

计算机网络原理与实践（第2版）配套课件
机械工业出版社 2013年

第7章 应用层



本章内容

- 7.1 应用层的基本概念
- 7.2 网络应用的工作模式
- 7.3 因特网上的域名机制
- 7.4 因特网上的基本应用
- 7.5 因特网上的新型应用




- 网络应用进程之间的消息交互由谁来进行约定？
- 谁来规定应用进程之间所传递信息的语义、语法和顺序？
- 网络应用主要有哪些工作模式？



7.1 应用层的基本概念

应用层：规定应用进程在通信时所遵循的协议。


因特网的应用层协议主要有：

- 文件传输协议：**FTP**。用于实现因特网中交互式文件传输功能。
 - 远程终端协议：**Telnet**。用于实现互联网中远程终端功能。
 - 电子邮件协议：如简单邮件传输协议(**SMTP**)、邮局协议(**POP**)。用于实现因特网中电子邮件传送功能。
- 

- 网络管理协议：如简单网络管理协议(SNMP)、动态主机配置协议(DHCP)。用于管理与监视网络设备。
- 域名解析协议：DNS。用于实现网络设备名到IP地址映射的网络服务。
- 超文本传输协议：HTTP。用于WWW服务。



应用层协议的特点

- 应用层协议大多采用客户/服务器方式。
 - 应用层协议通常定义客户和服务进程之间在通信时所遵循的规范。
 - **客户（Client）和服务器（Server）** 都是指通信中所涉及的两个应用进程，描述进程之间服务和被服务的关系。
 - 发起服务请求方是客户端进程，提供服务方是服务器进程。
- 

客户 / 服务器进程通信

- 客户端：

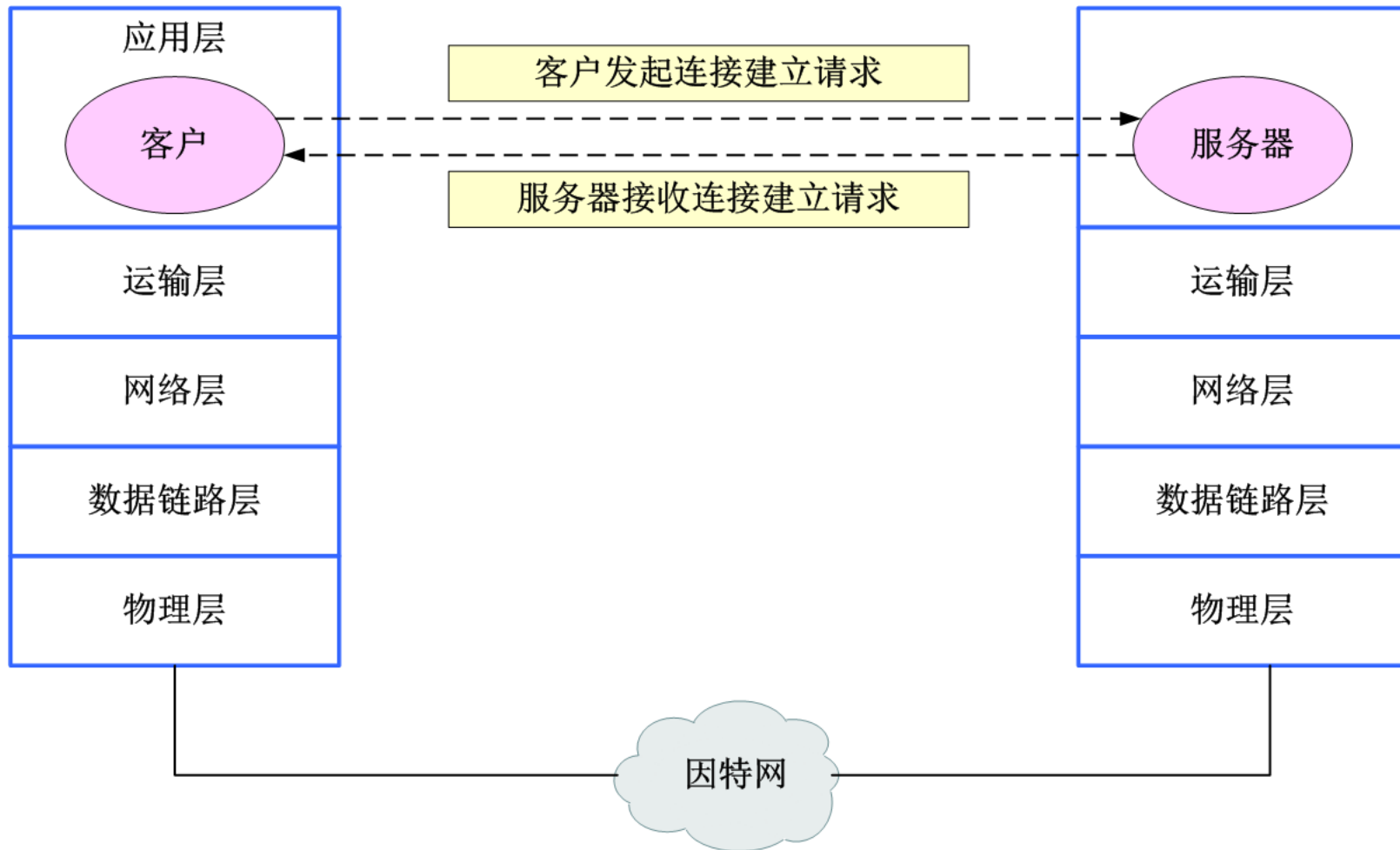
- (1) 在打算通信时主动向远程服务器发起通信。
- (2) 在进行通信时临时成为客户，但也可在本地进行其它计算。
- (3) 可与多个服务器进行通信。
- (4) 不需要特殊的硬件和很复杂的操作系统。

- 服务器：

- (1) 当系统启动时即自动调用并一直不断地运行着。
- (2) 被动地等待并接受来自客户的通信请求。
- (3) 可同时处理多个远程或本地客户的请求。
- (4) 一般需要强大的硬件和高级的操作系统支持。



客户/服务器进程通信示意图




7.2 网络应用的工作模式

网络应用软件的工作模式：


- C/S （ Client/Server） 工作模式
- B/S （ Browser/Server） 工作模式
- P2P （ Peer to Peer） 工作模式



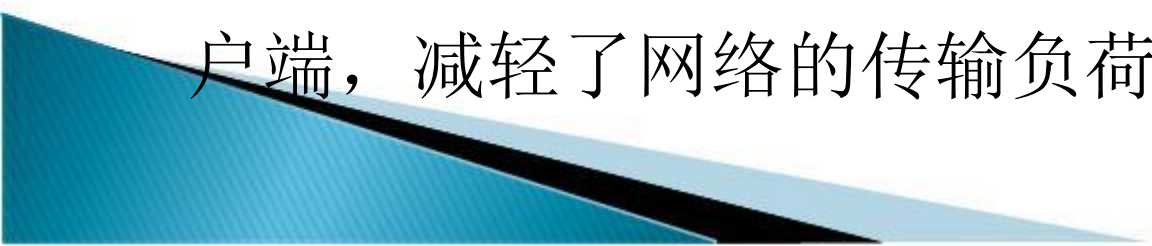
7.2.1 C/S工作模式

- 任务合理分配到客户端和服务服务器端来实现，有效降低系统的通信开销。
 - 服务器是整个应用系统的资源存储、用户管理以及数据运算的中心。大部分的运算工作由服务器完成。
 - 客户机则主要完成请求的传递以及界面的表示，对服务器有相当程度的依赖。
- 

C/S工作模式要点

- 在客户/服务器模式中，客户机与服务器扮演着完全不同的角色，其工作模式如下：
 - (1) 客户机向服务器发送请求。
 - (2) 服务器收到请求后，对请求进行处理。
 - (3) 服务器将处理结果返回给客户机。
 - (4) 客户机收到结果，将其以一定格式形成界面表示。
- 

C/S工作模式的特点


- C/S结构是数据库与网络技术发展相结合的结果。
 - 服务器控制管理数据的能力已由文件管理方式上升为数据库管理方式。服务端集中进行共享数据库的管理和存取，
 - 服务器注重数据定义及存取安全备份及还原，并发控制及事务管理。
 - 通过选择检索和索引排序等功能，能够通过网络把用户所需的处理后的数据而不是整个文件传送到客户端，减轻了网络的传输负荷。
- 

C/S工作模式的局限

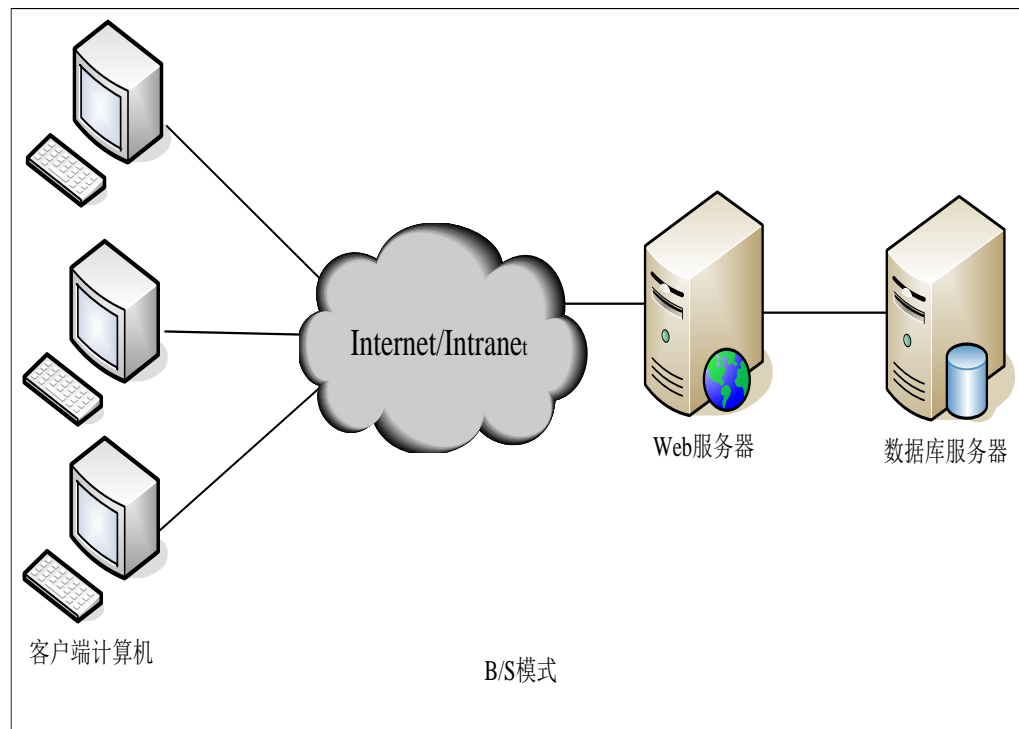
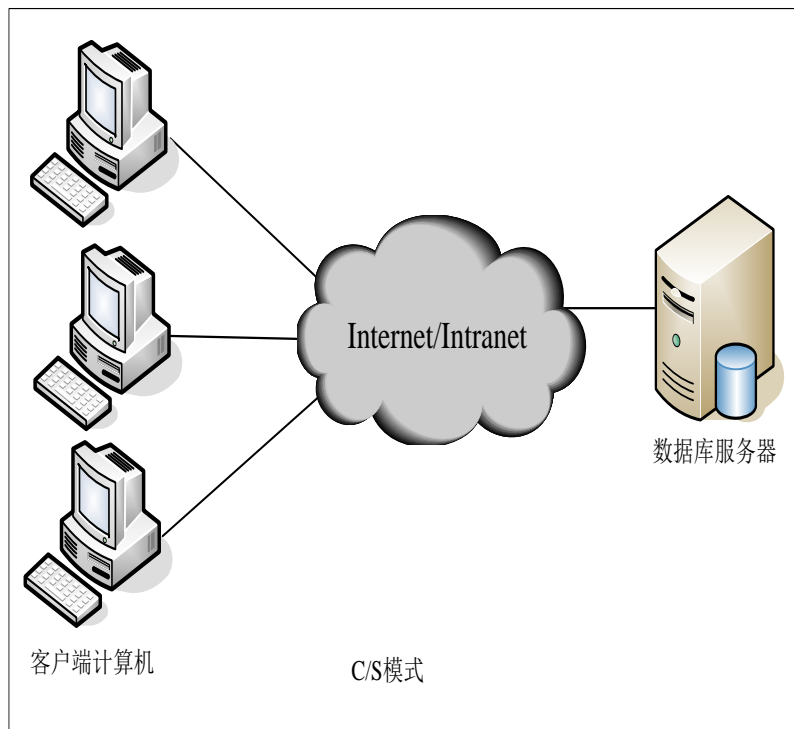
- 客户端软件需要针对不同的操作系统开发不同版本的软件。
- 客户端的部署和升级比较烦琐。
- 难适应百台电脑以上规模的局域网用户同时使用。




7.2.2 B/S工作模式

- B/S工作模式是随着Internet技术的兴起，是对C/S结构的一种变化或者改进。
 - 用户界面通过WWW浏览器来实现的，极少部分事务逻辑在前端（Browser）实现，主要事务逻辑在服务器端（Server）实现。
 - 服务器：Web服务器和数据库服务器，三层结构大大简化了客户端载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本。
- 


C/S模式与B/S模式



B/S工作模式要点

- (1) 客户端运行浏览器软件，浏览器以超文本传输协议向Web服务器发送请求。
 - (2) Web服务器接受客户端请求后，将请求转化为SQL语法，并交给数据库服务器。
 - (3) 数据库服务器收到请求后，验证其合法性，并进行数据处理，然后将处理后的结果返回给Web服务器。
 - (4) Web服务器将得到的所有结果进行转化，变成HTML文档形式，转发给客户端浏览器以Web页面的形式显示结果。
- 

B/S工作模式的特点

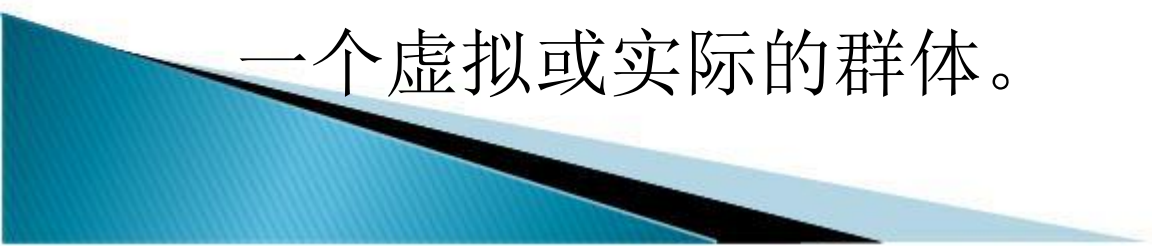
- 不需要在客户端额外安装软件，只需要浏览器就可以访问服务器，是一种瘦客户端的体系结构。
 - 能有效地保护数据平台和管理访问权限，服务器数据库也很安全。
 - 常见应用开发技术ASP/PHP/JSP等，方便、快捷、高效。
- 

C/S与B/S工作模式的比较


- C/S模式为两层结构，软件开发成本较高，客户端的维护和升级不方便。
- B/S模式是对C/S模式的一种扩展。瘦客户端，软件应用的业务逻辑完全在应用服务器端实现，用户表现完全在Web服务器实现。



7.2.3 P2P工作模式

- 近年在Internet上实施网络应用的新模式。
 - IBM的解释：P2P系统是由若干互联协作的计算机构成，且至少具有如下特征之一：
 - ✓ 系统依存于边缘化（非中央式服务器）设备的主动协作，每个成员直接从其它成员而不是从服务器的参与中受益。
 - ✓ 系统成员同时扮演服务器与客户机两种角色。
 - ✓ 系统应用的用户能够意识到彼此的存在，并构成一个虚拟或实际的群体。
- 

P2P工作模式特点

- 服务器与客户端的界限淡化或者消失。
 - 每个参与应用的结点均以“平等”的方式共享其它结点的共享资源，如CPU、存储空间等。
 - 在P2P系统中，实体一般既是资源的请求者，又是资源的提供者，即同时扮演客户机和服务器两种角色。
 - 从结构上看，P2P系统是分布式的。
- 

两类P2P系统

- 目前存在两类P2P系统：
 - 混杂P2P系统：由客户机与中央服务器构成，其典型案例为Napster。
 - 纯粹P2P系统：完全由客户机构成，其典型案例为Gnutella。



Napster


- MP3共享软件，是最早出现的P2P应用之一，第一代P2P软件。
- 非纯粹的P2P系统，它通过中央服务器（目录服务器）保存所有用户上传的音乐文件索引和存放位置的信息。
- 实现了文件查询与传输的分离，可节省中央服务器的带宽消耗，减少系统的文件传输延时。
- 中央服务器为瓶颈，服务器失效会导致系统瘫痪。
- 没有提供有效的安全机制。
- 后来陷入诉讼危机（版权问题）。

Napster（混杂P2P系统）的工作模式


系统由客户机与中央服务器（目录服务器）构成：

- （1）目录服务器记录在线用户的IP地址、端口号以及网络中MP3文件的目录信息。
- （2）客户机向目录服务器发送搜索某MP3文件请求，请求得到其它客户机的网络地址。
- （3）目录服务器根据请求搜索目录数据库，找到包含该文件的其它客户机，将其地址发送给该客户机。
- （4）客户机依据这些地址，向其它客户机发送请求。
- （5）其它客户机收到请求后，对请求进行处理，并将结果返回给发送方。

Napster（混杂P2P系统）的特点

- 与传统C/S模式相比，混杂的P2P系统只设目录服务器，为客户机的寻址提供服务，而不承担主要的运算工作。
 - 系统中的每个客户机均可以作为服务器，接受其它客户机发送的请求，为其它客户机提供服务。
 - 整个系统对中央服务器的依赖性明显降低。
- 

Gnutella

- 文件共享系统，第二代P2P软件，将P2P的理念向前又推进一步。
 - 真正意义上的对等分布式网络应用。
 - 纯粹的P2P系统，没有目录服务器，用户只要安装了该软件，立即变成一台能够提供完整目录和文件服务的服务器，并会自动搜寻其它同类服务器，从而联成一台由无数PC组成的网络超级服务器。
- 

Gnutella的工作模式

Gnutella系统中没有服务器，完全由客户机构成：

- (1) 每个客户机都维护一个相邻客户机列表，当客户机A需要搜索文件时，它向所有相邻客户机发送搜索请求。
- (2) 相邻客户机再将客户机A的搜索请求转发给它们各自相邻的客户机，该转发工作将持续进行，直到系统中所有客户机均收到该搜索请求。
- (3) 能提供该搜索文件的客户机按原路向客户机A发送其IP地址和端口号。



