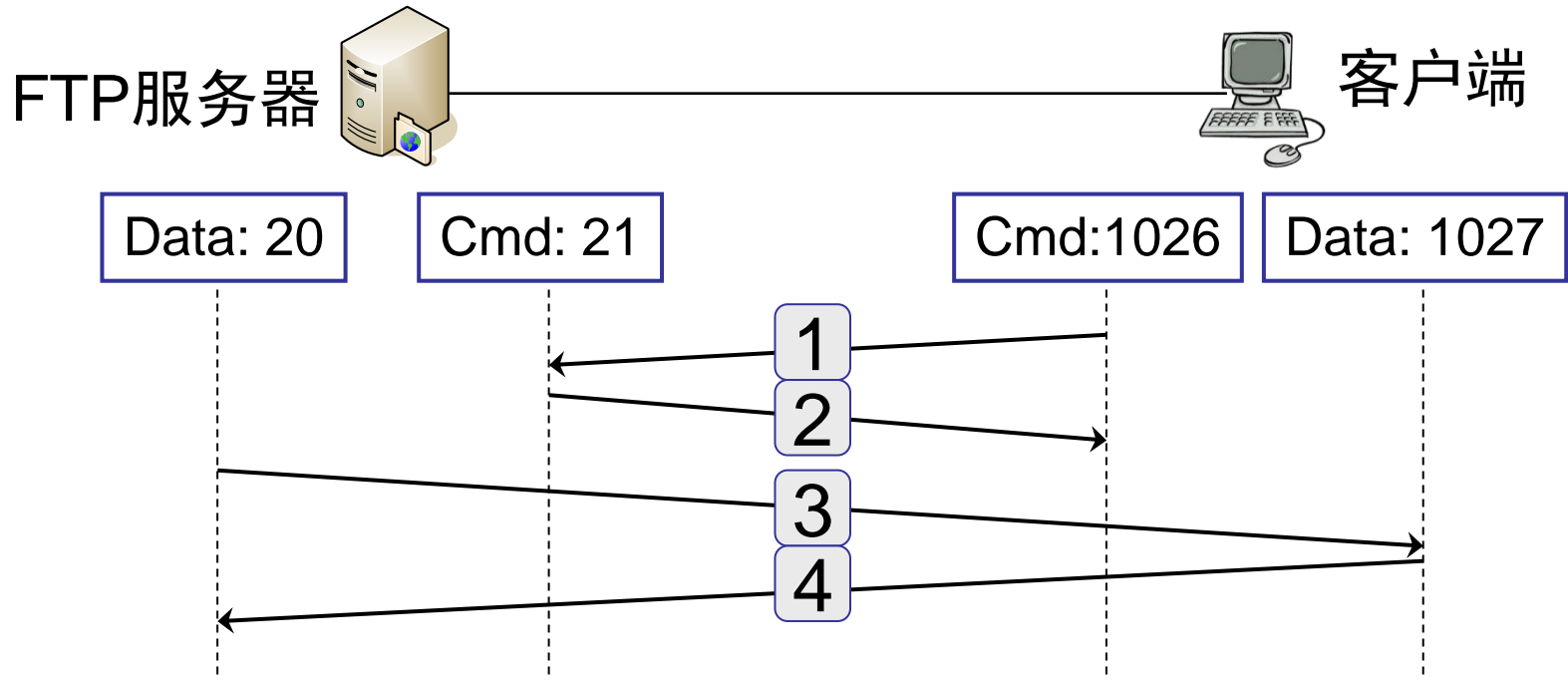
- FTP是一个客户机/服务器系统。
  - 用户通过一个支持FTP协议的客户机程序，连接到在远程主机上的FTP服务器程序。
  - 用户通过客户机程序向服务器程序发出命令，服务器程序执行用户所发出的命令，并将执行的结果返回到客户机。
- 当FTP客户端与服务器建立FTP连接时，将与服务器上的两个端口建立联系
  - 端口21用于控制连接，如用户标识、操作命令
  - 端口20用于数据连接
- 在一次会话过程中，控制始终保持，数据连接随文件的传输会不断地打开与关闭
- FTP的控制信息是带外传送的