(4) 客户机A接收到响应消息后，依据一定规则（如：路径最短规则等）与相应的客户机建立连接，下载文件。

- 根据以上的描述可以看出，在纯粹的P2P系统中，没有目录服务器，每个客户机都具有发现其它客户机的能力。



P2P与C/S工作模式的比较

(1) 安全性

- C/S模式：服务器是整个应用系统的资源存储、用户管理和数据运算中心，只需确保服务器的安全，就能有效地保障整个系统的安全，安全性较高。
- P2P模式：每个结点既可以是客户机又可以是服务器，因此系统的安全管理比较困难。



P2P与C/S工作模式的比较

(2) 资源可维护性

- C/S模式：共享资源通常集中存放在服务器上，资源的查找更新较为简单，数据的备份与恢复也容易实现，资源可维护性较好。
- P2P模式：共享资源可以存放在系统中的任何一台客户机上，资源的查找更新比较复杂，数据备份工作也更为困难，可维护性差。



P2P与C/S工作模式的比较


(3) 可扩充性

- C/S模式：服务器性能会随着系统客户端数量的增多而下降，因而制约了客户端数目的增长，系统可扩充性较弱。
- P2P模式：每个客户机都具有很强的处理功能，极少依赖或不依赖中央服务器，因此，客户机的增加不会导致服务器的性能下降，系统可扩充性强。




P2P与C/S工作模式的比较

(4) 可持续性

- C/S模式：大部分运算工作由服务器承担，当服务器发生故障时，所有客户端都无法进行正常工作，整个系统将陷于瘫痪之中，可持续性差。
 - P2P模式：每个客户机均可作为服务器，当某台为其它客户机提供服务的服务器发生故障时，其它客户机可承担该服务器的角色，接管发生故障的服务器的的工作，可持续性高。
- 

P2P与C/S工作模式的比较

(5) 网络通讯性能

- C/S模式：网络通讯主要集中在客户机与服务器之间，客户机与客户机之间一般并无直接交互行为，网络流量分布集中。
 - P2P模式：对于纯粹的P2P系统，客户机将向系统中的所有结点发送请求，寻找提供服务的结点，而对于混杂P2P系统，客户机也将首先访问中央服务器，获取其它客户机的地址，而后进行类似C/S模式的客户机与服务器的数据通讯。**P2P模式需要占用更多的网络带宽，网络通讯更频繁。**
- 

P2P与C/S工作模式的比较

- 结合实际情况选择系统的工作模式
- 对于需要强调集中控制、集中管理以及安全性较高的系统，可采用C/S模式
- 对于强调网络数据的直接共享以及网上直接交互的系统，则应选择P2P模式。




7.3 因特网上的域名机制

- 分层的域名空间
- 域名服务与域名解析



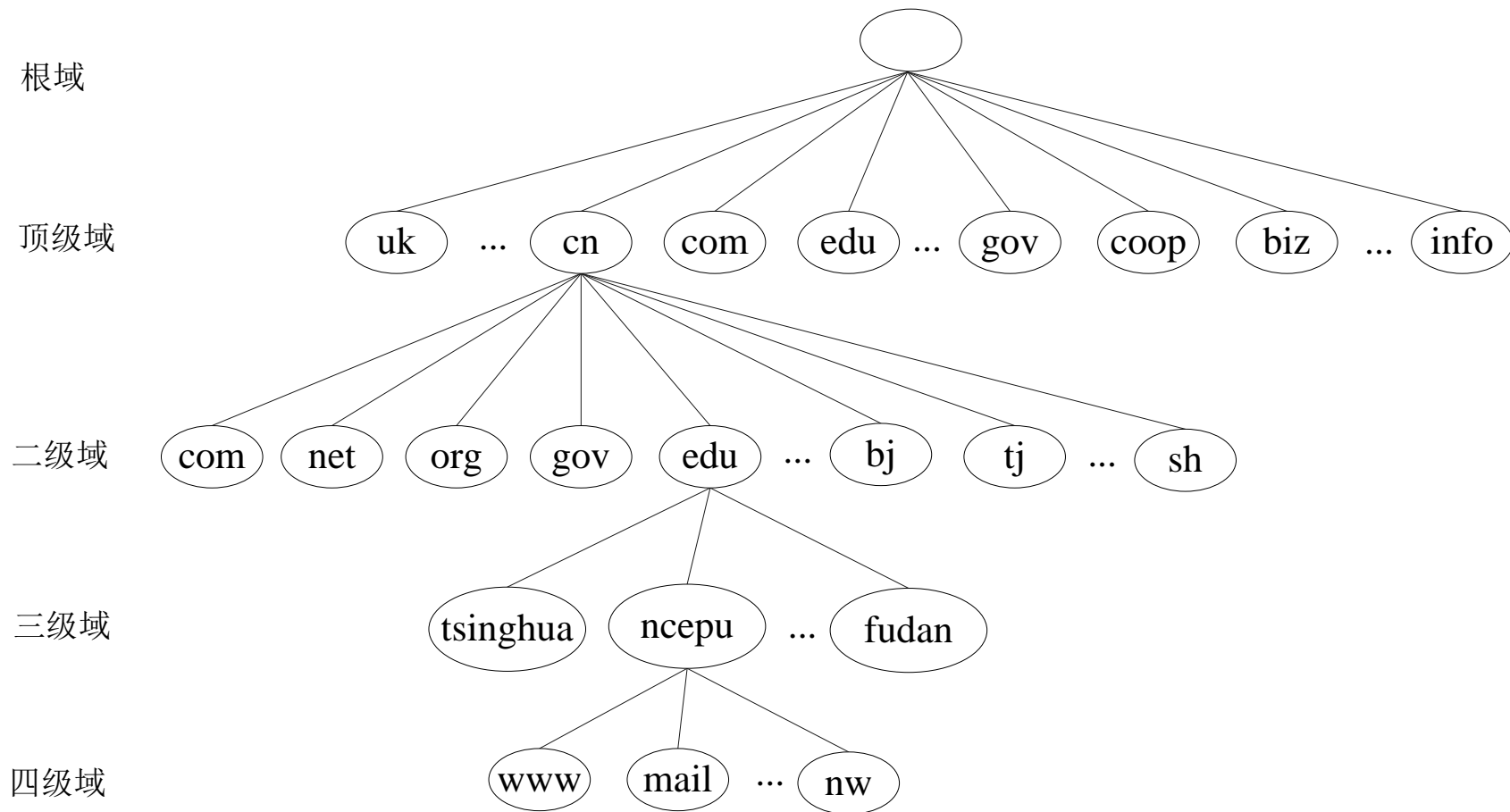
7.3.1 分层的域名空间

- 域名系统（Domain Name System, DNS）是采用层次结构和分布式数据库，在因特网中将域名映射到IP地址的系统。
 - IETF RFC 1034和1035对域名系统做出了明确的定义和规范。
 - 为满足因特网中大规模的名字空间，域名系统采用了层次名字空间（hierarchical namespace）。
- 

- 层次名字空间以分布的方式实现域名和IP地址的映射，同时支持对名字空间分配和管理的分级控制。
- 中心管理机构负责名字空间的统一划分，并为下面各级管理机构进行授权，各级管理机构负责对其授权范围内的名字空间进行分配和管理。
- 层次名字空间的域名系统采用域名分级命名方案。




域名的层次结构



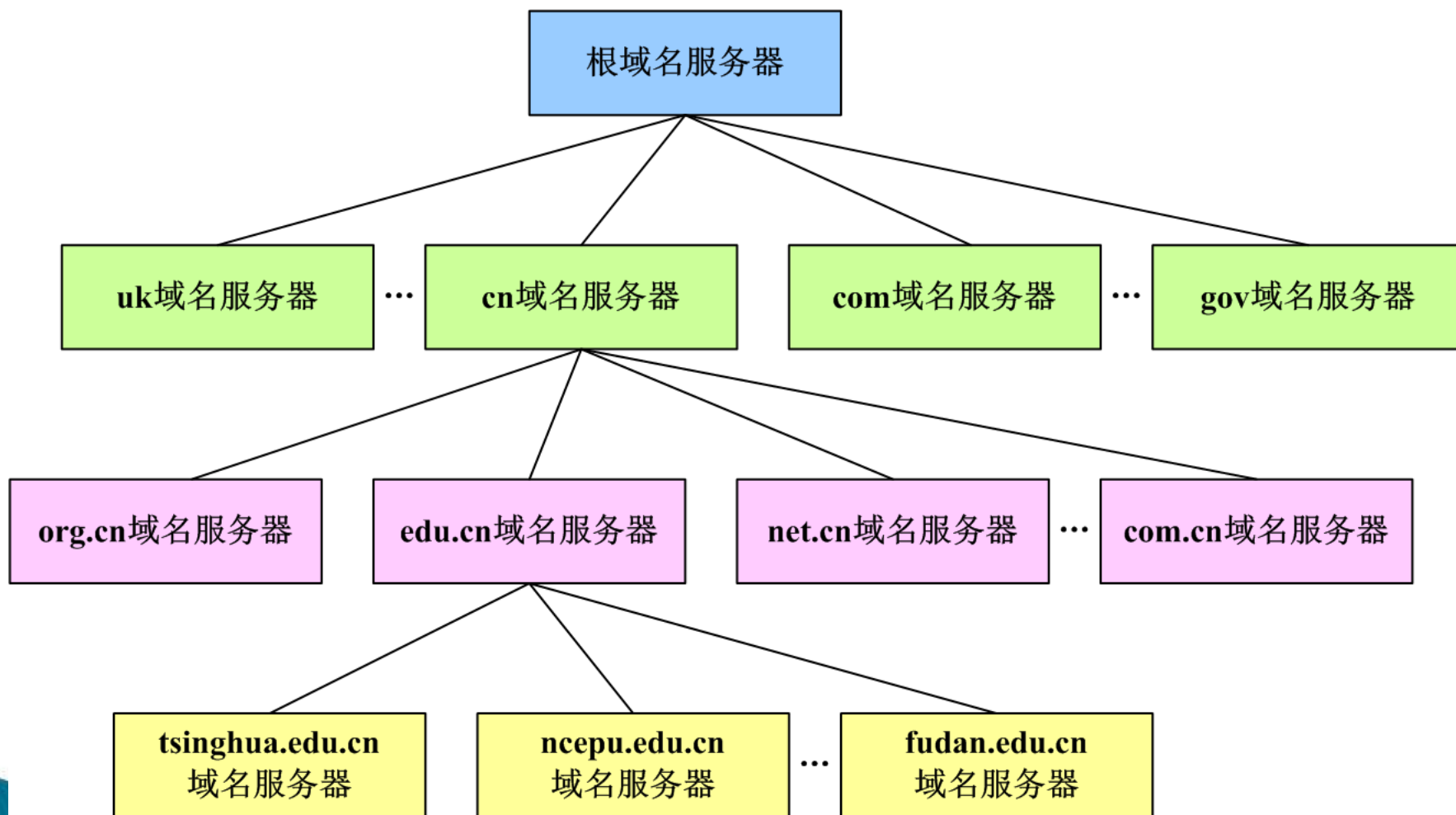
常用顶级域名及其描述

域名	描述
aero	航空运输
com	商业机构
edu	教育机构
gov	政府机构
int	国际组织
mil	军事机构
net	网络支持机构
org	其他组织
cn	中国
uk	英国

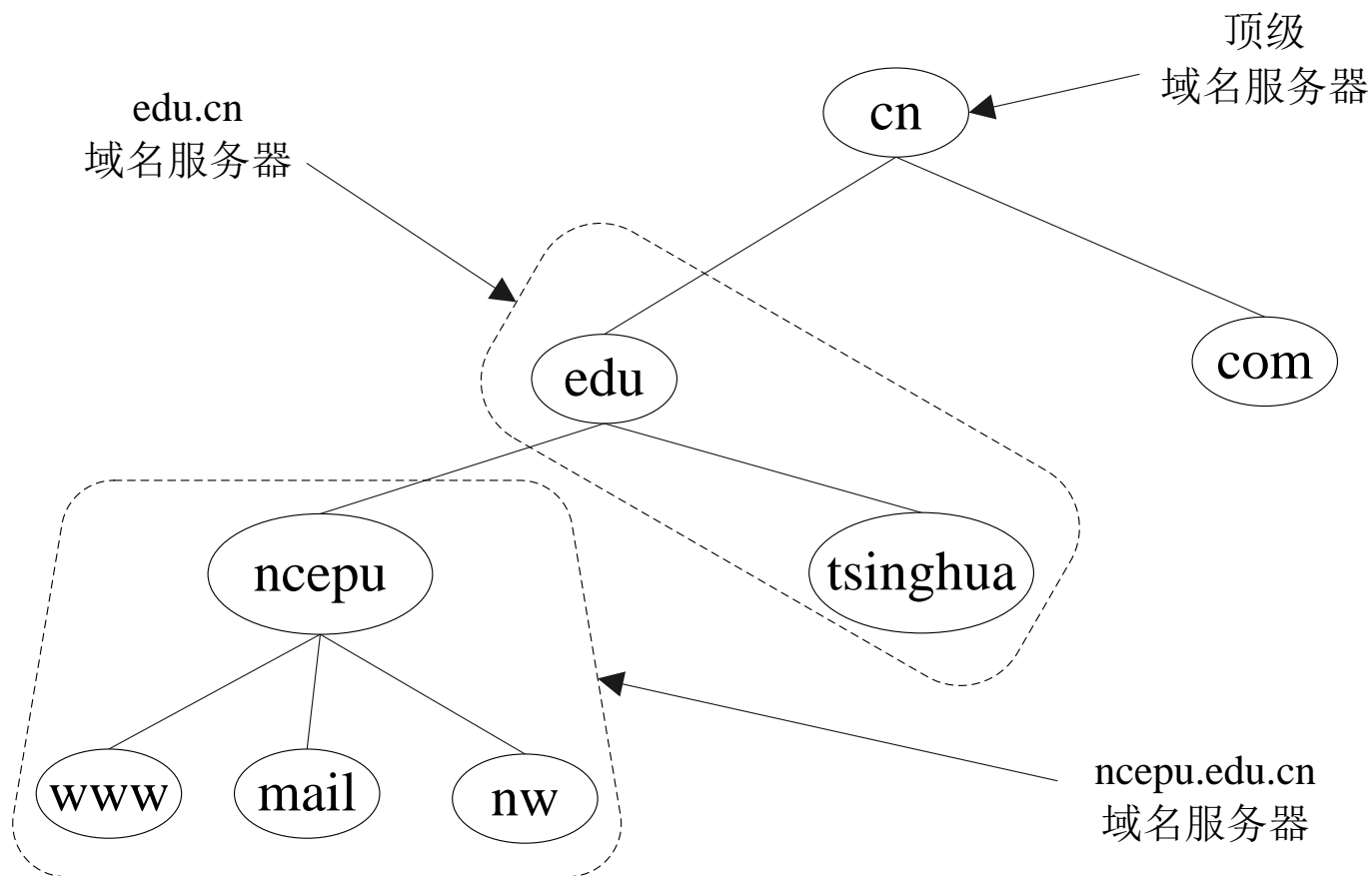
7.3.2 域名服务器与域名解析

- 域名服务器（Domain Name Server, DNS）：
：装有从域名到IP地址转换程序的机器。
 - 分布式的域名服务：分布式的域名服务器相互协作完成域名解析；域内多台域名服务器，防止服务器单点失效。
 - 域名解析：将域名转换为与其对应的IP地址的过程。
- 

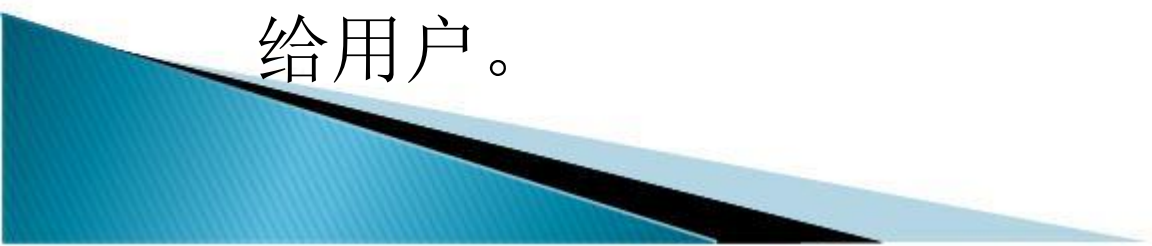
域名服务器的层次结构



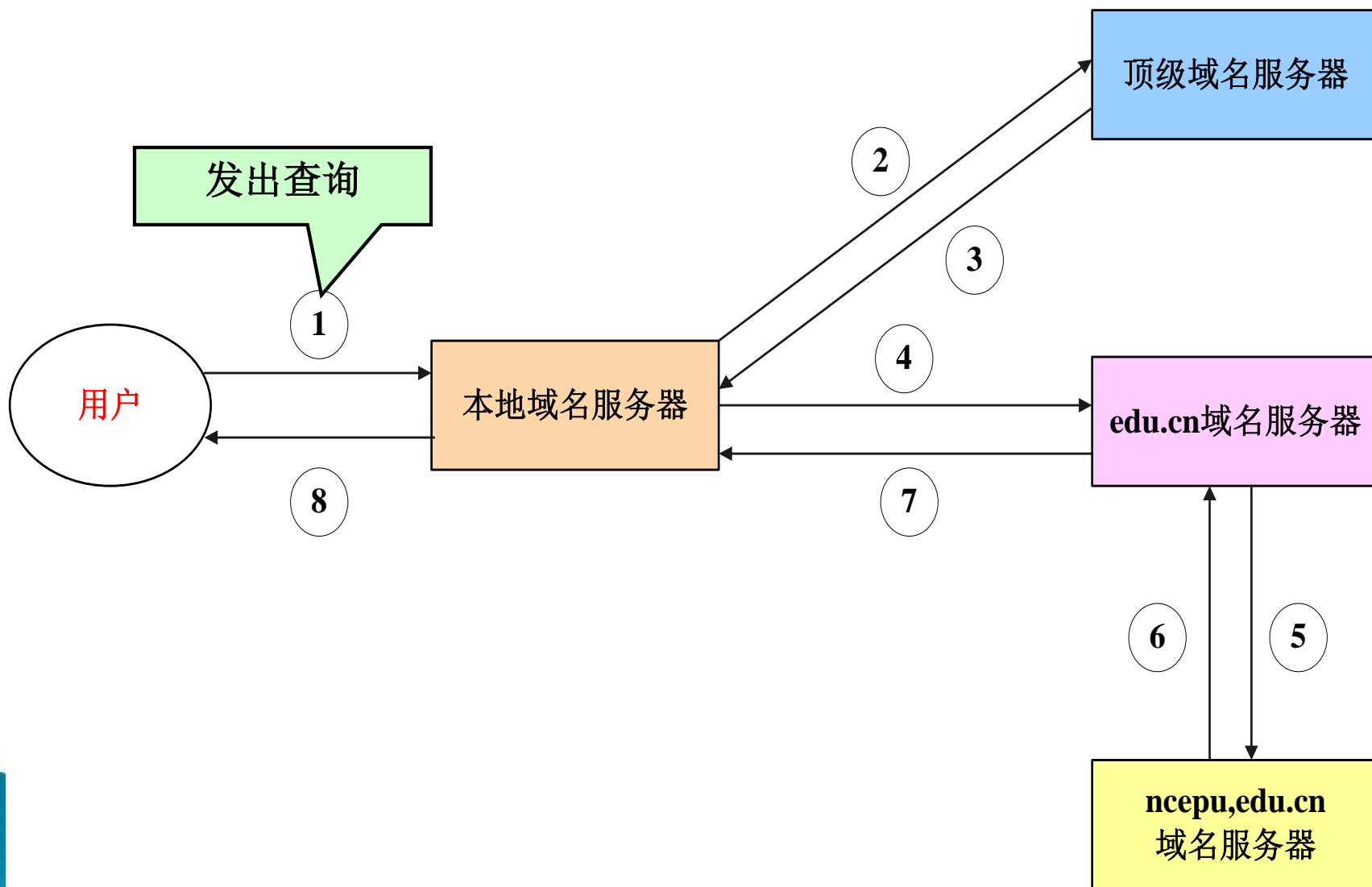
划分为区域后的域名层次结构

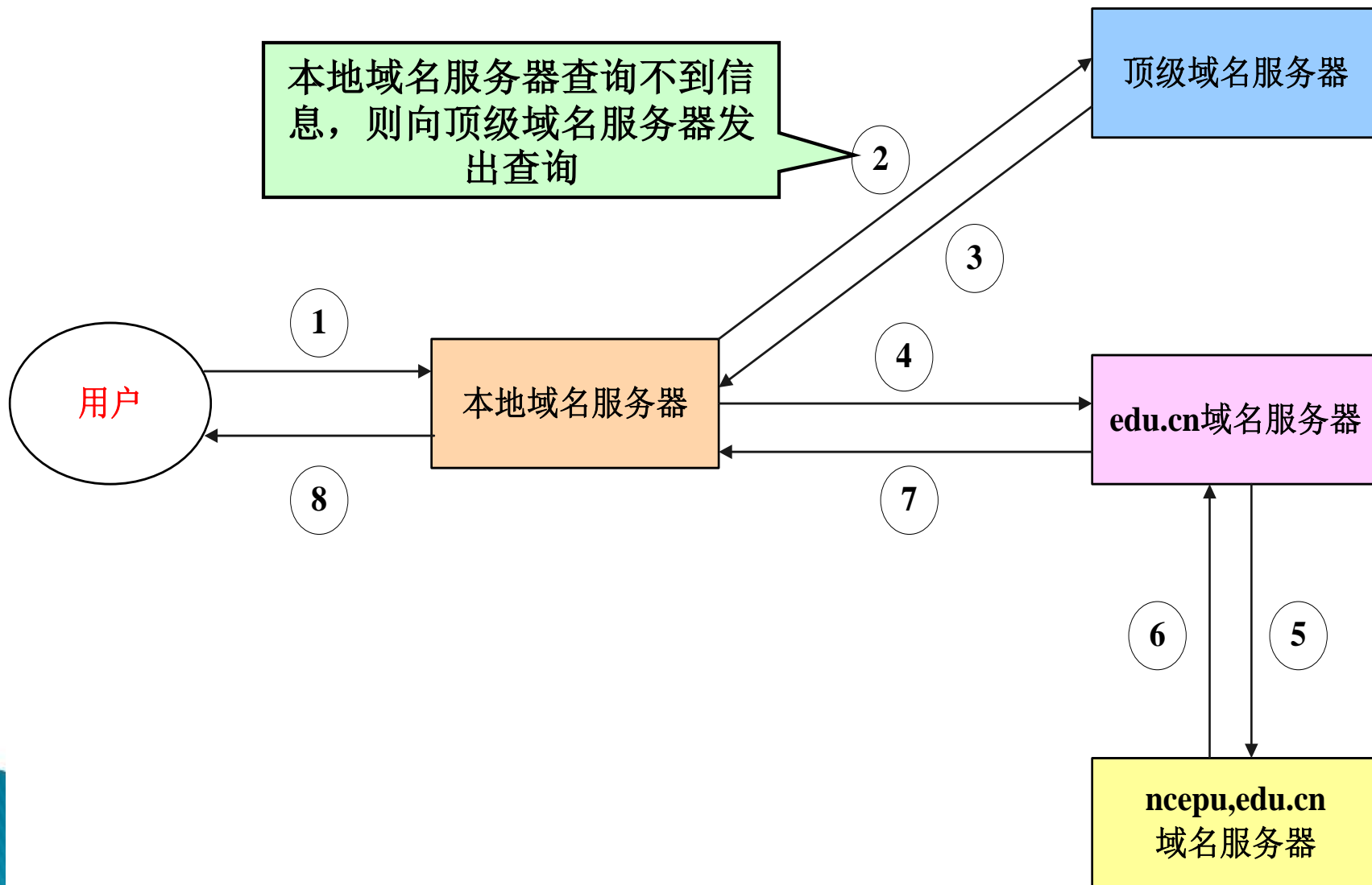


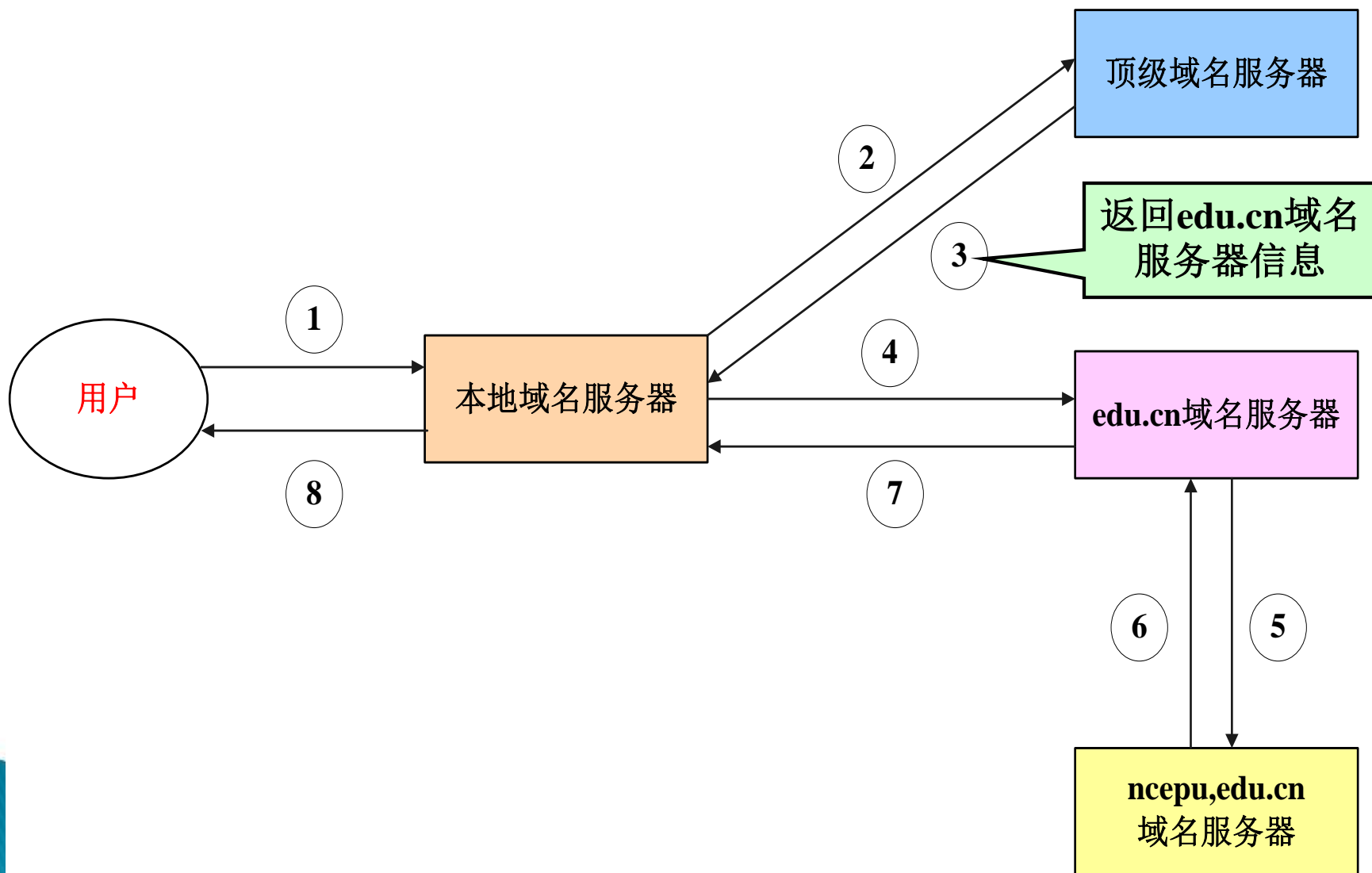
域名解析过程

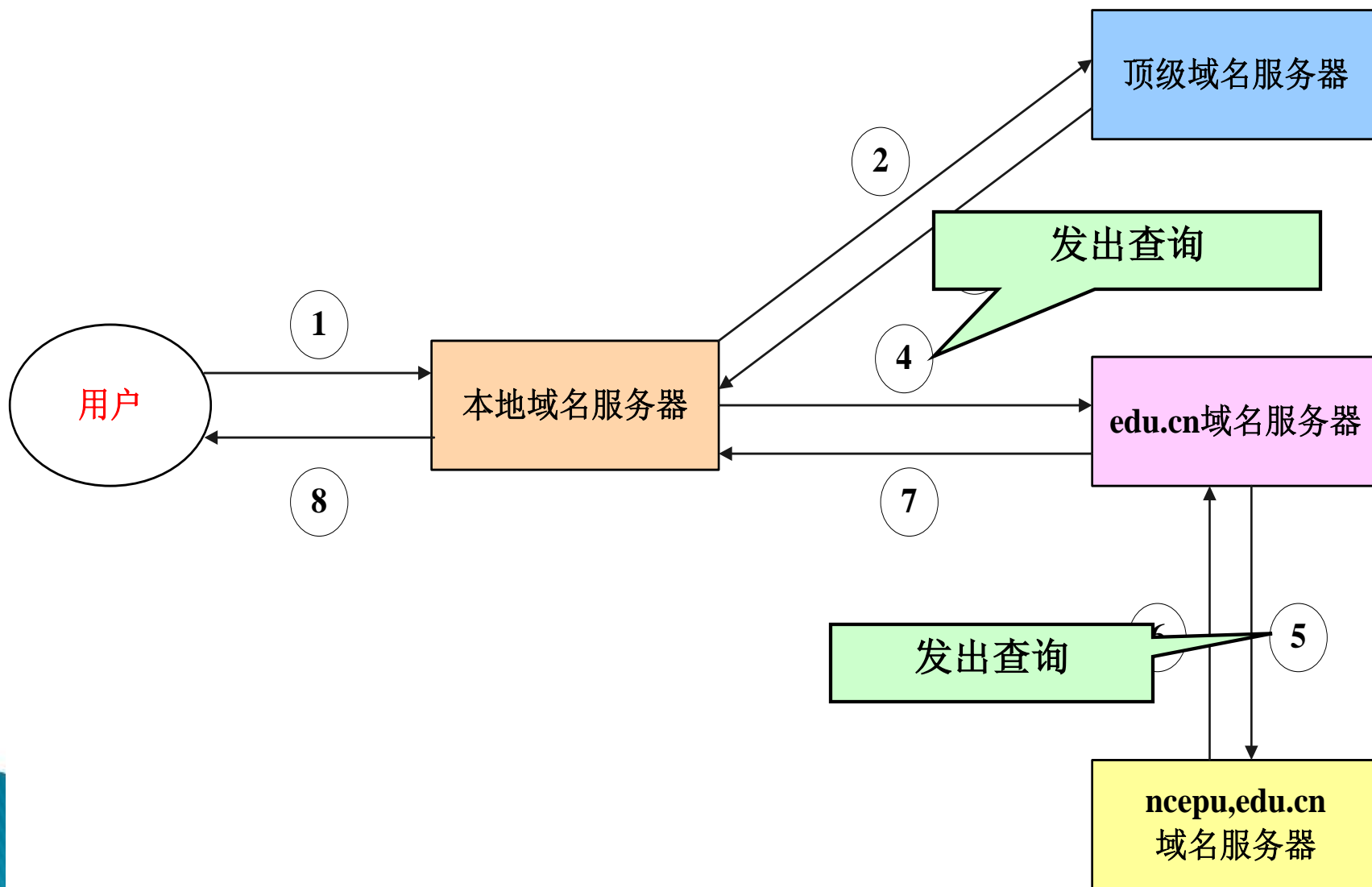
- 当用户发出一个域名解析的请求时，域名服务器将向用户返回与其相对应的IP地址。
 - 域名解析从本地域名服务器开始。如果本地域名服务器不能解析该域名，可采用两种方法：
 - 递归解析（recursive resolution），本地DNS负责和能够解析该域名的服务器进行通信，并将查询结果返回用户。
 - 迭代解析（iterative resolution），本地域名服务器将能进行下一步解析的域名服务器地址返回给用户。
- 

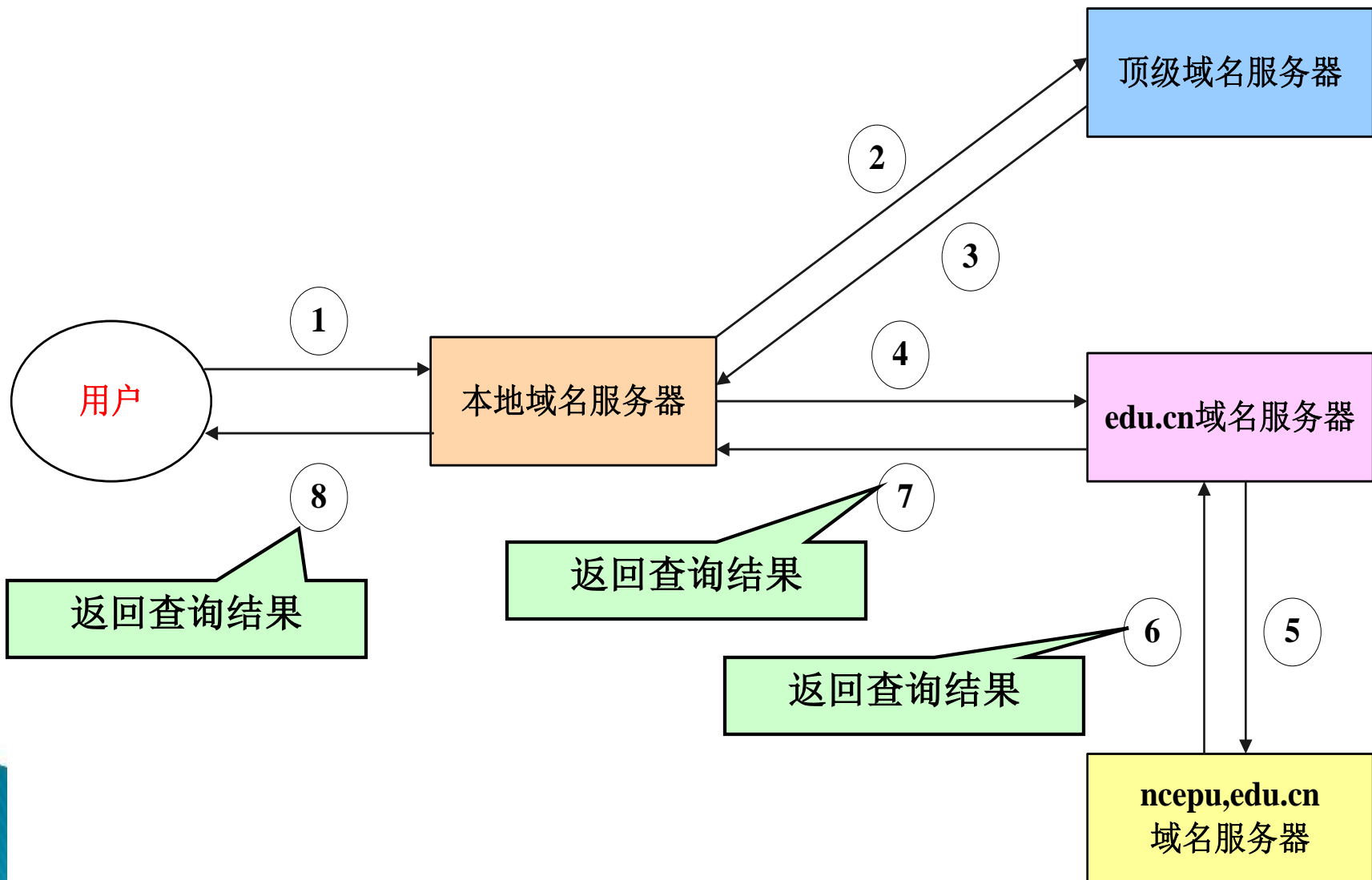
递归和迭代结合的域名解析过程













- 因特网域名服务器采用高速缓存（caching）来优化查询的开销。
- 每个域名服务器都维护一个高速缓存，存放最近使用过的名字以及从何处获得该名字映射信息的记录。
- 当用户发出域名解析的请求时，可以从高速缓存中获得最近被解析过的域名信息，从而减少网络开销。



7.4 因特网上的基本应用

- 电子邮件
 - 万维网
 - 文件传输服务
 - 远程登录
 - 动态主机配置协议
 - 网络管理及简单网络管理协议
- 

7.4.1 电子邮件

- 电子邮件（Electronic Mail，简记为E-mail），是因特网应用最广泛并最受欢迎的服务之一。
 - 电子邮件是建立在计算机网络基础上的一种通信形式，它为因特网用户提供读写、发送和接收邮件的功能，并允许单个用户或群组用户之间进行通信。
- 