# 文件传输协议(FTP)工作方式

## ■ FTP的两种工作方式:

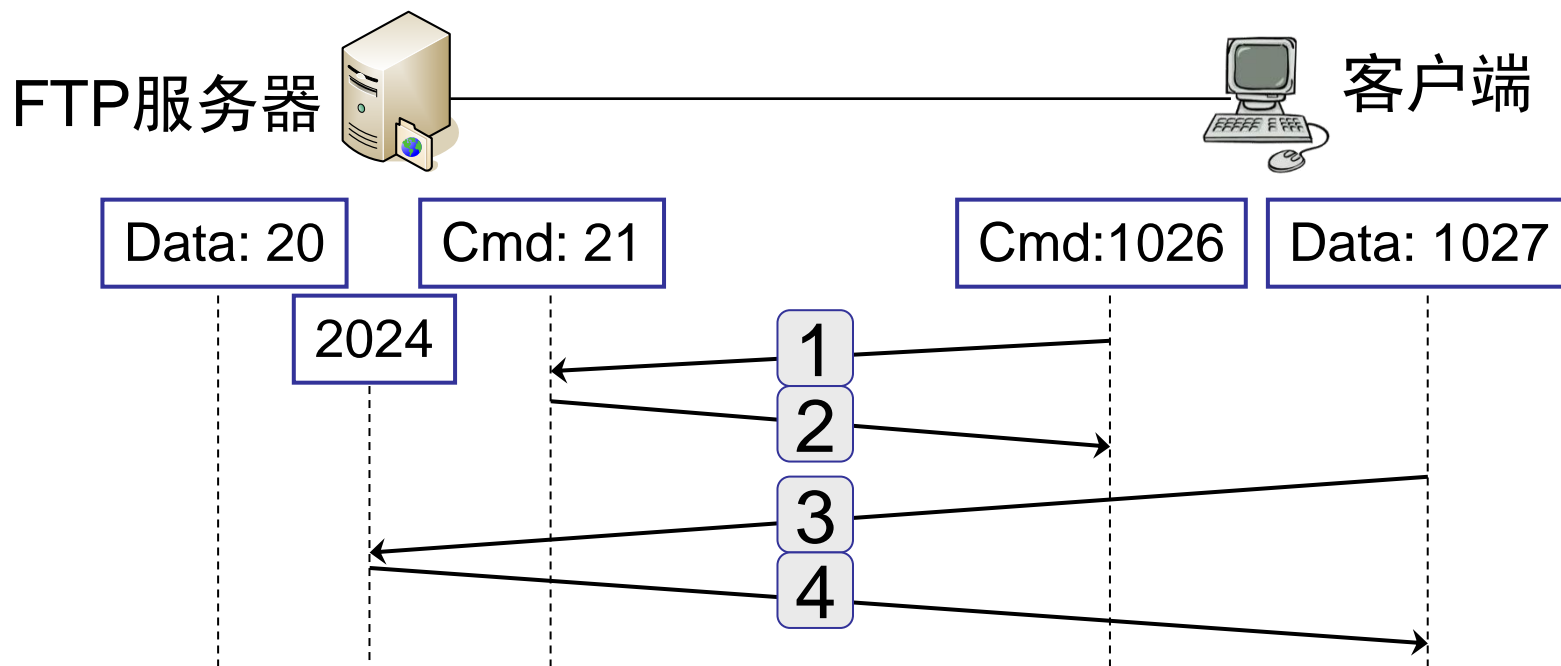
- 主动(PORT)模式
- 被动(PASV)模式

# 主动模式



1. Server's port 21 from anywhere (Client initiates connection)
2. Server's port 21 to ports > 1023 (Server responds to client's control port)
3. Server's port 20 to ports > 1023 (Server initiates data connection to client's data port)
4. Server's port 20 from ports > 1023 (Client sends ACKs to server's data port)

# 被动模式



1. Server's port 21 from anywhere (Client initiates connection)
2. Server's port 21 to ports > 1023 (Server responds to client's control port)
3. Server's ports > 1023 from anywhere (Client initiates data connection to random port specified by server)
4. Server's ports > 1023 to remote ports > 1023 (Server sends ACKs (and data) to client's data port)