1. 电子邮件的组成结构

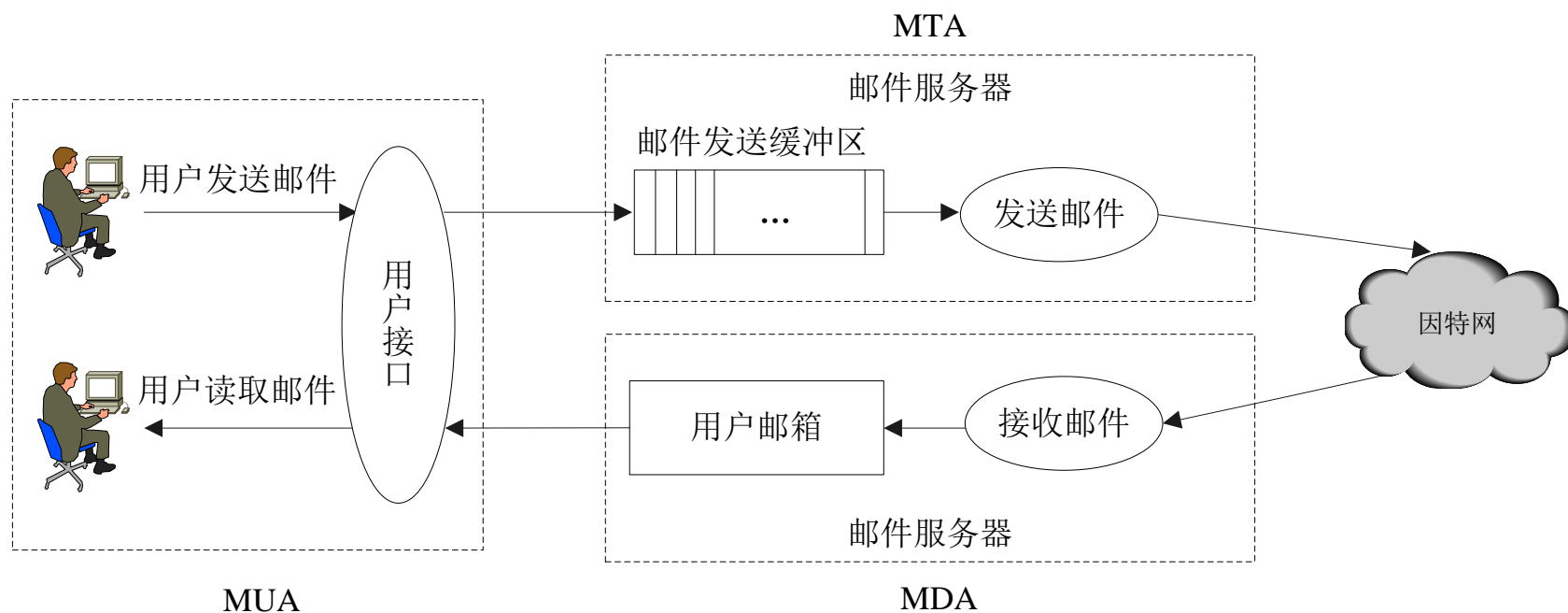
- 电子邮件的工作过程遵循C/S模式，发送方构成客户端，接收方构成服务器端。
- 工作过程可以与普通邮件的收发相类比
 - 收件人的姓名与地址（即邮件地址
 - 发送方的邮件服务器
 - 接收件方邮件服务器



- 电子邮件系统由三个部分组成：
 - ✓ 邮件用户代理（Mail User Agent, MUA）
 - ✓ 邮件传输代理（Mail Transport Agent, MTA）
 - ✓ 邮件投递代理（Mail Delivery Agent, MDA）



电子邮件的组成结构



2.电子邮件格式

- IETF RFC 2822定义。
- 电子邮件的地址格式由一个字符串组成，其格式如下所示：

local-name @ domain-name

- 该字符串通过符号@（通常读为“at”）分隔为两个部分
 - domain-name是目的邮件服务器的域名
 - local-name是用户在该邮件服务器上的邮箱名。

- 电子邮件的标准格式由两部分组成：
 - ✓信封
 - ✓信件内容
- 信件内容：分为首部（header）和主体（body）两个部分，中间用一个空行隔开。
 - 信件内容的首部：主要包括收信人地址、抄送人地址、主题、发送人地址和发信日期等信息。
 - 信件内容的主体：电子邮件所要传送的信息。

首部常用关键字及含义描述

关键字	含义
To	收信人地址
Cc	抄送人地址
Subject	主题
From	发信人地址
Date	发信日期
Bcc	暗送人地址
Reply-To	对方回信地址

3. MIME

- IETF RFC 2822中规定的电子邮件只能传输使用7位ASCII字符集表示的字符，而非英文字符、音视频和二进制格式文件等非ASCII数据不能在电子邮件中传输。
- 为了能够使用电子邮件传输非ASCII数据，多用途互联网邮件扩展（Multipurpose Internet Mail Extensions, MIME）对电子邮件标准进行扩展。



- MIME主要包括以下3个基本部分：
 - ✓ 首部行的扩充
 - ✓ 发送数据的类型
 - ✓ 转换编码方法



MIME新增5个字段名称及意义

名称	意义
MIME-Version	MIME的版本，目前的版本号为1.0
Content-Type	邮件内容所包含的数据类型
Content-Transfer-Encoding	邮件内容传输编码
Content-Id	邮件内容标识符
Content-Description	邮件内容描述，类似于邮件主题

MIME中的数据类型及子类型

内容类型	子类型	意义
文本（text）	plain	纯文本
	richtext	格式化文本
图像（image）	jpeg	JPEG图像
	gif	GIF图像
	png	PNG图像
音频（audio）	basic	一段声音记录
视频（video）	mpeg	MPEG视频
应 用 （application）	octet-stream	二进制数据流
	postscript	Adobe PostScript文档
	pdf	pdf文档
	msword	Microsoft Word文件
报文（message）	rfc822	RFC 822邮件
	partial	对邮件进行分割
	external-body	从网上获取邮件
多 部 分 （multipart）	mixed	内容中包含顺序排列的多种类型数据
	alternative	相同内容使用不同形式表示
	parallel	必须同时读取的几个数据类型
	digest	一个报文包含一组其他报文，每个部分为一个完整的RFC 822邮件

一个MIME邮件示例


```
From: lgd@ncepu.edu.cn
To: test@test.com.cn
MIME-Version: 1.0
Content-Type: multipart/mixed;boundary=ImageTransferTest
-- ImageTransferTest
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
test,
    邮件中为测试用的图片。
best
lgd
-- ImageTransferTest
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
...data for the image...
```

4.电子邮件协议

- 电子邮件协议包括：
 - ✓ 简单邮件传输协议
 - ✓ 邮件访问协议



1.简单邮件传输协议

- 简单邮件传输协议（Simple Mail Transfer Protocol, SMTP）定义了邮件服务器之间交换邮件的标准，用于发送方邮件服务器到接收方邮件服务器之间的邮件传输。
 - SMTP采用C/S模式，发送邮件的一方为客户端，接收邮件的一方为服务器端。
- 

- SMTP规定了收发邮件的服务器之间交换信息的命令及应答格式，包括14条命令和21种应答信息。
- SMTP通信包括连接建立、邮件传输和连接关闭三个阶段。




一个SMTP传输邮件的示例

```
S: 220 ncepu.edu.cn Simple Mail Transfer Service Ready
C: HELO ncepu.edu.cn
S: 250 Hello ncepu.edu.cn
C: MAIL FROM: lgd@ ncepu.edu.cn
S: 250 Ok
C: RCPT TO: test@ ncepu.edu.cn
S: 250 Ok
C: DATA
S: 354 Start mail input; End with <CR><LF>.<CR><LF>
C: Subject: this is a simple mail
C: From: lgd@ ncepu.edu.cn
C: To: test@ ncepu.edu.cn
C:...
C: Hello,
C: This is a test.
C: Goodbye.
C: ...
C: <CR><LF>.<CR><LF>
S: 250 Ok
C: quit
S: 221 Bye
```

2. 邮件访问协议


- 邮件访问协议提供用户从本地邮件服务器读取邮件功能，有两个邮件访问协议：
 1. 邮局协议（Post Office Protocol v3, POP3）：
 - 电子邮件客户端与服务器建立TCP连接，经过用户验证后，下载服务器上未阅读的邮件，然后从邮箱中删除邮件。
 - 目前改进的POP3允许下载邮件后在服务器上保留邮件。
 - IETF RFC 1939对POP3做出了定义和规范。

2. 因特网报文访问协议（Internet Message Access Protocol v4, IMAP4）
- IMAP4是版本号为4的因特网报文访问协议
 - IMAP4允许用户通过多个地点在邮件服务器上直接访问邮件，也称联机的邮件访问协议。
 - 由IETF RFC 3501定义和规范。
- 


7.4.2 万维网

- 万维网（World Wide Web, WWW）是指在因特网上以超文本为基础形成的分布式的信息网，由以下3个部分组成：
 - ✓ 统一资源定位符
 - ✓ 超文本传输协议
 - ✓ 超文本标记语言

1. 万维网体系结构

- 万维网是一个庞大的信息网络集合，由大量的万维网页面组成，用户可以通过因特网访问万维网页面。
 - 万维网主要由万维网浏览器（web browser）和万维网服务器（web server）构成。
 - 万维网的工作过程遵循C/S模式，万维网浏览器构成客户端，万维网服务器构成服务器端。
- 

2. 统一资源定位符

- 每一个万维网页面都对应一个唯一的标识，该标识被称为统一资源定位符（Uniform Resource Locator，URL）。
 - 统一资源定位符是对从因特网上获取资源的位置和访问方法的一种简洁表示。
 - IETF RFC 1738对URL做出了明确的定义和规范。
- 

统一资源定位符的标准格式如下：


<通信协议>://<主机>:<端口号>/<文件路径>

- <通信协议>指明了访问资源时所使用的通信协议。
- <主机>是存放资源的网络主机的域名或IP地址。
- <端口号>是可选的协议端口号，如果省略则使用其默认端口号。每种通信协议都有各自默认的端口号，如http的默认端口号为80。
- <文件路径>表示资源存储的路径。

常用的通信协议及其意义

通信协议	意义
http	使用超文本传输协议传输
https	使用加密的超文本传输协议传输
File	访问本地计算机上的文件
ftp	使用文件传输协议传输
mailto	访问电子邮件地址
news	访问Usenet新闻组
gopher	使用Gopher协议传输

例如： <ftp://ftp.pku.edu.cn/pub/rfc/rfc2091.txt>

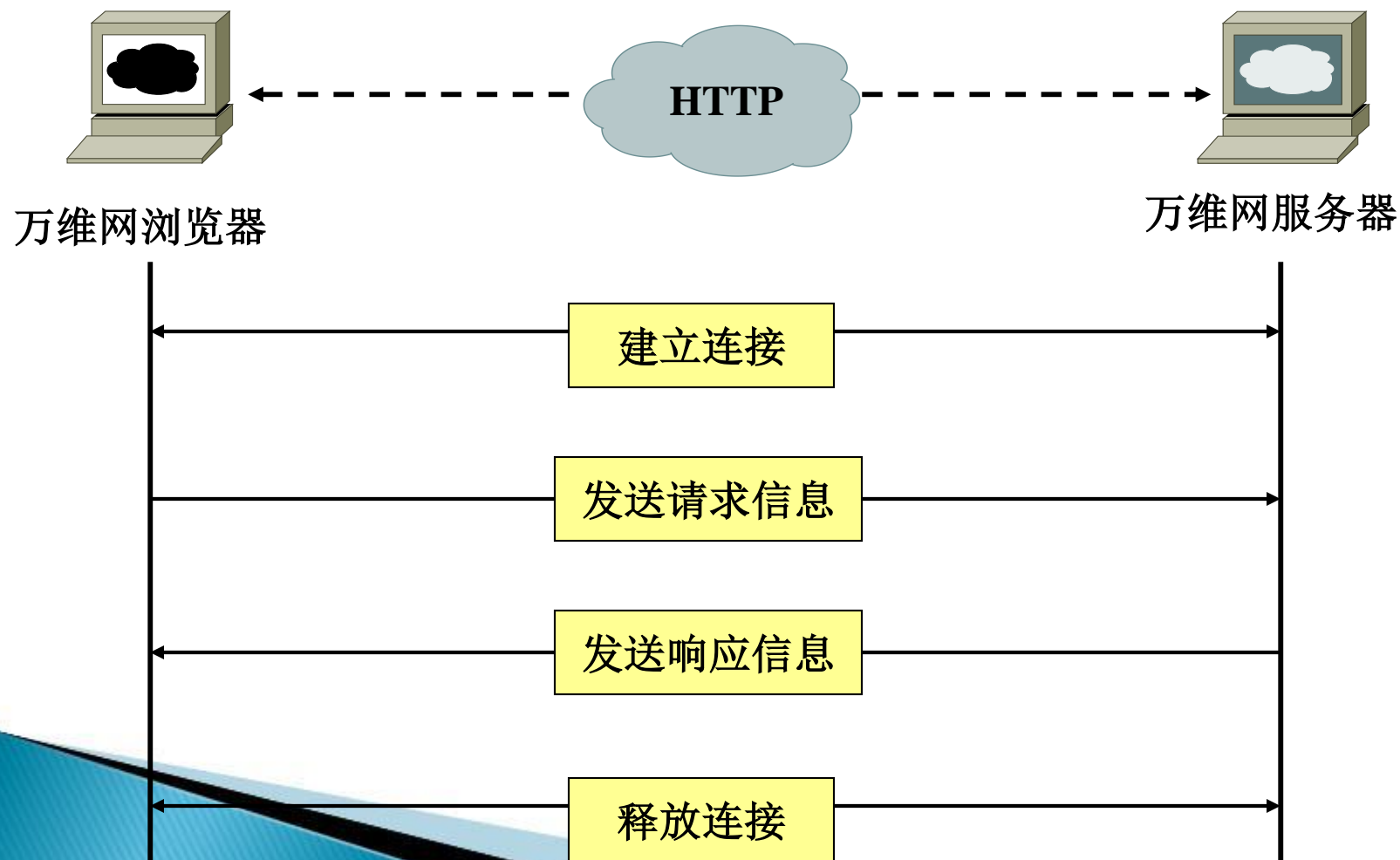


3. 超文本传输协议


- 超文本传输协议（HyperText Transfer Protocol, HTTP）是万维网浏览器和万维网服务器之间的通信协议。
- 因特网上应用最为广泛的协议之一。
- HTTP1.1由IETF RFC 2616定义和规范。



HTTP工作原理



HTTP请求信息与响应信息

- HTTP 请求信息：主要包括请求行、请求头部、空行和可选的消息体。
 - 其中，请求行包括请求类型、URL和HTTP版本。
 - HTTP 响应信息：主要包括状态行、响应头部、空行和可选的消息体。
 - 其中，状态行包括HTTP版本、状态代码和描述状态的短语。
- 

HTTP请求报文

请求方法	空格	URL	空格	HTTP 版本	回车换行符	请求行
头部字段名	:	空格	值	回车换行符	头部	
:						
头部字段名	:	空格	值	回车换行符		
回车换行符	空行					
消息体						

HTTP请求操作方法

名称	方法描述
OPTIONS	返回服务器针对特定文档所支持的HTTP请求方法
GET	请求读取服务器上指定URL下的文档
HEAD	请求获取服务器上指定URL下文档的状态信息
POST	向服务器发送信息
PUT	从客户端更新服务器文档
DELETE	请求服务器删除指定URL下的文档
TRACE	回送服务器反馈 http的请求信息
CONNECT	预留给代理服务器使用

HTTP 响应报文

HTTP版本		空格	状态码		空格	状态短语		回车换行符		状态行	
头部字段名		:	空格	值	回车换行符				}	头部	
		:									
头部字段名		:	空格	值	回车换行符						
回车换行符			空行								
消息体											

HTTP的状态代码

名称	类型	意义
1xx	消息	服务器接收请求，继续处理
2xx	成功	请求已成功被服务器接收、理解、并接受
3xx	重定向	需要继续操作以完成请求
4xx	客户错误	请求语法错误或者无法被执行
5xx	服务器错误	服务器不能响应某个正确的请求


HTTP 报文示例

一个HTTP请求报文

```
GET / example/image HTTP/1.1  
Accept: image/gif, image/jpeg
```

一个HTTP响应报文


```
HTTP/1.1 200 OK  
Content-Length: 1024  
Server: NCEPU  
Date: Sat, 8 May 2010 15:55:02 GMT  
Connection: keep-alive
```



4. 超文本标记语言

- 超文本标记语言（HyperText Markup Language, HTML）是用于设计和构成网页文档的主要语言之一。
- 用户使用HTML语言可以编写包括文本、图形、声音、动画以及连接等的万维网页面，使用HTML编写的超本文档称为HTML文档。



- 每个HTML文档总是以<HTML>开始，</HTML>结束，并使用标记<HEAD></HEAD>和<BODY></BODY>把文档分成两个部分。
 - <HEAD>与</HEAD>用于标记文件头部，文件头部包含文件的说明信息，它们并不和文件一起显示。
 - <BODY>与</BODY>用于标记文件的主体部分，包括标题、段落、列表、图形和超文本链接等内容。
- 

常用的HTML标记


标记	说明
<HTML>...</HTML>	声明这是用HTML写成的万维网文档
<HEAD>...</HEAD>	定义页面的首部
<TITLE>...</TITLE>	定义页面的标题
<BODY>...</BODY>	定义页面的主体
...	设置...为黑体字
<Hn>...</Hn>	定义一个n级题头
 	强制换行
<P>	一个段落开始
	插入一张图像，其文件名为...
X	定义一个超链接，连接的起点为X，目的地址为...


一个HTML文档示例

<HTML>	{HTML文档开始}
<HEAD>	{文档头部开始}
<TITLE>一个HTML示例</TITLE>	{“一个HTML示例” 是标题}
</HEAD>	{文档头部结束}
<BODY>	{文档主体开始}
<H1>一个HTML的简单例子</H1>	{“一个HTML的简单例子” 是一个一级题头}
<P>这个一个段落	{一个段落的开始}
 另起一行	{强制换行，另起一行开始}
</BODY>	{文档主体结束}
</HTML>	{HTML文档结束}



7.4.3 文件传输服务

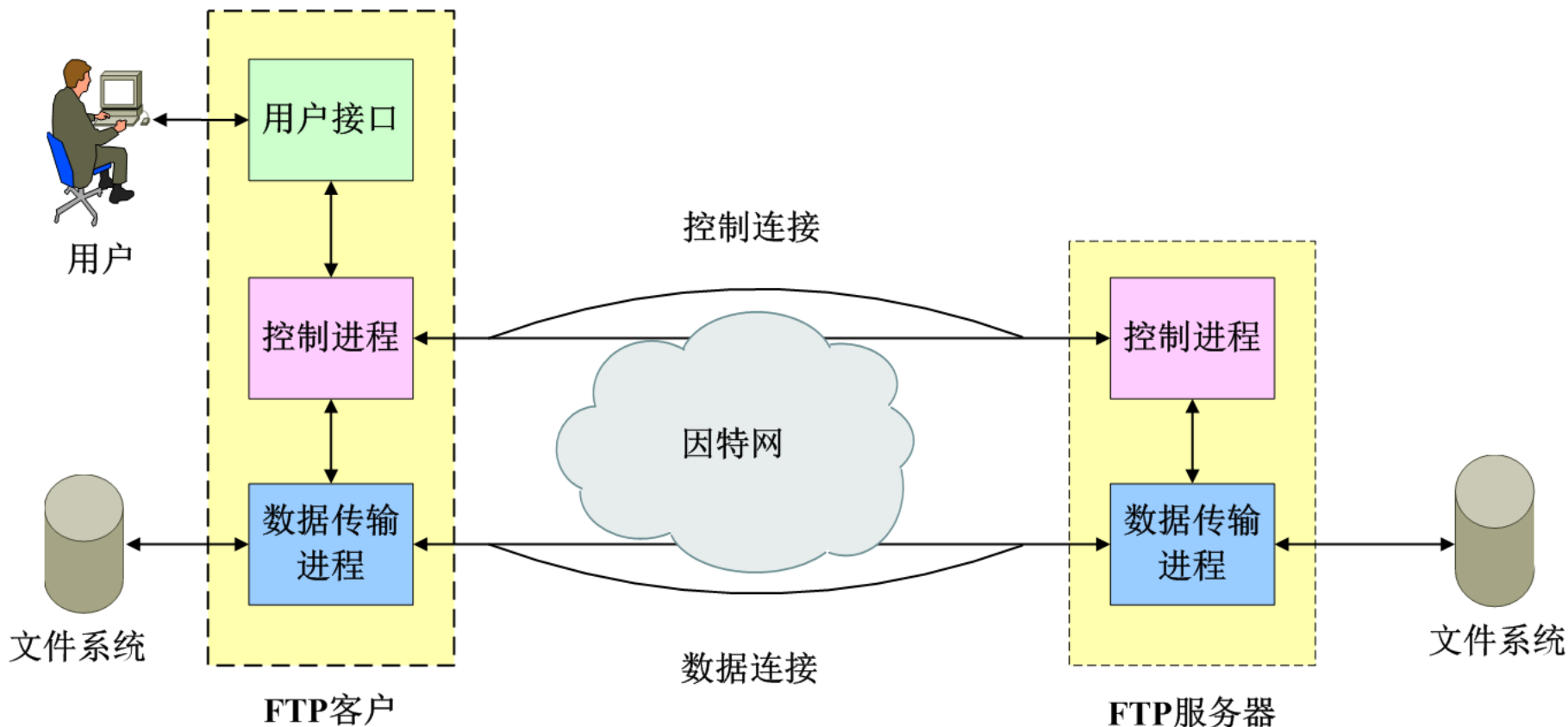
- 文件传输协议（File Transfer Protocol, FTP）主要用于在网络上实现文件的双向传输。
 - 文件传输服务是指将文件从一台计算机上传送到另一台计算机上，传输的文件类型包括普通文字处理文档、声音和图像等多媒体文件。
- 

- 将文件从本地计算机上发送到远程计算机上的过程称为FTP上载（upload），而将远程计算机上的文件传输到本地计算机上的过程称为FTP下载（download）。
 - 文件传输服务遵循TCP/IP协议组中的文件传输协议。
 - 因特网用户可以通过FTP在任意两台因特网主机之间进行文件的拷贝。
 - 使用FTP时必须首先登录，在远程主机上获得相应权限以后，才可上载或下载文件。
- 


FTP工作原理

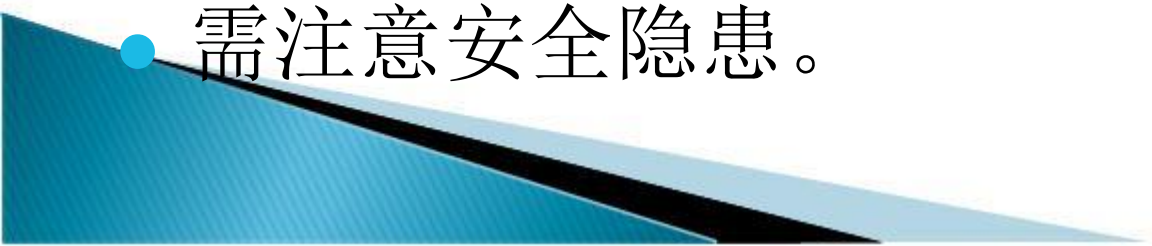
- 与大多数因特网服务一样，FTP采用C/S工作模式。
- 用户通过客户端程序向服务器发出命令请求，服务器执行用户所发出的命令请求，并将执行的结果返回到客户机。
- 客户和服务端之间建立两个TCP连接：
 - 控制连接：服务器监听端口，TCP/21
 - 数据连接：数据传输端口，TCP/20

FTP客户与服务服务器之间的连接

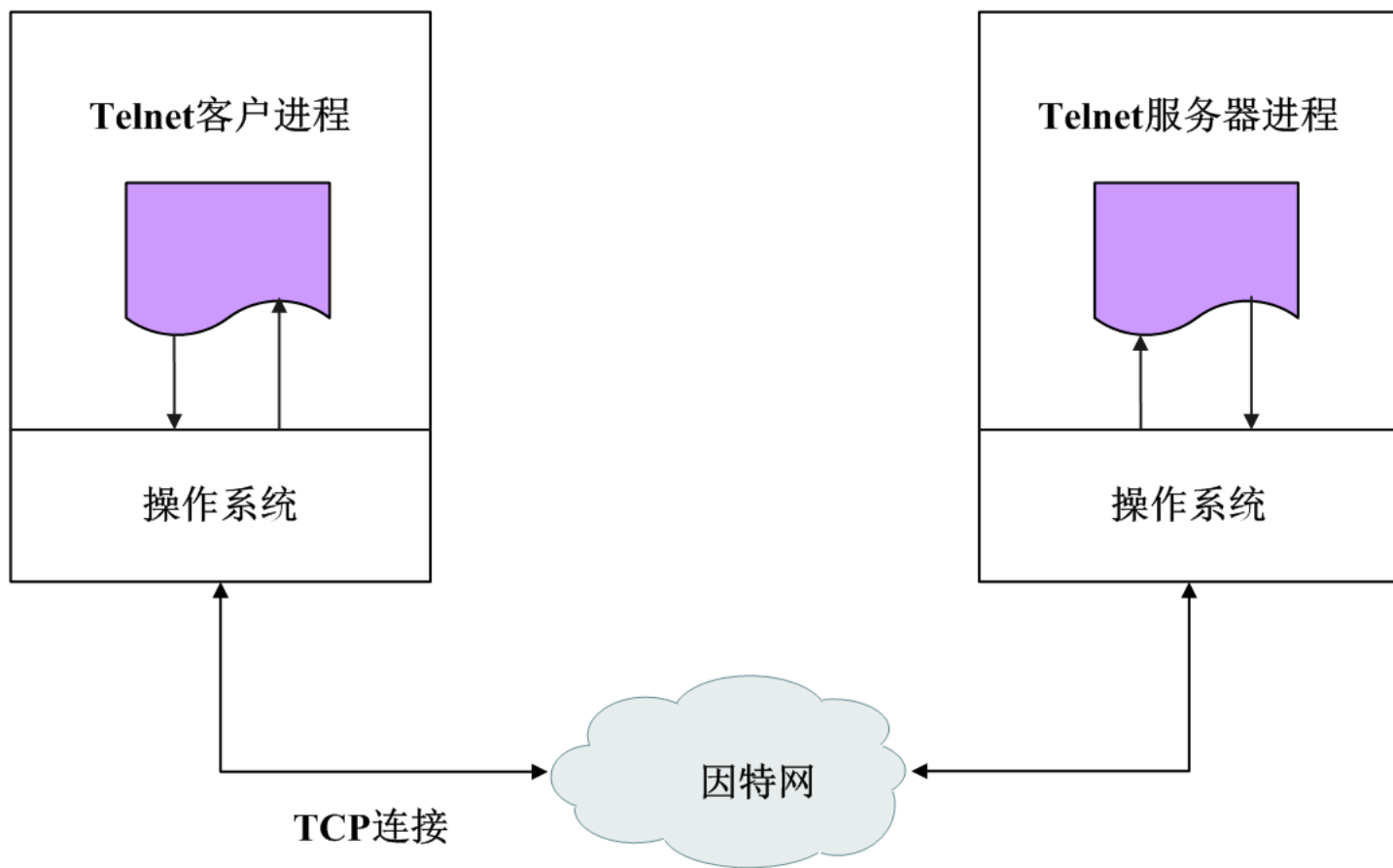


7.4.4 远程登录

- 远程登录是一种访问因特网的方式，也是因特网上较早提供的服务之一。
 - 实现远程登录的标准协议和主要方式是TELNET协议。
 - 用户把本地主机当作远程主机的一个终端，通过TELNET命令，直接访问和控制远程系统。
 - IETF RFC 854对TELNET做出了明确的定义和规范。
- 

- TELNET采用客户/服务器的工作模式
 - 本地计算机上运行TELNET客户进程，远程计算机上运行TELNET服务器进程。
 - 远程登录最大的优势：把本地计算机作为终端，跨系统和跨平台地访问和控制远程计算机系统。
 - 实现了其他网络服务所不能实现的功能。
 - 需注意安全隐患。
- 

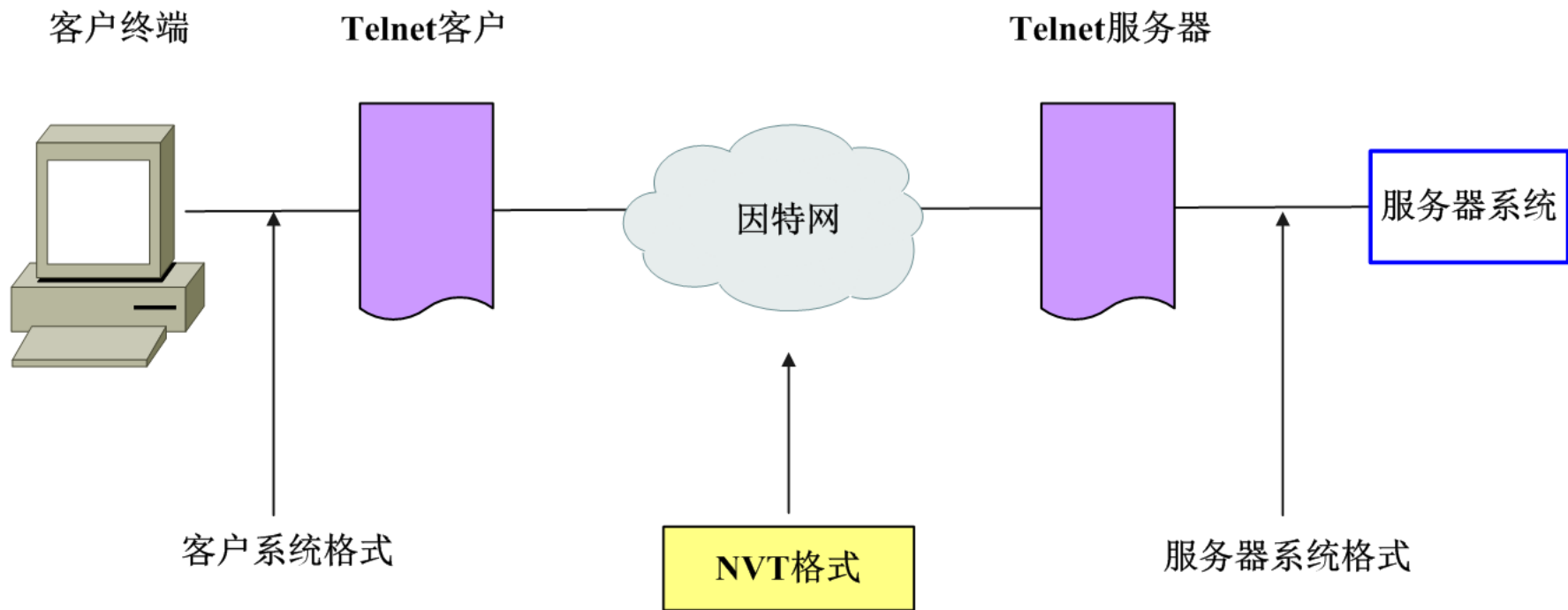
TELNET工作原理




- 不同的远程计算机及其操作系统会存在一定的差异。
- 为了适应这种异构性，TELNET定义了网络虚拟终端NVT格式，规定了TELNET数据和命令在因特网上传输的方式。



TELNET对NVT格式的使用



7.4.5 动态主机配置协议

- 动态主机配置协议（Dynamic Host Configuration Protocol, DHCP）是一个局域网的通信协议，用于对局域网内部的所有计算机自动配置（如自动分配IP地址）。
 - DHCP的前身是引导程序协议（BOOTstrap Protocol, BOOTP）。
 - IETF RFC 2131对DHCP做出了明确的定义和规范。
- 

- DHCP也采用C/S工作模式。
- DHCP客户向DHCP服务器请求IP地址，DHCP服务器响应客户请求，向DHCP客户分配IP地址。

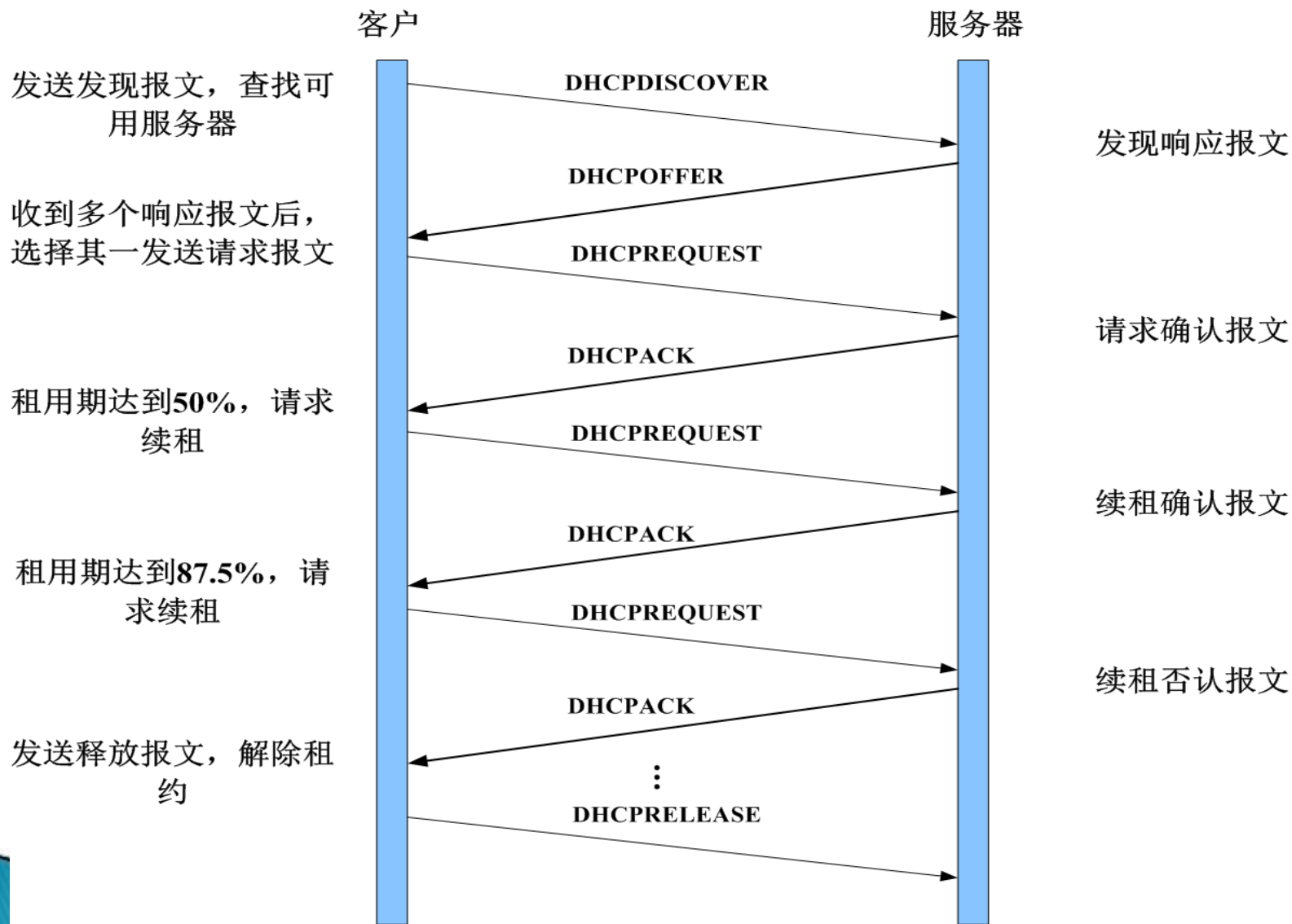


DHCP报文类型及其描述

名称	意义
DHCPDISCOVER	客户通过广播方式查找可用的服务器
DHCPOFFER	服务器响应DHCP发现报文
DHCPREQUEST	客户向服务器请求租用或续租IP地址
DHCPACK	服务器对DHCP请求报文进行确认
DHCPNACK	服务器对DHCP请求报文进行否定应答
DHCPDECLINE	客户向服务器说明IP地址被占用
DHCPRELEASE	客户不再租用服务器IP地址
DHCPINFORM	客户向服务器请求本地配置参数

- DHCP的工作过程主要包括DHCP发现、DHCP提供、DHCP选择、DHCP确认、DHCP续租和DHCP释放等阶段。
- DHCP基于传输层的UDP工作，客户使用UDP的68号熟知端口，服务器使用UDP的67号熟知端口。
- DHCP可以动态地分配IP地址，此时，分配给DHCP客户的IP地址都是临时地址，服务器将这些地址在一段有效时间内租给客户使用，在租用期间，服务器不会再将同一地址租用给其他用户。