## 7.4.2 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

## 7.4.3 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
- 只支持文件传输，不支持交互



## 7.5 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）

## 7.5.1 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
  - **结构化设计**
- **统一资源定位器(URL)**：标识网络上资源位置而设计的一种编址方式。一般由3部分组成  
**传输协议://主机地址[:port]/路径和文件名**

## 7.5.2 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

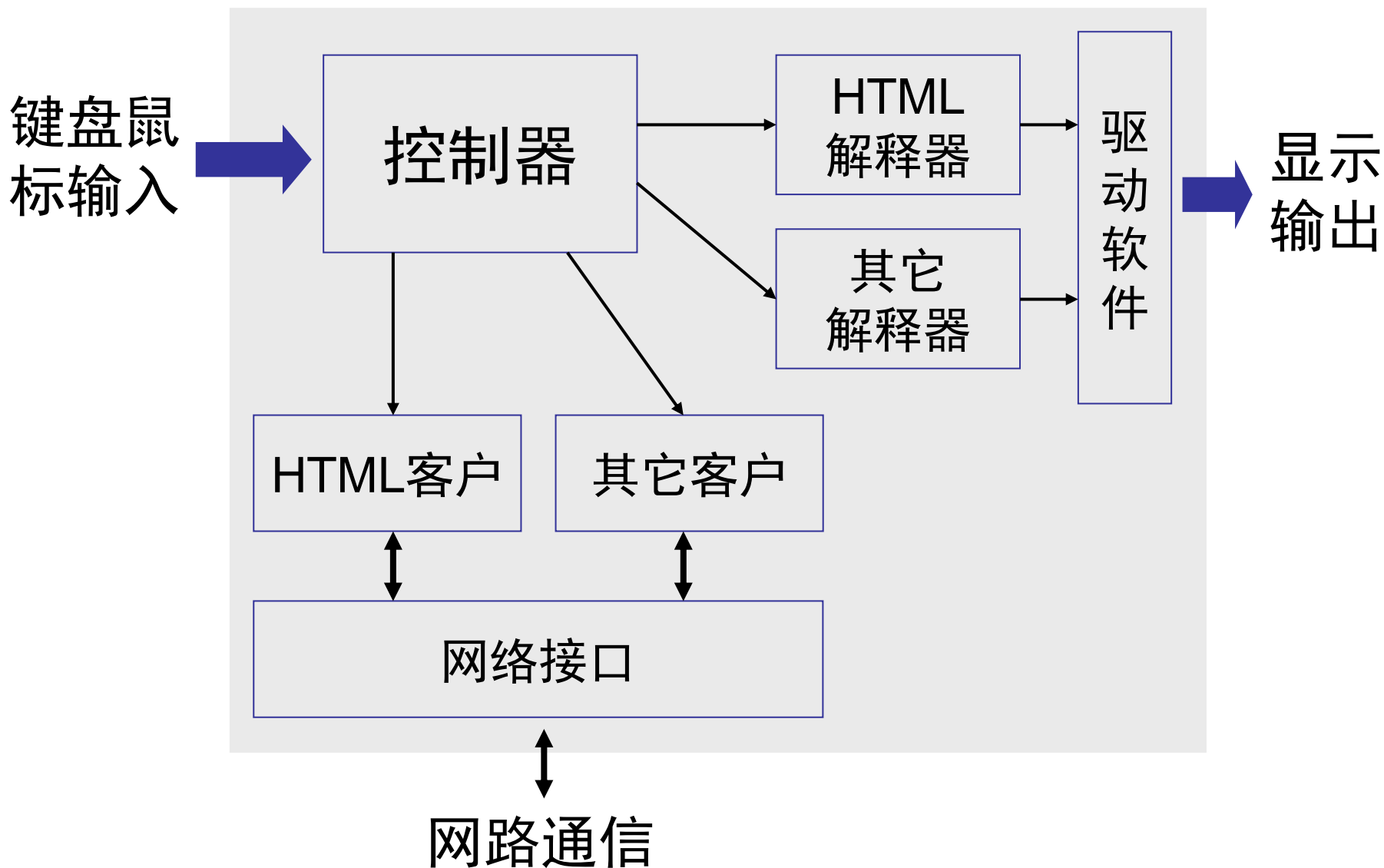
## ■ POST：发送数据给服务器

- 服务器将该数据添加到指定的项上

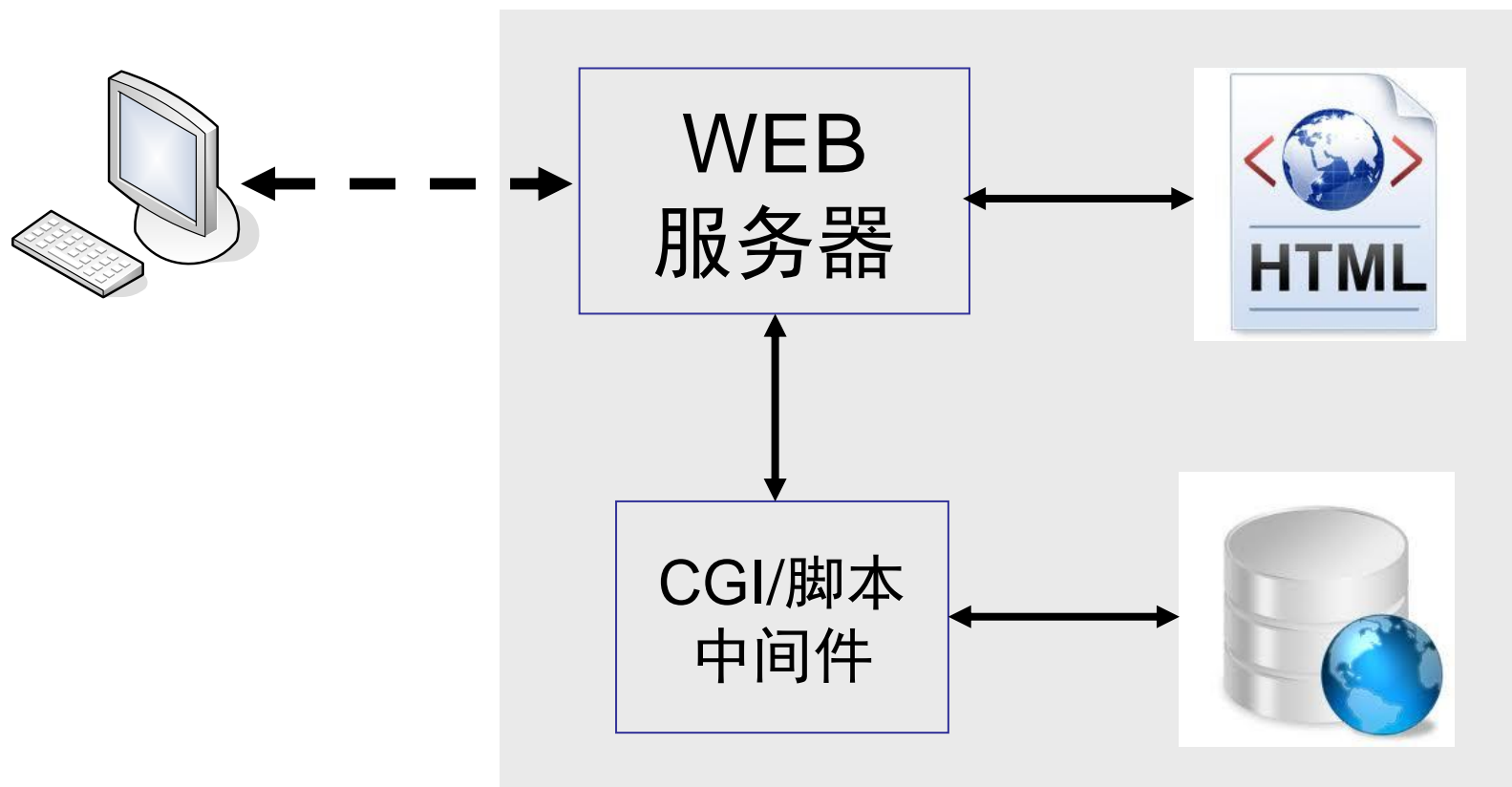
## ■ PUT：发送数据给服务器

- 服务器用该数据完全替代指定项

## 7.5.3 浏览器结构



## 7.5.4 WEB服务器结构



## 7.5.5 简单示例