DHCP工作过程



7.4.6 网络管理及简单网络管理协议

- 网络管理
- 简单网络管理协议

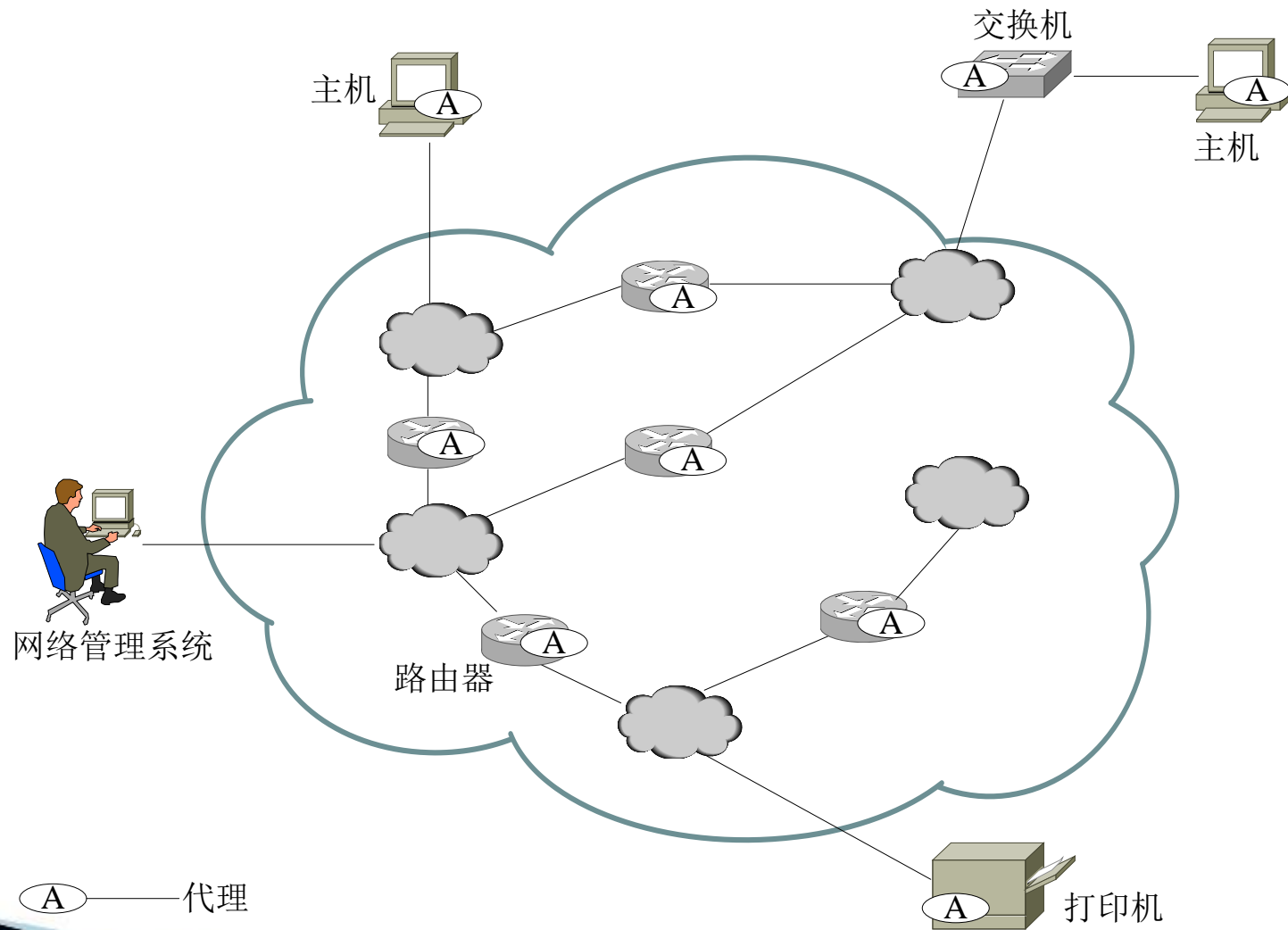


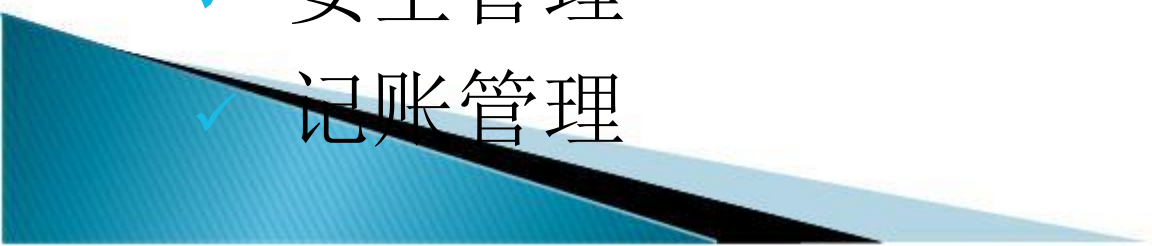
1. 网络管理

- 网络管理（network management）是指网络管理员对网络上的资源进行监视、配置、规划、综合、协调和控制时所发生的所有行为。



网络管理的体系结构




- 网络管理一般由三个部分组成：
 - ✓ 网络管理系统
 - ✓ 被管理设备
 - ✓ 代理组成
 - 网络管理主要功能包括以下几个方面：
 - ✓ 配置管理
 - ✓ 故障管理
 - ✓ 性能管理
 - ✓ 安全管理
 - ✓ 记账管理
- 

- 配置管理（configuration management）用于初始化网络，定义、识别和监控被管理设备，实现对被管理对象的控制及网络性能的优化。
- 故障管理（fault management）主要是对网络中的被管理设备进行检测、分析和排除，以保证网络服务的连续性和可靠性。
- 性能管理（performance management）通过收集和分析相关被管理设备及网络的当前状态，优化网络性能，以实现使用最少的网络资源并在最小时延的前提下，保障网络的连续、可靠的通信能力。

- 安全管理（security management）通过授权机制、访问控制、加密及密钥管理和安全访问日志的管理和维护等手段，提供对网络资源的安全访问。
- 记账管理（accounting management）对用户使用网络资源的情况进行记录，计算其费用和代价，同时通过监视和控制网络资源，并对网络资源使用情况进行合理调整，以提高网络资源利用率

2. 简单网络管理协议SNMP

- 简单网络管理协议（Simple Network Management Protocol, SNMP）是一个支持网络管理的标准。
 - SNMP使用最简洁的方式定义了一组操作，实现了网络管理的基本功能。
 - SNMP基于UDP（SNMP代理进程端口为161/UDP，网管进程端口为162/UDP）。
 - 目前使用的版本是SNMPv3。
- 

- SNMP体系的三个组成部分：
 - ✓ SNMP协议
 - ✓ 管理信息库（Management Information Base, MIB）
 - ✓ 管理信息结构（Structure of Management Information, SMI）



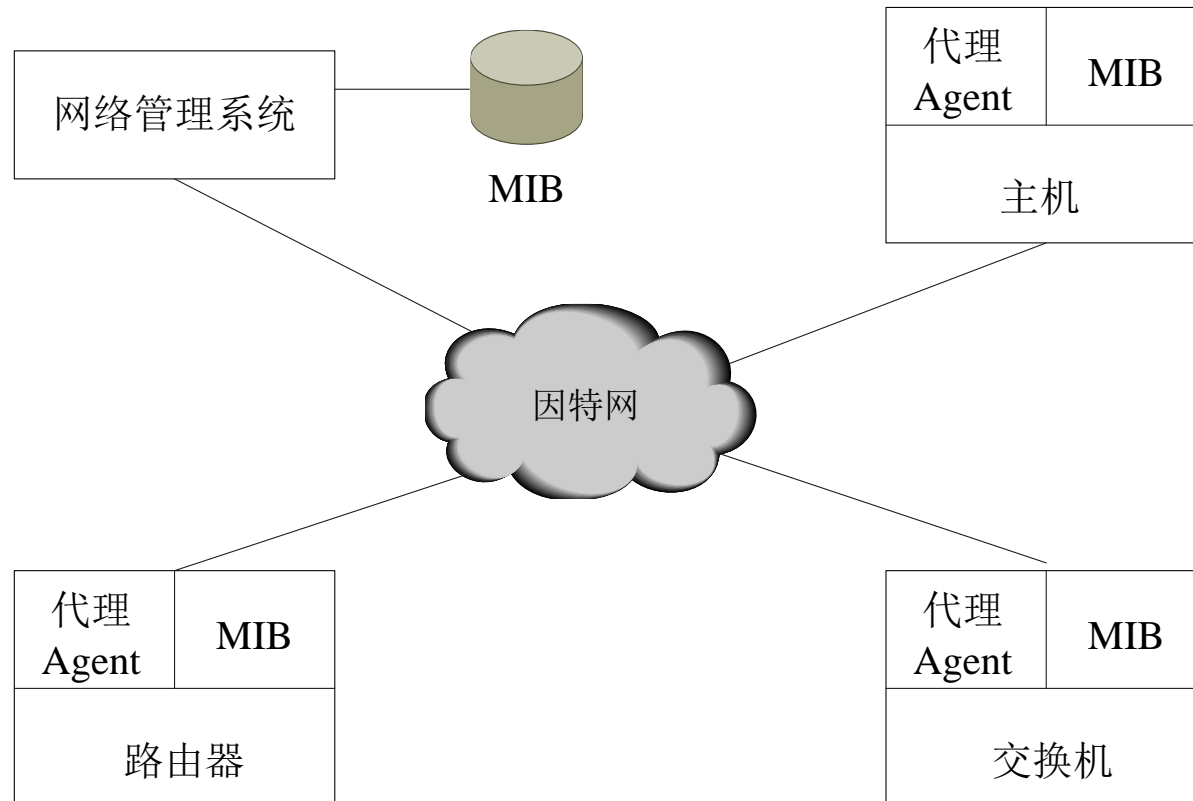
SNMP网络管理系统的基本结构

SNMP网络管理系统的构成：

1. 运行SNMP客户管理软件的网管工作站。
2. 嵌入SNMP代理程序的被管理网络设备。
3. 客户管理进程和代理进程通过SNMP进行通信。



SNMP网管系统的基本结构



SNMPv3的8种协议数据单元

名称	发送方/接收方	使用说明
GetRequest	管理员/代理	读取一个或多个变量的值
GetNextRequest	管理员/代理	读取列表中下一个变量的值
GetBulkRequest	管理员/代理	读取大容量数据
SetRequest	管理员/代理	设置一个变量的值
Response	代理/管理员	对任意请求的响应
InformRequest	管理员/管理员	获取第三方信息
Trap	代理/管理员	报告异常事件
Report	管理员/管理员	管理员间差错报告

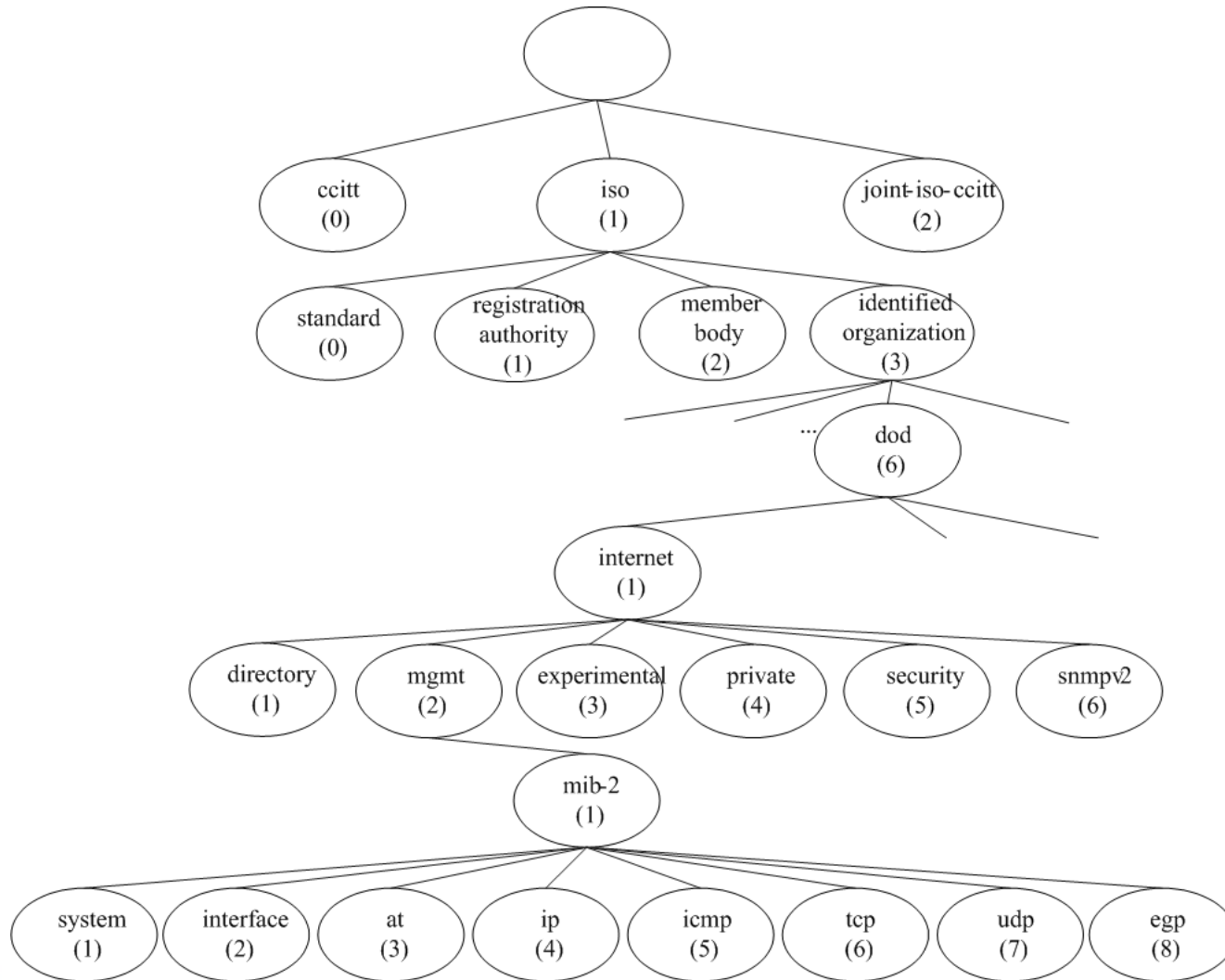
SNMP的管理信息库MIB

- 每个被管理的网络设备都有一个MIB，MIB中保存该设备的各项信息。
- MIB中的每个对象都有一个唯一标识——对象标识符（object identifier），由对象在树型数据结构中的位置决定。
- 管理信息结构SMI规定了树状的管理信息的分层结构。
- SMI采用抽象语法记法（Abstract Syntax Notation 1，ASN.1）的命名方案定义MIB对象。

MIB2包含的部分信息

类别	标识符	包含信息
System	1	主机或路由器的操作系统
Interfaces	2	各种网络接口
Address Translation	3	地址转换（例如，ARP映射）
IP	4	因特网协议软件
ICMP	5	因特网控制报文协议软件
TCP	6	传输控制协议软件
UDP	7	用户数据报协议软件
EGP	8	边界网关协议软件
SNMP	11	简单网络管理协议软件

ASN.1对象命名树



7.5 因特网上的新型应用


- 网络应用的新模式——P2P架构。
- 对等方既是客户机， 又是服务器。对等方不为服务提供商所有， 可为用户控制。
- 不像C/S模式要求要求服务器总是打开， 任意间断连接的主机（对等方）可直接通信。
- P2P的两个重要应用
 - 基于P2P的文件分发
 - 基于P2P的因特网电话服务

7.5.1 基于P2P的文件分发

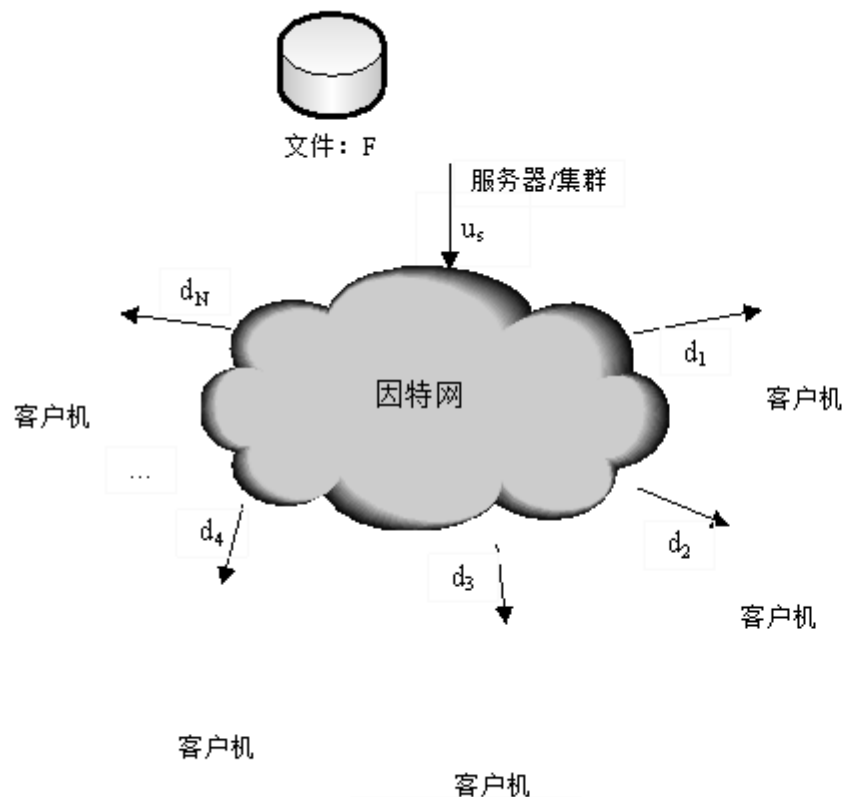
- 基于P2P的文件共享协议，如：
 - Napster
 - Gnutella
 - eDonkey
 - eMule和BitTorrent等
- 核心思想：每个结点既是文件的下载者，也是文件的上传者。



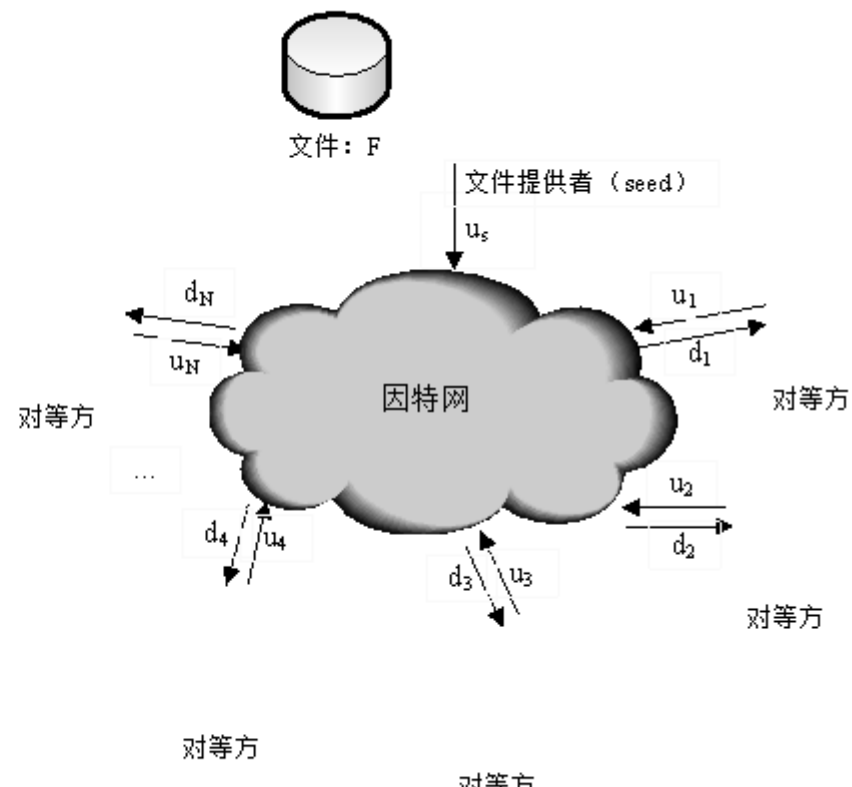
1. C/S和P2P模式下的文件分发

- 在基于客户机/服务器的文件分发中，文件的副本需要从服务器发向每个对等方，服务器因此消耗了大量的网络带宽。
 - 在基于P2P的文件分发中，每个对等方得到某个文件的一部分后可以重新将这部分文件继续分发出去，从而提高了服务器的分发效率。
- 

文件分发问题示例



a) C/S体系结构



b) P2P体系结构

简化的定量模型

- 服务器将一个文件分发给一组固定的对等方,通过因特网相连。一个C/S和P2P都适用的简单定量模型:
 - u_s 服务器(或种子结点seed)接入链路的上载速率
 - u_i 第 i 个对等方接入链路的上载速率
 - d_i 第 i 个客户端 (或对等方) 接入链路的下载速率
 - F 被分发的文件长度
 - N 要获得文件副本的对等方的数量
 - $D_{mini-CS}$ 和 $D_{mini-P2P}$ 分别是两种体系结构下 N 个对等方得到文件副本所需要的最小分发时间 (**minimum distribution time**) 。

- 为了进一步简化问题，可以忽略一些次要因素。
- 不考虑服务器集群和单服务器的区别。
- 因特网核心拥有足够的带宽，网络传输的瓶颈在网络接入链路上。
- 各结点网络接口带宽都用于文件分发。



基于P2P的文件分发

对最小分发时间的简单描述（ Rakesh Kumar）：

- 1) 设 d_{min} 为下载速率最小的对等方的下载速率，即 $d_{min}=\min\{d_1, d_2, \dots, d_N\}$ 。下载速率最小的对等方至少需要 F/d_{min} 秒才能获得整个文件，这意味着 $D_{min-P2P}$ 不会小于 F/d_{min} 。
- 2) 在文件分发的初始阶段，只有种子结点拥有文件。种子结点向网络中分发第一份文件拷贝至少需要 F/u_s 秒，这意味着 $D_{min-P2P}$ 不会小于 F/u_s 。

3) 整个P2P系统的总上载能力为:

$$u_{total} = u_s + u_1 + \dots + u_N,$$

系统的总上载量为 NF 比特, 这不可能以快于 u_{total} 的速率完成。所以, 系统总的上载时间至少是 NF/u_{total} 秒, 这意味着 $D_{min-P2P}$ 不会小于 NF/u_{total} 。

综上所述，最小分发时间取决于上述三项中的最大值：

$$D_{\min - P2P} \geq \max \left\{ \frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^N u_i} \right\}$$

- 不考虑其它影响因素（如对等方继续分发文件的不同算法），可以取上式的下限，得到：

$$D_{\min - P2P} = \max\left\{ \frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^N u_i} \right\}$$

基于C/S的文件分发

- 对应地，客户/服务器体系结构的最小分发时间就可以简单表述为：

$$D_{\min - cs} = \max\left\{ \frac{NF}{u_s}, \frac{F}{d_{\min}} \right\}$$

C/S和P2P的最小分发时间比较


- 为考察系统的可扩展性，对比系统总的上载时间。
- 假定所有对等方上载速率 u ，服务器 $u_s=10u$
- 设 $F/u=c$ 为常量
- P2P和C/S系统总的上载时间分别为：

$$\frac{NF}{u_{Total}} = \frac{NF}{u_s + Nu} = \frac{NF}{10u + Nu} = \frac{F}{u} \left(\frac{N}{10 + N} \right) = c \left(\frac{N}{10 + N} \right)$$

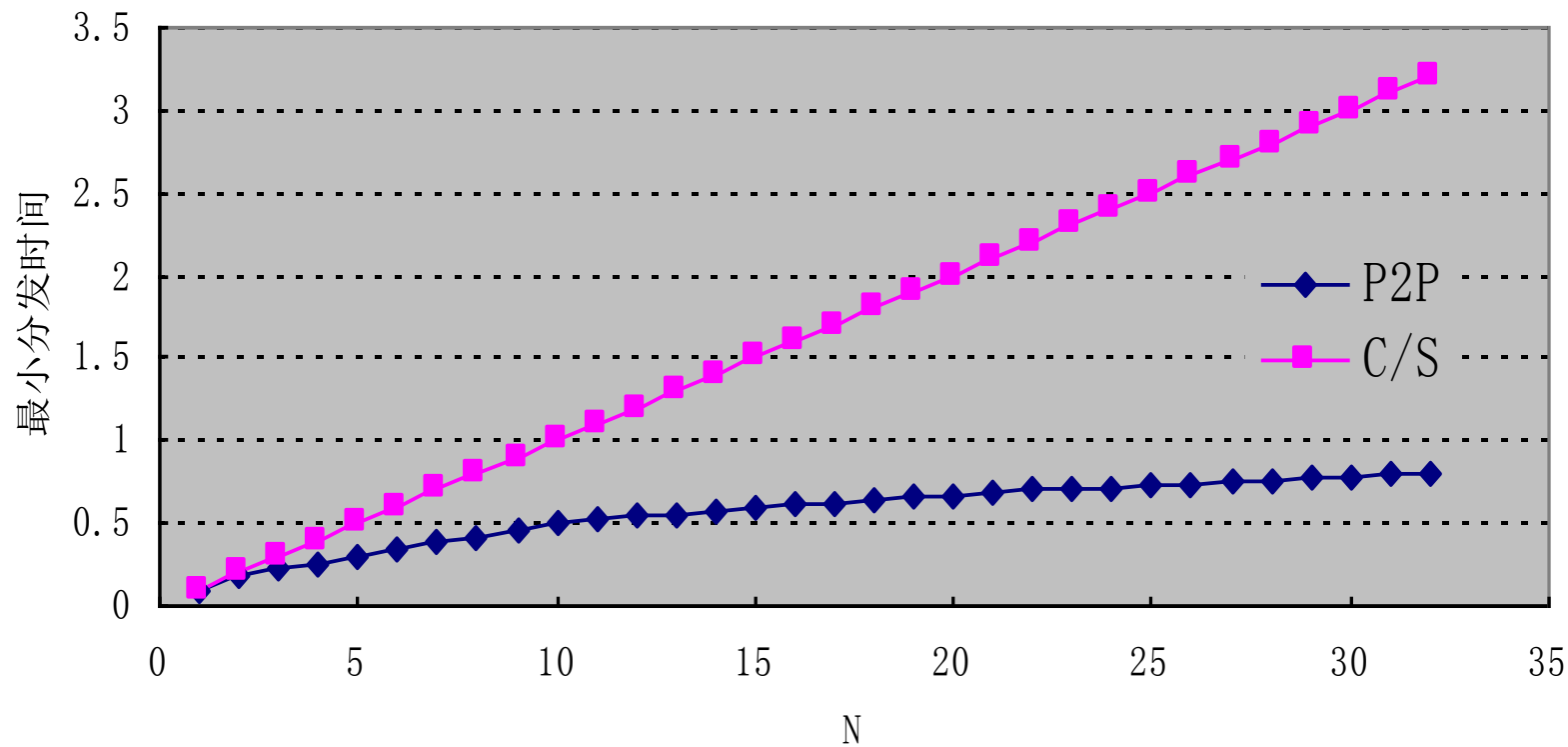
$$\frac{NF}{u_s} = \frac{NF}{10u} = c \frac{N}{10}$$

1. C/S体系结构：随着对等方数量的增加，总的上载时间随 N 呈线性增长并且没有界。
2. P2P体系结构：总的上载时间并不随着线性增长，而是接近单个对等方的分发时间 c （设 $c=1$ ，见下图）。

结论：采用P2P体系结构的应用系统具有良好的自我扩展性，这是因为每个对等方除了下载文件外还是文件的上传者。



C/S和P2P的最小分发时间比较




2. Bit Torrent （BT） 工作原理

- Bit Torrent是由Bram Cohen设计的开源软件。
- 一种支持多点传输的P2P文件分发协议。
- Bit Torrent系统：
 - 追踪器（tracker）
 - 种子（seed）
 - 下载者（leecher）



Bit Torrent系统

- **追踪器 (tracker)**：目录服务器，记录当前在线所有结点的信息（IP地址，所拥有的文件段等）。帮助新加入的结点找到其他结点。如果用户想加入BitTorrent系统，需要先连上追踪器，得到其他结点的信息，然后才能开始P2P通信。
 - **种子 (seed)**：拥有完整文件的在线结点，不需要下载数据，只给其他结点上传数据。
 - **下载者 (leecher)**：只拥有部分文件的在线结点，下载数据的同时也上传数据给其他下载者。
- 

BitTorrent工作原理

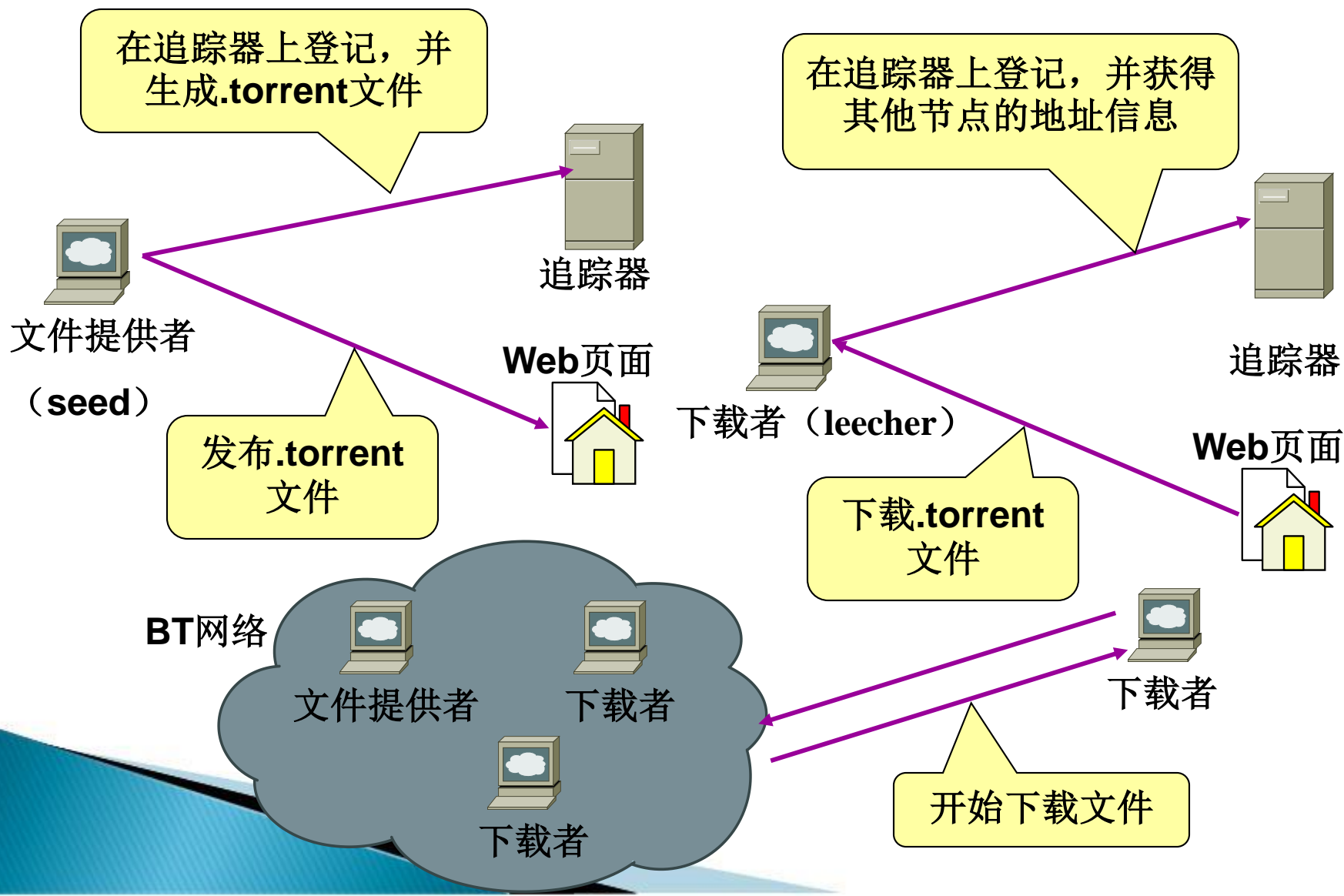
- (1) 文件提供者（种子）在追踪器上登记，并使用种子制作软件生成.torrent文件。（在BitTorrent系统中，追踪器并不提供文件下载，文件由种子提供）。
- (2) 文件提供者在Web页面上发布.torrent文件，供其他结点下载。
- (3) 下载者从Web页面下载.torrent文件，或者通过其它途径获得.torrent文件。



- (4) 运行BitTorrent软件，BitTorrent软件会根据.torrent文件中注明的追踪器地址与其建立连接，并从追踪器处获取其他结点的地址信息。
- (5) 根据结点的地址信息，BitTorrent软件与这些结点建立连接并下载数据。



BitTorrent工作原理示意图




Bit Torrent的文件段分发算法

- 决定文件段在结点间的传播过程的两个算法：
 - 最稀罕优先算法 (**Rarest First**)
 - 针锋相对算法 (**Tit-for-Tat, TFT**)




最稀罕优先算法

- 结点通过与它的邻居结点交换信息，可以知道在它的邻居结点中每个文件段的拷贝数。
 - 拷贝数最少的文件段组成一个集合，称为**最稀罕文件段（Rarest Pieces）**集合。
 - 结点在这个集合中随机选取一个自己感兴趣的文件段来下载。这就是最稀罕优先算法。
 - 保证了每个文件段的拷贝在结点之间能够较平均的分布。
- 


针锋相对算法

下载者的针锋相对算法：

- 每隔10秒，根据自己从邻居结点下载数据的速率来对邻居结点进行排序。
 - 选出速率最快且向自己提出下载请求的前三个结点来上传数据。
 - 每隔30秒，随机选择一个对自己提出下载请求的结点来上传数据。
 - 因此，在同一时间，下载者最多只能向四个邻居结点上传数据。
- 


针锋相对算法

种子的针锋相对算法


- 种子在同一时间也只能向四个结点上传数据。
 - 每隔30秒根据四个结点的下载速率对它们进行排序，断开与下载速率最低的那个结点的连接。
 - 然后随机选择一个对自己提出下载请求的结点来上传数据。
 - 种子只根据下载速率来选择结点，并没有综合考虑结点对系统的贡献，因为下载速率快的结点并不一定上传速率大。
- 

针锋相对算法的改进

对种子针锋相对算法的改进（ BitTorrent v4.0.0）：

- 种子每隔一段时间按下载时间的长短对结点进行排序。
 - 断开与下载时间最长的结点的连接。
 - 然后随机选择一个向自己提出下载请求的结点来上传数据。
 - 针锋相对算法的目的是为了激励结点更多的上传自己的数据。
- 

“搭免费车”问题


- P2P网络是基于结点自愿共享资源的。
 - 网络中的结点更多考虑最大化本结点的效用而不考虑P2P网络的整体效用，从而导致了非常严重的“搭免费车”（free-riding）现象。
 - 这对网络的高效工作不利，会降低对等网的可用性、可靠性、健壮性及可扩展性，进一步加剧会导致整个P2P应用系统最终崩溃。
- 


- “搭免费车”问题，是指P2P网络中结点的自私行为所带来的共性问题：结点只享用系统中的资源，而不对系统贡献自己的资源，从而导致网络中可共享资源数量不断减少。
- 措施：“阻塞”算法。让每个结点限制允许下载的连接数，只允许那些为它上传数据最多的结点从它这里下载数据；对于那些不传输数据，或者传输数据较少的结点进行阻塞。

“阻塞”算法举例

- 假设结点S分别与4个结点A、B、C和D建立连接，A向S传了10MB数据，B向S传了20MB，C向S传了5MB，D向S传了15MB。
- 在结点S看来，4个结点的优先级为B、D、A、C。
- 如果S只允许3个结点下载，那么S允许B、D和A的请求下载，阻塞结点C。
- 通过阻塞算法，BitTorrent系统就成功的杜绝了“搭免费车”现象，从而保证了系统公平性。

7.5.2 基于P2P的因特网电话服务

- Skype是即时通讯（Instant Message, IM）的软件。
 - 提供PC到PC、PC到电话、电话到PC和PC到PC的电话及视频会议服务。
 - 和传统即时通讯软件不同之处：Skype是一种以P2P技术为基础的应用程序。
 - 由创建Fastrack和Kazaa的开发人员设计开发，并于2005年被eBay公司以26亿美元的价格收购。
- 

- 实现了将网络资源分散（即不是利用集中式的服务器资源，而是利用各个结点的网络资源）。
 - 语音呼叫的接通率、语音质量在很大程度上甚至超过传统的电话网络。
 - 任何时候通常都有数以百万计的用户与之连接。
 - Skype有以下一些特性：
- 

Skype的特性1——音质清晰

- 采用第三代的P2P网络技术(Global Index, GI), 是一个由超结点组成的多点网络, 每个点之间均保持沟通。将这些结点资源动态组合, 参与流量分配、路径选择、处理需要较大带宽的任务等, 并保证最小的延时。
- 设计上彻底解放传统意义上300Hz到3000Hz频率的电话语音效果。可以使用户听到所有频率的语音, 即使是最低沉的或最尖锐的语音。




Skype的特性2——安全性好

- 采用了端到端的加密方式，Skype在信息发送前进行加密，在接收端进行解密。保证信息的安全性。
- 采用了数字签名的方式，对公共目录中存储的和用户相关的数据都采用了数字签名，保证数据无法被篡改。




Skype的特性3——跨平台性


- 有适用于Windows操作系统、Pocket PC、和Mac OS和Linux操作系统的版本，语音通话、文件交换等数据传输都可跨平台进行。
 - 所采用的底层技术保证了其可以很容易的移植到不同的终端设备上，更加适应终端设备和通讯技术的发展。
 - 其他优势：使用简单、功能强大；节省资源；超强的穿透能力；超大文件传输能力以及开放的应用程序接口。
- 

Skype的基本原理


与其他VOIP通信软件一样，Skype也是利用因特网来打电话。其基本原理如下：

- (1) 通过电话机将语音转变成模拟信号；
 - (2) 模拟信号通过VOIP终端计算机转换为IP数据报；
 - (3) IP数据报通过因特网传送到远端一个VOIP终端计算机；
 - (4) VOIP终端主机将IP数据报转换为模拟信号；
 - (5) 模拟信号通过电话机转换成语音。
- 

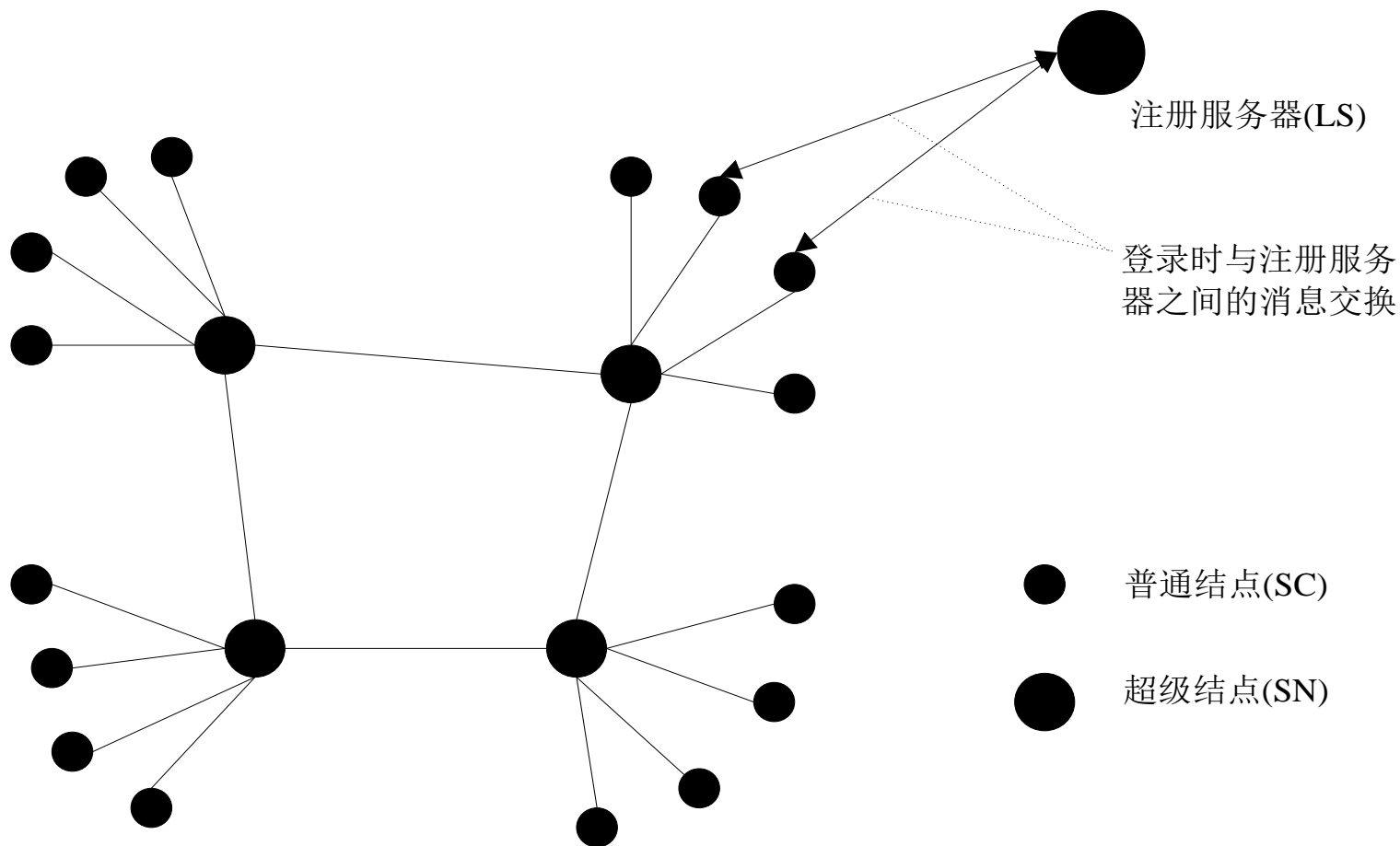
Skype的网络结构与通信过程

- 与其他VOIP通信软件不同的是，Skype使用私有VOIP协议（属于商业秘密）。
 - Skype的网络结构：
 - 注册服务器（Login Server, LS），此外没有任何其它的集中服务器。
 - 用户结点分为：普通结点或称为普通对等方（Skype Client, SC)和超级结点或称为超级对等方（Super Node, SN）。
 - 普通结点必须连接到超级结点上。
- 

Skype的网络结构

- **注册服务器：** Skype惟一需要维护的设备，负责完成客户端的注册，保证用户名的全球唯一性，存储并管理用户名和密码信息，对用户进行身份认证。
 - **普通结点：** 普通主机终端，只需要下载了Skype的应用，就具有提供语音呼叫和文本消息传送的能力。
 - **超级结点：** 满足某些要求的普通结点：具有公网地址、具有足够的CPU、存储空间足够大、具有足够的网络带宽。
- 

Skype的网络结构示意图




Skype网络结构

Skype的通信过程

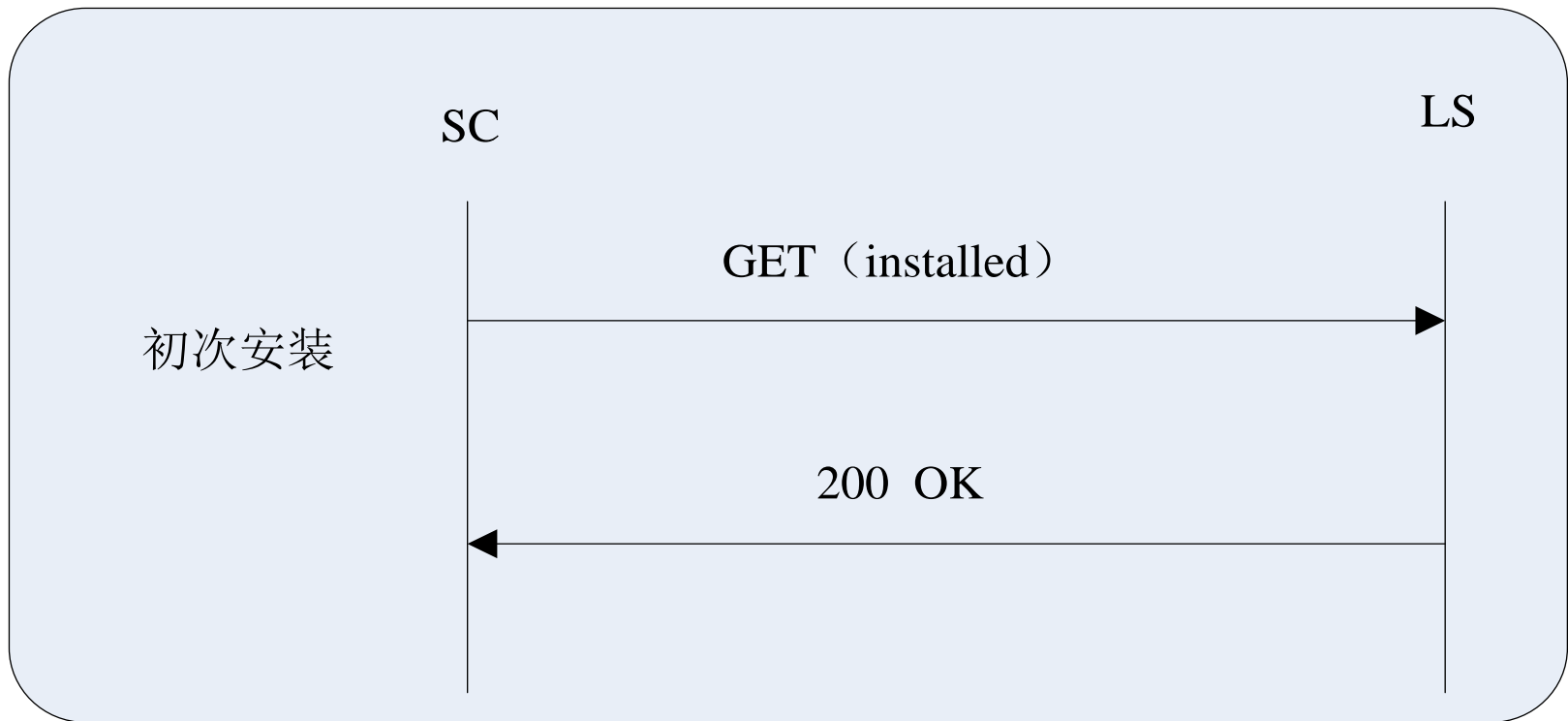
- Skype的通信过程：
 1. 启动
 2. 注册/认证
 3. 查找用户（用户搜索）
 4. 呼叫和释放。
- 注册流程只是在用户初次安装了Skype的客户端软件后进行，后期使用的过程中该步骤就变成认证过程。

1. 启动流程

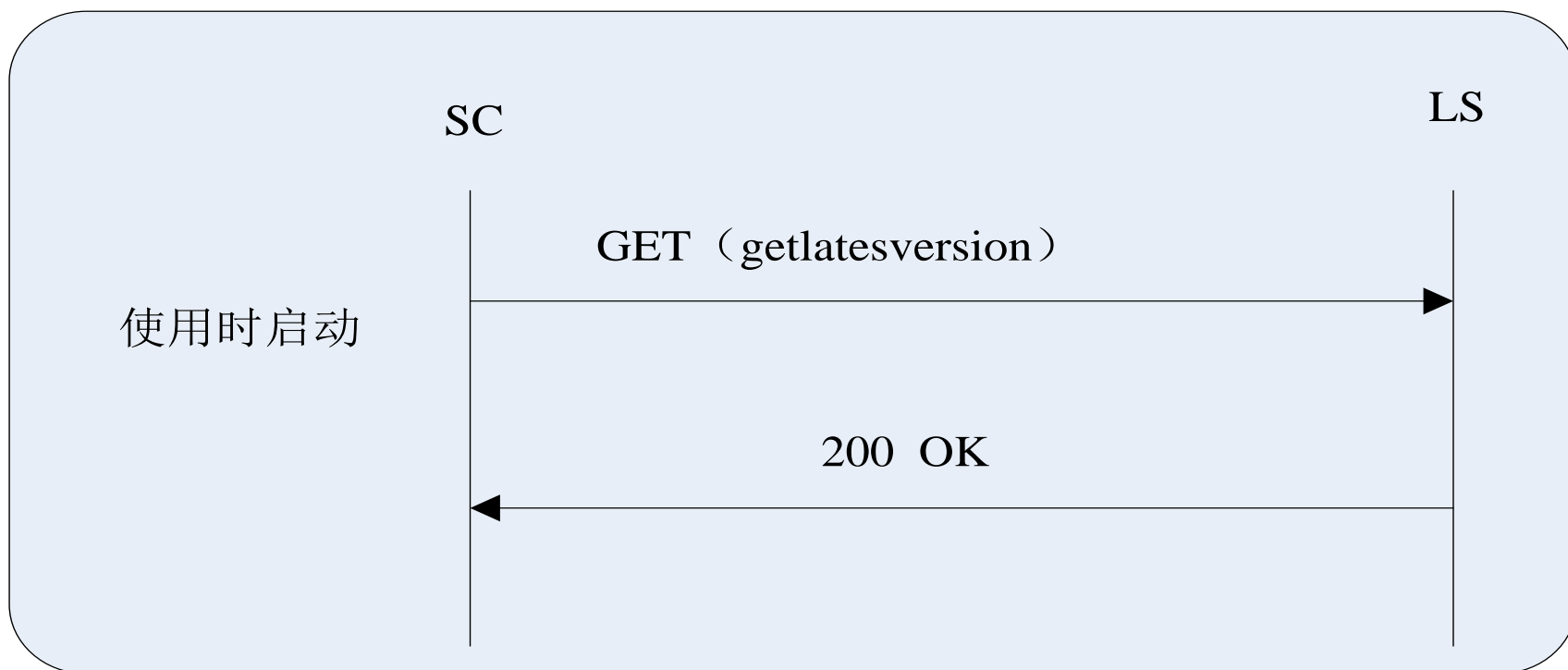
1.启动流程

- Skype的用户终端启动时，采用HTTP协议连接到注册服务器。
 - 用户初次安装的启动流程中携带“installed”的参数以及所装Skype的版本号。
 - 使用时的启动流程则在消息中携带“getlatesversion”参数，检查是否有新的版本。具体流程分别见如下两图。
- 

初次安装Skype的启动流程



用户使用Skype时的启动流程



2. 注册/认证流程


2. 注册/认证流程

- 用户启动Skype后，首先需要连接到超级结点，通过超级结点向注册服务器发送身份认证信息。
- 注册服务器验证用户名和密码的合法性。然后向其它对等结点发送在线信息。
- 注册服务器需要判断该终端所在私网的NAT和防火墙类型。如果该终端先前默认的超级结点已不可用，则还要查找具有公网地址的Skype结点来作为该终端的超级结点，以维持该终端与Skype网络的连接。

- 一旦超级结点都不可用，Skype的客户端采用了尽力而为的方式进行注册：
 - 先用UDP包试注册
 - 不成功超时则用TCP 80端口
 - 再不成功则用TCP 443端口
 - 如果仍然失败就显示无法登录。
 - 整个的登录过程可以重复4次。
- 通常为防止其超级结点不可用，客户端定期维护一个可选连接结点列表。

3. 查找用户

3. 查找用户

- 采用全球索引（GI）的技术进行用户搜索，超级结点之间采用全分布式的连接。每个超级结点具有最小时延前提下所有可用的用户和资源的全部信息。
 - 采用了两种机制来保证对用户搜索的顺利完成：
 1. 启动后向所有列表中的用户发送其上线信息，其它用户响应各自的信息。
 2. 在中间结点缓存查找到的用户信息。
- 

查找用户的过程

- 对于有公网地址的客户端：点击发送要查找的用户信息→通过超级对等方获取四个结点地址→若不成功→报告超级对等方→获取八个结点地址→.....→成功（或失败返回）。
- 对于位于私网内的受限客户端：首先客户端将需要查找的用户信息发送给其超级结点，然后由超级结点完成查找后返回给私网内的客户端。

4. 呼叫建立和释放

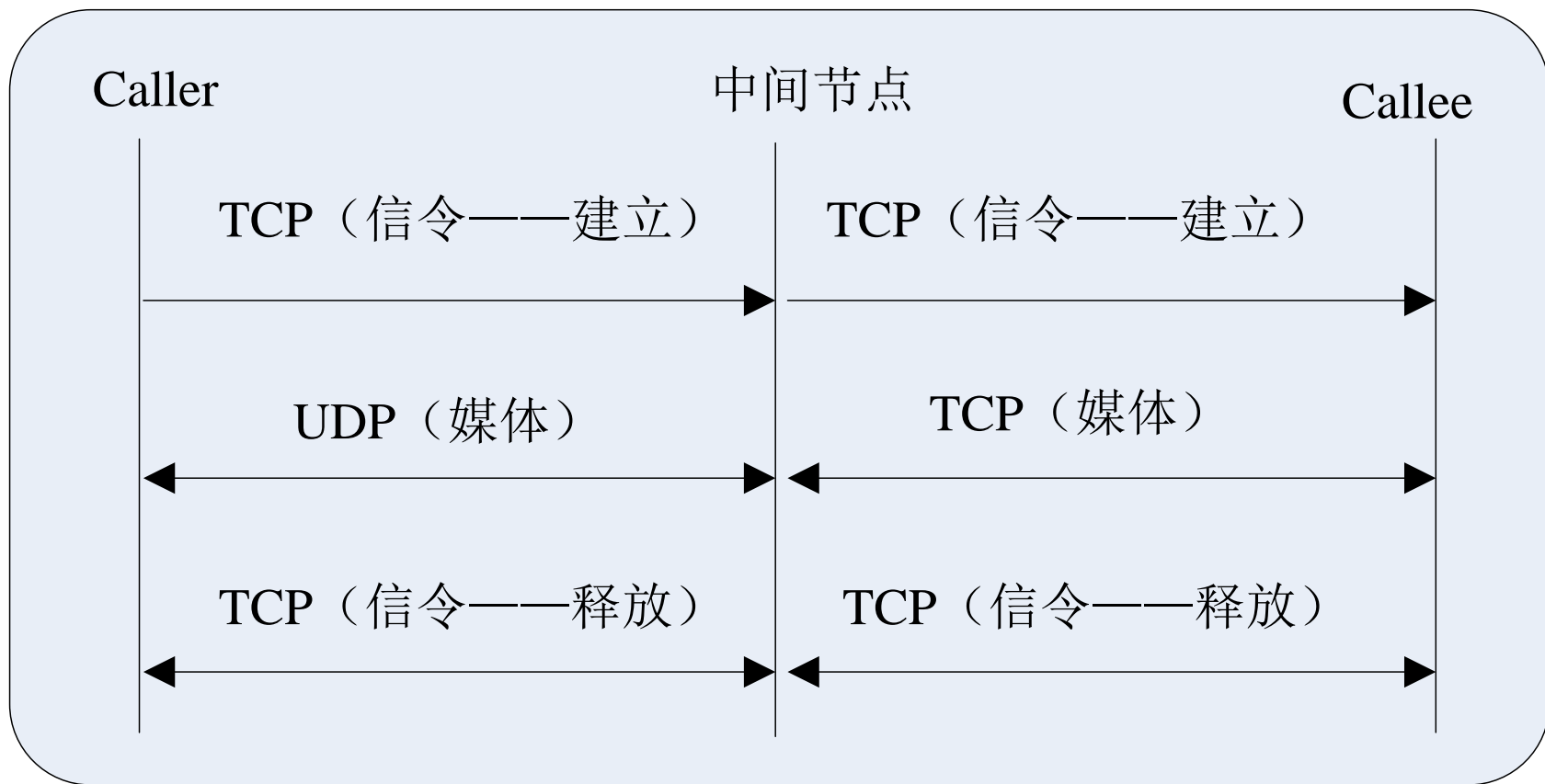
4. 呼叫建立和释放：

- 查找到希望连接的用户后，可以将其加入好友列表。Skype用户可以随时与在线的好友进行呼叫。
- 呼叫建立和释放的过程相对简单。分用户位于公网和位于私网内部的两种情况，两种情况下的呼叫建立和释放流程分别如下两图所示。






Skype主被叫都在公网的呼叫建立和释放过程



Skype用户至少有一方在私网内的呼叫建立和释放过程

- Skype的呼叫信令都采用TCP封装。
 - 媒体流则使用UDP封装。
 - 当任何一方用户位于限制UDP包的防火墙内时，受限制一方的媒体流就会采用TCP封装。
 - 当至少有一方用户位于私网内时，所有的信令和媒体消息都经过一个或多个中间结点（一般为超级结点SN）转发。
 - 无需担心用户通话的媒体流经过中间结点转发而被窃听，因为Skype采用了消息的端到端加密机制。
- 

课后思考题

- 应用层协议和传输层协议的区别是什么？
- 应用层协议的种类为什么比传输层多？
- www 服务和ftp 服务有什么区别？
- www服务能否替代ftp服务？ www服务能否替代Telnet服务？

